

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМ. В. ДАЛЯ  
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ  
КАФЕДРА КОМП'ЮТЕРНИХ НАУК ТА ІНЖЕНЕРІЇ

До захисту допускається  
Завідувач кафедри  
\_\_\_\_\_ Скарга-Бандурова І.С.  
« \_\_\_\_ » \_\_\_\_\_ 2018 р.

**ДИПЛОМНИЙ ПРОЕКТ (РОБОТА) БАКАЛАВРА**  
**ПОЯСНЮВАЛЬНА ЗАПИСКА**

НА ТЕМУ:

Інформаційна система розрахунку стажу роботи та нарахування  
пенсій

Освітньо-кваліфікаційний рівень “бакалавр”  
Напрямок підготовки 6.050101 – “Комп’ютерні науки та інформаційні технології”

Керівник проекту:

\_\_\_\_\_

(підпис)

Сафонова С.О.

\_\_\_\_\_

(ініціали, прізвище)

Консультант з охорони праці:

\_\_\_\_\_

(підпис)

Критська Я.О.

\_\_\_\_\_

(ініціали, прізвище)

Здобувач вищої освіти:

\_\_\_\_\_

(підпис)

Шипілов Г.О.

\_\_\_\_\_

(ініціали, прізвище)

Група:

КН-14з

Севєродонецьк 2018

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет Інформаційних технологій та електроніки  
Кафедра Комп'ютерних наук та інженерії  
Освітньо-кваліфікаційний рівень бакалавр  
Напрямок підготовки 6.050101 Комп'ютерні науки та інформаційні технології  
(шифр і назва)  
Спеціальність \_\_\_\_\_  
(шифр і назва)

**ЗАТВЕРДЖУЮ:**

Завідувач кафедри \_\_\_\_\_  
І.С. Скарга-Бандурова  
« \_\_\_\_\_ » \_\_\_\_\_ 2018 р.

**З А В Д А Н Н Я  
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) БАКАЛАВРА**

Шипілову Глібу Олеговичу

(прізвище, ім'я, по батькові)

1. Тема роботи Інформаційна система розрахунку стажу роботи та  
нарахування пенсій

керівник проекту (роботи) Сафонова С.О., к.т.н., доцент

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від " \_\_\_\_\_ " \_\_\_\_\_ 201\_ р. № \_\_\_\_\_

2. Термін подання студентом роботи \_\_\_\_\_

3. Вихідні дані до роботи матеріали переддипломної практики

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) Аналіз предметної області постановка задачі. Розробка структури програмного забезпечення. Розробка бази даних. Розробка алгоритму роботи програмної системи. Програмна реалізація. Розробка інтерфейсу. Охорона праці.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)  
Електронні плакати

## 6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Охорона праці	ст.викл. кафедри КНІ Критська Я.О.		

7. Дата видачі завдання \_\_\_\_\_

Керівник

\_\_\_\_\_ (підпис)

Завдання прийняв до виконання

\_\_\_\_\_ (підпис)

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту ( роботи )	Примітка
1	Огляд літератури з теми ДП і постановка задачі	14.05.18-16.05.18	
2	Розробка бази даних	17.05.18-20.05.18	
3	Розробка алгоритму роботи програмної системи	21.05.18-24.06.18	
4	Програмна реалізація	25.06.18-30.06.18	
5	Розробка інтерфейсу	01.06.18-04.06.18	
6	Розробка розділу охорона праці	05.06.18-07.06.18	
7	Оформлення електронних плакатів	08.06.18-10.06.18	
8	Оформлення пояснювальної записки	11.06.18-13.06.18	

Здобувач вищої освіти

\_\_\_\_\_ ( підпис )

Шипілов Г.О.

\_\_\_\_\_ (прізвище та ініціали)

Керівник

\_\_\_\_\_ ( підпис )

Сафонова С.О.

\_\_\_\_\_ (прізвище та ініціали)

## РЕФЕРАТ

Пояснювальна записка до дипломного проекту (роботи) бакалавра: 154 с., 47 рис., 21 табля., 26 бібліографічних джерел посилань, 4 додатка.

Об'єкт розробки: процеси роботи працівника відділу кадрів.

Мета роботи: розробка програмного забезпечення для підвищення ефективності роботи працівника відділу кадрів з формування подання на призначення пенсії.

В проекті виконано:

1. Розглянута загальна схема роботи працівника відділу кадрів з формування подання на призначення пенсії та сформульоване технічне завдання дипломного проекту.
2. Побудована структура програмного забезпечення.
3. Розроблена база даних.
4. Розроблено алгоритм роботи програмної системи.
5. Розроблено програмне забезпечення та наведені інструкції оператора.
6. Здійснений аналіз потенційних небезпечних і шкідливих виробничих чинників проєктованого об'єкта, що впливають на персонал та приведені рекомендації щодо пожежної безпеки.

Отримано наступні результати: було розроблено технічне завдання та обрані програмні засоби для реалізації, база даних, програмне забезпечення для розрахунку стажу роботи та нарахування пенсій.

Практичне значення, галузь застосування роботи: розроблена ІС забезпечує автоматизацію діяльності працівника відділу кадрів, що дає можливість швидкого прийняття рішень, планування роботи, скорочення часу пошуку необхідної інформації, зменшення кількості паперових носіїв; спрощення роботи зі звітами.

**Ключові слова:** НАРАХУВАННЯ ПЕНСІЙ, ВІДДІЛУ КАДРІВ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ДОВІДНИКИ, БАЗА ДАНИХ.

Умови одержання дипломного проекту: СНУ ім. В. Даля, пр. Центральний 59-А, м. Сєвєродонецьк, 93400.

## ЗМІСТ

<b>СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ.....</b>	<b>5</b>
<b>ВСТУП.....</b>	<b>6</b>
<b>1 ДОСЛІДЖЕННЯ ОБ’ЄКТА АВТОМАТИЗАЦІЇ. ПОСТАНОВКА ЗАДАЧІ.....</b>	<b>8</b>
<b>1.1 Аналіз та оцінка існуючих процесів розрахунку пенсійних відрахувань ....</b>	<b>8</b>
<b>1.2 Оцінка можливих результатів автоматизації.....</b>	<b>12</b>
<b>1.3 Висновки та постановка задачі.....</b>	<b>14</b>
<b>1.4 Технічне завдання на розробку програмного забезпечення .....</b>	<b>15</b>
<b>2. ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ ПЗ.....</b>	<b>18</b>
<b>3. РОЗРОБКА ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....</b>	<b>22</b>
<b>3.1 Структура програмного забезпечення.....</b>	<b>22</b>
<b>3.2 Детальні вимоги за варіантами використання.....</b>	<b>28</b>
<b>3.3 Опис бази даних.....</b>	<b>37</b>
<b>3.4 Алгоритм розробки програмного забезпечення .....</b>	<b>47</b>
<b>3.5 Запити SQL.....</b>	<b>90</b>
<b>4. ОХОРОНА ПРАЦІ .....</b>	<b>92</b>
<b>ВИСНОВКИ.....</b>	<b>109</b>
<b>ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ .....</b>	<b>110</b>
<b>ДОДАТОК А Порядок та оформлення документів для призначення (перерахунку) пенсій.....</b>	<b>111</b>
<b>ДОДАТОК Б Елементи даних.....</b>	<b>114</b>
<b>ДОДАТОК В Лістинг програми .....</b>	<b>119</b>
<b>ДОДАТОК С Комп’ютерна презентація.....</b>	<b>147</b>

## СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

ОС - операційна система

ПК - персональний комп'ютер

ПЗ - програмне забезпечення

ЕД - елемент даних

НВ - нефункціональні вимоги

СПВ - специфікація програмних вимог

ФВ - функціональні вимоги

НДІ - нормативно довідкова інформація

БД - база даних

ІС - інформаційна система

## ВСТУП

Використання персональних комп'ютерів для автоматизації розрахунку пенсій є важливою складовою частиною системи інформаційного забезпечення діяльності будь-якого підприємства. Сам по собі автоматизований розрахунок усього потоку напрямків пенсійного обліку складається з безлічі рутинних операцій, пов'язаних з багаторазовим виконанням одних і тих же арифметичних дій, підготовкою різноманітних за формою звітних та платіжних документів і перенесенням даних з одних документів в інші.

Незважаючи на уявну простоту автоматизації бухгалтерської діяльності, не так-то легко отримати рішення зручне для недосвідченої в застосуванні комп'ютерів людини. Крім того, певну трудність представляє постійна зміна вимог, що пред'являються до розрахунку стажу, що збільшує потребу в гнучких програмних комплексах, що швидко адаптуються до нових умов.

Безумовно, комп'ютерна програма не може замінити грамотного бухгалтера або працівника відділу кадрів, але вона дозволить заощадити його час і сили за рахунок автоматизації рутинних операцій, знайти арифметичні помилки в обліку і звітності, оцінити поточне фінансове становище громадян, а також підприємства та його перспективи. Крім того, автоматизовані системи розрахунку пенсій здатні допомогти підготувати і зберегти в електронному вигляді первинні документи та звітні документи, а так само бланки часто повторюваних форм з уже сформованими реквізитами підприємства. Представлення для призначення пенсії - це багатокористувальницьке мережеве завдання, не пов'язане з іншими системами.

Завдання, яке повине вирішити програмне забезпечення:

- планування і складання списку працівників, які виходять на пенсію;
- формування списків по особам, які досягають пенсійного віку;
- підготовка пакету документів по кожному працівнику, який виходить на пенсію;

- звірка П.І.Б. і стажу шкідливої роботи за переданими від пенсійного фонду даними з інформацією, що знаходиться на підприємстві;
- облік працюючих пенсіонерів та інвалідів пенсіонерів;
- та ін..

Складання документа «Представлення для призначення пенсії» є трудомісткою і невід'ємною роботою при оформленні пенсії. Створення програмного продукту, який дозволить сформувати необхідний документ, автоматично розраховувати тривалість стажу роботи заощадить час роботи старшого інспектора групи з оформлення пенсій. Крім того, в процесі роботи в програмному продукті буде накопичуватися інформація по вже сформованим документами, що полегшить роботу при необхідності перерахунку пенсії або внесенні змін до документа.

Тому створення такого програмного продукту є актуальним завданням для багатьох підприємств.

Для реалізації даної задачі необхідно визначення специфічних вимог для призначення пенсії і детальна специфікація характеристик, можливостей, критичних атрибутів, і головних характеристик пропонованої системи.



# 1 ДОСЛІДЖЕННЯ ОБ'ЄКТА АВТОМАТИЗАЦІЇ. ПОСТАНОВКА ЗАДАЧІ

## 1.1 Аналіз та оцінка існуючих процесів розрахунку пенсійних відрахувань

Для оформлення пенсії необхідно зробити наступні операції:

- Планування та складання списків працівників, які виходять на пенсію в наступному році. Списки готуються на кожен місяць наступного року. Період підготовки списків з вересня по листопад. Списки підписує директор з кадрів і передають в пенсійний фонд.

- Підготовка пакету документів по кожному працівнику, який виходить на пенсію. Пакет документів включає:

1) заяву від працівника з проханням про призначення пенсії (за відсутності заяви

-пенсія нараховується автоматично, через 3 місяці після досягнення пенсійного віку);

2) спільне подання адміністрації та профспілкового комітету;

3) пільгову довідку (якщо була шкідливість);

4) додаток до пільгової довідки;

5) виписку з наказу з атестації;

6) аркуш про перевірку результатів атестації (Держ експертиза);

7) довідку про зарплату (за період, обраний працівником);

8) паспорт та інші особисті документи, що підтверджують особистий стаж (диплом, військовий квиток тощо);

9) анкету відомість про пенсію (для отримання пластикового пенсійного посвідчення);

10) пам'ятку пенсіонера;

Якщо йде відмова з надання пенсії, пенсійний фонд повідомляє про таке рішення керівництво.

- Звірка П.І.Б і стажу шкідливої роботи за переданими від пенсійного фонду даними з інформацією, що знаходиться на підприємстві. Це необхідно, коли підприємство платить частину пенсії працівникам, які вийшли на пенсію за шкідливим стажем роботи, а не по досягненню пенсійного віку.

Якщо виявляються розбіжності в отриманій і наявній на підприємстві інформації, працівники групи пишуть службову записку на ім'я головного бухгалтера за підписом директора з кадрів із зазначенням виявлених розбіжностей.

- Отримання пенсійних посвідчень на працівників, що працюють на підприємстві в Пенсійному фонді і передача їх працівникам. Передача в пенсійний фонд і повернення, трудових книжок працівників підприємства.

- Реєстрація працівників, яким призначено пенсію.

- Перерахунок пенсій. Формування пакету документів для перерахунку пенсій:

1) заява;

2) довідка з місця роботи;

3) ксерокопія трудової книжки (нової частини, не заповненої на момент попереднього перерахунку або оформлення пенсії);

- Облік працюючих пенсіонерів.

- Облік працюючих інвалідів пенсіонерів.

- Робота з пенсійним фондом за роз'ясненнями інформації, зібраної в переданому пакеті документів.

- Консультації працівників підприємства з різних питань, пов'язаних з оформленням і нарахуванням пенсій.

- Звірка достовірності інформації в трудовій книжці з первинними документами.

- Робота зі з'ясування наявності дійсного шкідливого стажу і всіх обставин роботи у шкідливих умовах конкретних осіб.

- Перерахунок фактично відпрацьованого в шкідливих умовах часу для працівників, які працюють у шкідливих умовах непостійно.

На рисунку 1.1 зображена схема виконуваних операцій для нарахування пенсій до автоматизації.

В обов'язки співробітника відділу кадрів входить складання списків працівників, які виходять на пенсію в наступному році. Введення даних виконується вручну. Також величезна кількість часу йде на пошук цих даних. На це може піти від одного до трьох місяців. Після складання і перевірки списків їх відносять на підпис. Списки підписує директор з кадрів і передаються в пенсійний фонд. Готується пакет необхідних документів. Після підготовки пакету документів проводиться розрахунок стажу: вводяться дані по кожному працівнику з відповідних джерел ( 30-60 хвилин); вводяться дані в програму по кожному запису в трудовій книзі і виконується їх підсумовування (2-3 години); вводяться отримані дані по всіх записів у трудовій книжці і виконується їх підсумовування (близько 1-2 годин); виконується обчислення загального стажу з допомогою калькулятора (близько 20 - 40 хвилин); виконується обчислення загального стажу за отриманими даними за допомогою калькулятора (30-60 хвилин).

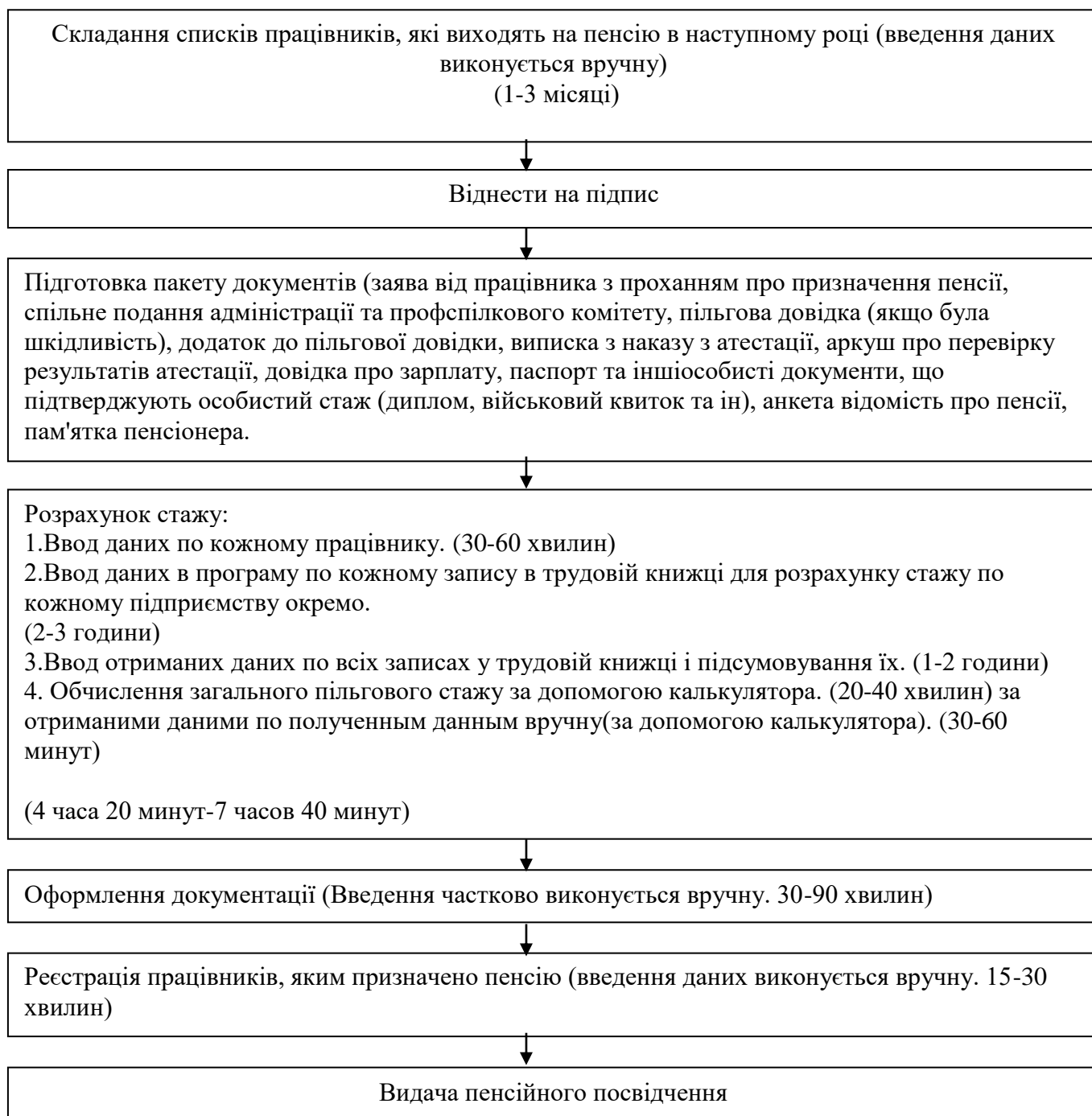


Рисунок 1.1 - Схема виконуваних операцій для нарахування пенсій до автоматизації

## 1.2 Оцінка можливих результатів автоматизації

Розглянемо схему виконуваних операцій для нарахування пенсій після автоматизації. Обов'язки працівника відділу кадрів у цілому не змінюються, але часу на них має витратитися набагато менше. Відповідно на складання списків працівників, які виходять на пенсію в наступному році може витратитися від 2 до 5 хвилин. Так як дані беруться з електронного довідника, який попередньо створений і заповнений, це набагато полегшить роботу працівника відділу кадрів, так як кожен раз він не буде вводити дані, а буде їх брати в довіднику. Після чого списки відносять на підпис директору з кадрів і передаються в пенсійний фонд. Також готується пакет документів - тут все залишається без змін.

Операція розрахунку стажу істотно дозволяє заощадити час. Так як дані про працівників вже будуть внесені в довідники, а на обчислення пільгового та загального стажу має витратитися значно менше часу, тому що це все має бути враховано в програмі. Це все буде займати близько 12-25 хвилин.

На оформлення документації теж буде витратитися менше часу (близько 15-30 хвилин), так як шаблони деяких документів будуть внесені.

Відповідно і на реєстрацію працівників, яким призначено пенсію, буде йтименше часу (7-15 хвилин).

Після чого вони отримають пенсійне посвідчення.

Проаналізувавши отримані вище результати можна полегшити роботу співробітників відділу кадрів за допомогою автоматизації деяких рутинних операцій, на які витрачається досить багато часу. Схему виконуваних операцій для нарахування пенсій з урахуванням автоматизації деяких операцій зображена на рисунку 1.2.

Відповідно після автоматизації час, що витрачається співробітником відділу кадрів значно зменшиться. Тепер співробітник буде витрачати на розрахунок стажу одного працівника: мінімум 34 хвилини, максимум 1 годину 15 хвилин.

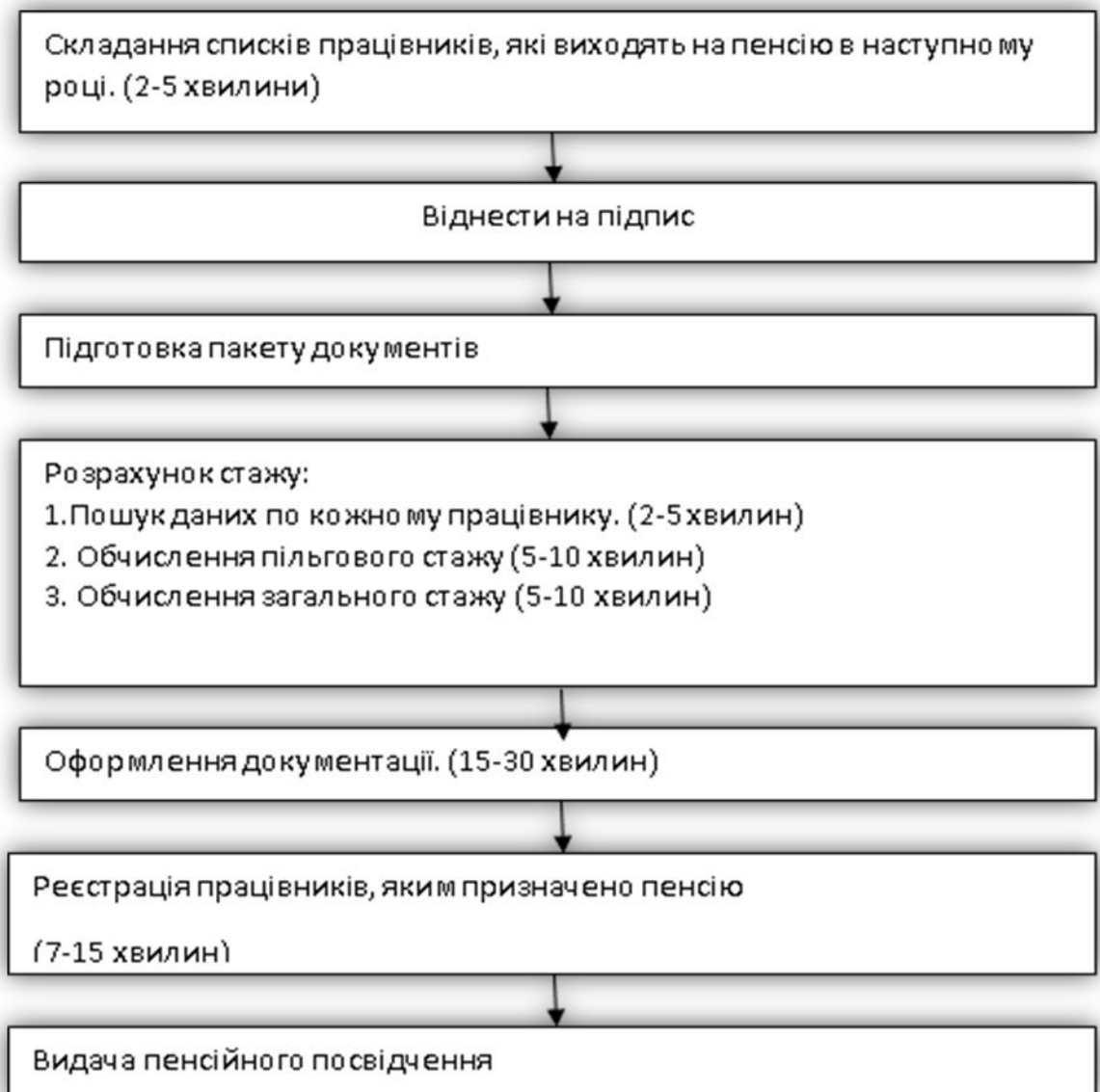


Рисунок 1.2 - Схема виконуваних операцій для нарахування пенсій після автоматизації

### 1.3 Висновки та постановка задачі

Після розгляду двох процесів розрахунку стажу співробітникам, що виходять на пенсію можемо зробити відповідні висновки, які наведені в таблиці 1.1.

Таблиця 1.1 - Витрати часу на виконання операцій для нарахування пенсій до і після автоматизації

Операція	До автоматизації	Після автоматизації
Складання списків працівників, які виходять на пенсію в наступному році	1-3 місяця	2-4 хвилини
Розрахунок стажу:	7-15 години	12-25 хвилин
Введення даних по кожному працівнику	30-60 хвилин	2-5 хвилин
Введення даних у програму по кожному запису в трудовій книжці для розрахунку стажу по кожному підприємству	2-3 години	-----
Введення отриманих даних по всіх записах у трудовій і підсумовування їх	1-2 години	-----
Обчислення загального пільгового стажу	20-40 хвилин	5-10 хвилин
Обчислення загального стажу за отриманими даними	30-60 хвилин	5-10 хвилин
Оформлення документації	30-90 хвилин	15-30 хвилин
Реєстрація працівників, яким призначено пенсію	15-30 хвилин	5-15 хвилин
РАЗОМ	4 години 20 хвилин -7 годин 40 хвилин	34 хвилини -1 година 15 хвилин

З наведеної вище таблиці можна зробити висновки - час що витрачається працівником відділу кадром істотно скоротився на мінімум 3 години 46 хвилин – максимум 6 годин 35 хвилин. І це без врахування економії часу на складання списків працівників, які виходять на пенсію в наступному році. Відповідно процес нарахування пенсій потрібно автоматизувати.

## 1.4 Технічне завдання на розробку програмного забезпечення

### 1.4.1 Технічні вимоги

Цей розділ містить головні характеристики системи, учасників, які використовують систему для виконання основних процесів, а також варіанти використання, які система виконує для відповідних учасників (ролей користувачів).

У даному розділі класифікуємо головні характеристики системи «Подання на призначення пенсії» за трьома категоріями.

Істотні характеристики - обов'язково повинні бути реалізовані. Бажані характеристики - це ті характеристики, без яких система може функціонувати, але бажано, щоб вони також по можливості були реалізовані. Наступні характеристики це ті, для яких існує невизначеність або незрозумілість. Вони не будуть включені в перші версії системи, але можуть бути реалізовані пізніше. У будь-якому випадку цей список не повний, але він відображає найбільш важливі характеристики з точки зору замовника або потенційного користувача (табл.1.2).

Таблиця 1.2 - Варіанти використання

Ім'я варіанту використання	Приоритет*	Номер	Описание
Авторизація в системі	Б	1	Система повинна ідентифікувати користувачів
Створення нового подання	I	1	Система повинна давати користувачеві можливість створювати нові подання
Зміна інформації по існуючому поданню	I	2	Система повинна давати користувачеві можливість проводити зміни інформації за існуючим поданням
Видалення подання	I	3	Система повинна давати користувачеві можливість видаляти подання з системи
Пошук подання	I	4	Система повинна давати користувачеві можливість пошуку подання в системі
Формування документа «Подання»	I	5	Система повинна давати користувачеві можливість автоматично формувати документ «Подання»
Розрахунок стажу	I	6	Система повинна давати користувачеві можливість розрахунку стажу роботи за трудовою книжкою і іншими документами



### **1.4.2 Вимоги до безпеки**

- Необхідно проводити аутентифікацію користувача в системі.
- Необхідно вести протокол робіт користувачів в системі.
- Вся інформація повинна зберігатися на сервері баз даних.

### **1.4.3 Вимоги до програмного забезпечення**

Програмне забезпечення повинно бути написано для використання в операційній системі Microsoft Windows XP або пізнішої, мова програмування Delphi. Інтерфейс користувача повинен бути реалізований російською мовою.

### **1.4.4 Вимоги до технічних засобів**

Розробляємо ПЗ повинне відповідати наступним технічним вимогам:

- програма повинна працювати на IBM-сумісних ПЕОМ з процесором Intel або AMD частотою не менш 800 МГц;
- ПЗ має запускатися на комп'ютерах з ОЗУ не менше 512 Мбайт;
- монітор повинен бути SVGA 17 дюймів і вище;
- жорсткий диск повинен мати обсяг не менше 60 Гбайт.

### **1.4.5 Доступність**

Система повинна бути доступна користувачам як мінімум протягом робочого дня для внесення та коректування інформації.

### **1.4.6 Зручність і простота використання**

Необхідно передбачити створення довідки і спливаючих підказок, що допомагають користувачу в роботі з системою, а так само підготувати інструкцію для користувачів по роботі в системі.

## 2. ВИБІР ТА ОБГРУНТУВАННЯ ЗАСОБІВ РОЗРОБКИ ПЗ

Серед багатьох представників мов програмування здатних підтримувати архітектуру клієнт/сервер і мову SQL-запитів можна виділити наступні:

- Delphi Enterprise 7.0 - Inprise;
- Visual Basic 6.0 Enterprise Edition - Microsoft;
- Visual C++ 6.0 Enterprise Edition - Microsoft;
- PowerBuilder 6.0 - Powersoft.

У таблиці 2.1 представлений порівняльний аналіз деяких засобів розробки додатків.

Таблиця 2.1 - Порівняльний аналіз засобів розробки додатків

Параметр	Delphi Enterprise 7.0	VisualBasic Enterprise Edition 6.0	PowerBuilder Enterprise for Windows 6.0
Компілятор	+	-	-
Оптимізація коду	+	-	-
Інтерпретатор коду	-	+	+
Створення DLL	+	-	-
Використання компонент третіх фірм	+	+	-
Підтримка архітектури клієнт/сервер	+	+	+
Робота з SQL запитам	+	+	+
Мовна підтримка	+	+	+
Підтримка ODBC	+	+	-
Підтримка BDE	+	-	-
Підтримка Native	+	-	+
Візуальна розробка інтерфейсу	+	+	+
Зайняте місце	1	2	3

ПРИМІТКА. «+» - так; «-» - ні.

Для написання програми використана середовище розробки Delphi. Дане середовище розробки використовує мову програмування Object Pascal і надає безліч

можливостей. Вона включає добре продуману і якісно реалізовану бібліотеку класів, надає широкий вибір візуальних компонентів, зручні засоби для створення віконних форм, володіє зрозумілим графічним інтерфейсом підтримує написання крос-платформних додатків і т.д.

Середовище розробки Delphi надає розробнику зручні засоби відлагодження, що володіють великою функціональністю. Це полегшує відладку, дозволяючи зменшити час налагодження та своєчасно виправити помилки, що підвищує надійність розроблюваної програми.

Даний програмний продукт виробляється фірмою Inprise Corporation (колишньою Borland International), яка є автором і виробником BDE. Крім вище сказаного, цей пакет лідирує по числу продажів і має ще одну перевагу в порівнянні з усіма - це наявність відповідної літератури, що робить сам програмний продукт легко доступним для розуміння, вивчення і розробки додатків.

В Delphi вдало поєднуються засоби візуального проектування додатків і оптимізуючий компілятор, чого не можна сказати про інші системи RAD. Наявність у системі компілятора або генерування її виконуваного машинного коду ще не означає, що код, який отримується, є оптимальним. Такі системи, як PowerBuilder і Visual Basic, від самого початку створювалися на основі концепції створення псевдокоду. При виконанні програм, створених за допомогою цих систем, отриманий псевдокод інтерпретувався. З виходом Delphi як компанія Microsoft, так і компанія Powersoft спробували впровадити у свої продукти повноцінні компілятори. Однак проблема полягає в тому, що ні мова Visual Basic, ні мова PowerScript не призначалися для компіляції, тому перетворення їх в машинний код виявилось досить важким завданням.

Якщо порівнювати вищезазначені мови з ObjectPascal, який використовується в системі Delphi, то різницю видно відразу. ObjectPascal завжди був компільованою мовою, і при його розробці були дотримані всі вимоги, виконання яких обов'язково при компіляції та оптимізації. Підсумок вищесказаного такий: Delphi є повноцінним засобом промислової розробки систем клієнт/сервер. Слід зазначити, що у разі

Delphi слова «оптимізує компілятор» не означають «повільний компілятор». В останній версії продукту представлений кращий компілятор з мови Pascal компанії Borland, яка вже протягом багатьох років утримує пальму першості в цій області. Компілятори з Pascal цієї компанії здобули заслужену славу за генерацію виконуваного коду, який економно використовує ресурси комп'ютера і одночасно має високу продуктивність. Компілятор Object Pascal, що використовується в Delphi, не є винятком. Більш того, генератор коду Delphi - це той самий генератор коду, який застосовується компанією Borland в її компіляторах з мови C++. Таким чином, використовуючи Delphi, отримується швидкість програм, написаних на мові C++, не маючи труднощів, пов'язаних з цією мовою програмування. Однак сучасних розробників хвилює не тільки ефективність виконуваного коду. Їм потрібно засіб, який був би, з одного боку, досить потужним і гнучким, щоб виконати завдання, що стоїть перед ним, і, з іншого боку, досить простим і зручним в роботі. Розробники хочуть мати систему, побудовану на принципах об'єктно-орієнтованого програмування. Їм потрібна система, яка генерувала б не тільки виконувані файли у форматі EXE, але і могла створювати файли DLL і драйверів пристроїв. Нарешті їм потрібне середовище для швидкої розробки баз даних, яке не змушувало б їх кожен раз при програмуванні спускатися до ядра СУБД.

Все це і навіть більше надає Delphi. До складу Delphi входить велика бібліотека компонентів, за допомогою якої можна уникнути ручного написання програм, що широко поширене в інших засобах розробки. З іншого боку програміст в будь-який момент може вдатися до низькорівневих асемблерних процедур. Можна створювати додатки у візуальному режимі, просто поміщаючи потрібні компоненти на форму і, разом з тим, зберігаючи доступ до всіх функцій програмного інтерфейсу Windows, системним повідомленнями та процесам. Працюючи в Delphi, програміст може за допомогою натиснення однієї клавіші створити виконуваний файл у форматі EXE, проте при необхідності можна компілювати файли DLL, драйверів пристроїв, а також консольних додатків.

Отже, існує безліч достоїнств, завдяки яким можна виділити Delphi з ряду інших засобів розробки:

- велика бібліотека класів;
- швидкий оптимізуючий компілятор, що генерує машинний код;
- вбудований відладчик;
- простий в освоєнні механізм доступу до баз даних;
- потужне і зручне в роботі середовище розробки;
- наявність інформаційно-технічної літератури про середовище розробки додатків.

При розробці додатків в Delphi програміст має можливість використовувати декілька механізмів, що забезпечують обробку виключних ситуацій. Це і спеціальні оператори мови Object Pascal, і класи, призначені для програмування реакції на помилки. Компілятор Delphi генерує код, який перехоплює будь-які нештатні події, зберігає необхідні дані про стан програми, і видає розробнику об'єкт.

Для роботи з об'єктами виняткових ситуацій існують спеціальні конструкції мови Object Pascal. Вони контролюють виконання операторів, поміщених всередині блоку до ключового слова `except` або `finally`. У разі виникнення виняткової ситуації штатне виконання програми негайно припиняється і управління передається операторам, що йдуть за зазначеними ключовими словами. Якщо в процедурі ці блоки відсутні, управління все одно буде передано найближчому блоку, всередині якого виникла ситуація. Будь-який створений в Delphi додаток повинен забезпечувати обробку можливих виключних ситуацій.

На відміну від інших засобів RAD, система Delphi повнофункціональна - у розпорядженні користувача є стільки можливостей нарощування її потужності, що він ніколи не зіткнеться з завданням, яке виявиться «не під силу» Delphi.

Крім того, Delphi, починаючи з моменту своєї появи на ринку ПЗ підтримує і просуває сервера БД InterBase, тому що на даний момент це продукти однієї фірми, також слід зазначити і той факт, що система управління та адміністрування InterBaseServer 6.0 IBConsole написана на Delphi.

### 3. РОЗРОБКА ПРИКЛАДНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

#### 3.1 Структура програмного забезпечення

Аналізуючи технічне завдання, вивчаючи структуру і діяльність працівників відділу кадрів можна побудувати структуру програмного забезпечення. Так як робота персоналу відділу кадрів передбачає роботу з великим потоком даних, який не тільки редагується, але і доповнюється новими даними, то результати обчислень, паспортні та інші дані зручно зберігати в базі даних і файлах даних, створених за допомогою структурного типу даних - запису. Запис - це структура даних, доступ до елементів якої здійснюється по імені (за назвою елементу). Елементи запису повинні мати різний тип, що відповідає елементу, який там буде зберігатися. Так само для зберігання різного роду даних повинні створюватися довідники, в яких буде зберігатися вся необхідна інформація. Фактично поля запису будуть зберігати в основному важливу інформацію про кожного працівника, яка може служити для здійснення пошуку даних для нарахування пенсії. Цими даними можуть бути, наприклад паспортні дані, ідентифікаційний код, адреса та інші. Також повинні зберігатися дані про різні підприємства. Для кожної людини буде створений окремий відповідний запис, з відповідною кількістю і типом полів. Головне завдання, яке доводиться вирішувати працівникам відділу кадрів - це збір всіх відповідних документів на кожного співробітника, що виходить на пенсію, а також розрахунок його стажу. Програмне забезпечення дозволить без особливих зусиль і витрат часу створювати списки працівників, які виходять на пенсію в наступному році, проводити розрахунок стажу та пошук даних. Перед описом компонент і модулів, які використовуються у програмі, потрібно побудувати абстрактну модель предметної області (рис. 3.1) [10].

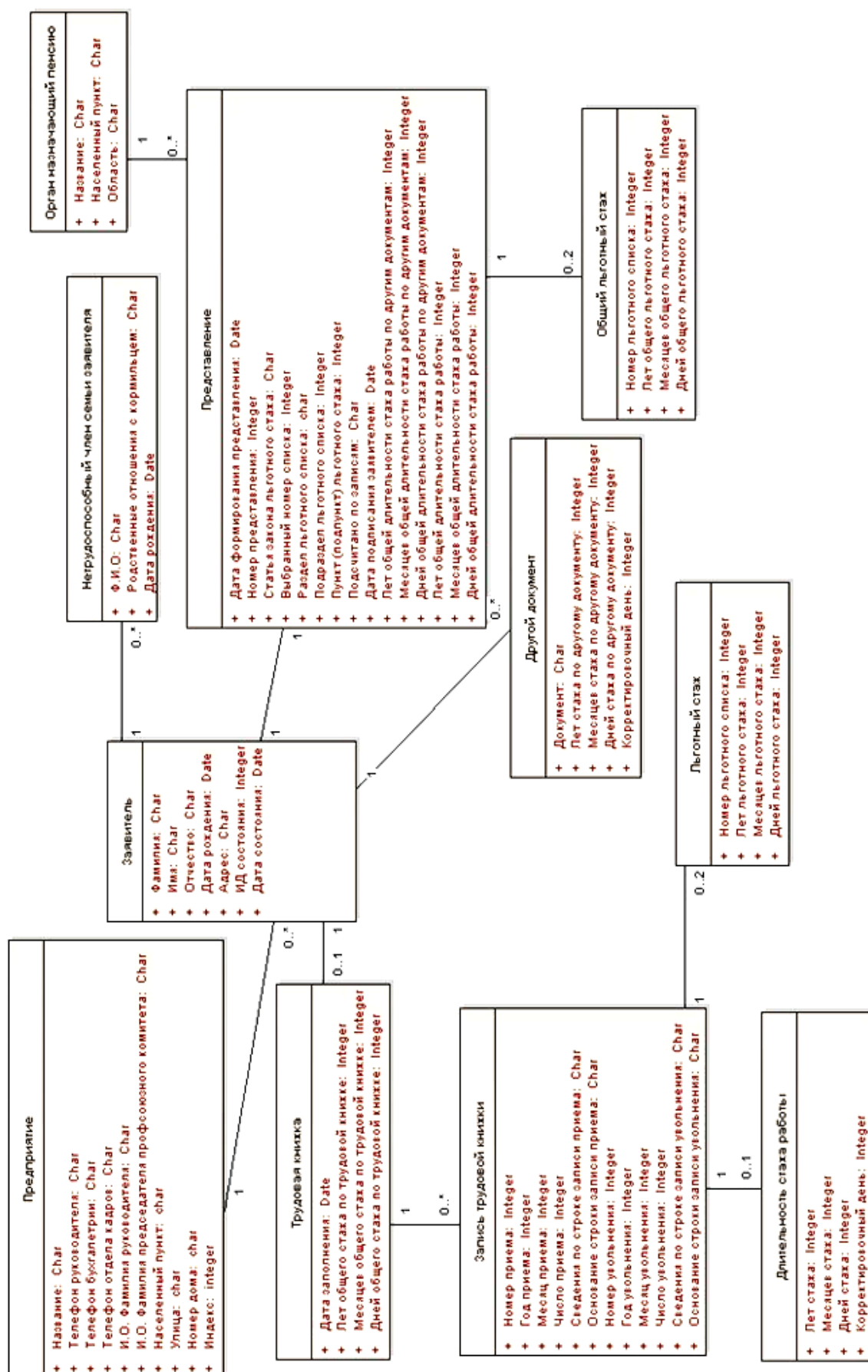


Рисунок 3.1 – Абстрактна модель предметної області



Нижче наведено опис предметної області.

Старший інспектор групи з оформлення пенсії (Відділ кадрів) готує і передає в орган, що призначає пенсію, пакет документів, необхідних для призначення пенсії працівникам підприємства, які одержали право на пенсію. У пакет документів входить Представлення для призначення пенсії. У цей документ працівники групи вносять інформацію про підприємство, заявника, непрацездатних членів сім'ї заявника. Крім того переписують інформацію з трудової книжки та інших документів, і підраховують стаж роботи та пільговий стаж, якщо такий є. Після внесення всієї необхідної інформації в Подання на призначення пенсії, документ підписується у Директора з кадрів, Голови профспілкового комітету і працівника, на якого готується подання. Після чого, Представлення для призначення пенсії передається до Пенсійного фонду разом з іншими документами.

Предметна область, яка представлена на рисунку 3.1 має наступні характеристики: орган, що призначає пенсію; підприємство; заявник; непрацездатні члени сім'ї заявника; інший документ; пільговий стаж, загальний пільговий стаж; трудова книжка; тривалість стажу роботи;

представлення; запис трудової книжки.

Орган, що призначає пенсію має характеристики:

- назва органу, що призначає пенсію (ЕД5119);
- місто органу, що призначає пенсію (ЕД5120);
- область органу, що призначає пенсію (ЕД5121).

З органом, що призначає пенсію, може бути пов'язано декілька Подань (0 ..\*).

Підприємство має характеристики:

- назва підприємства (ЕД5127);
- індекс (ЕД5129);
- населений пункт (ЕД5130);
- вулиця (ЕД5131);
- номер будинку (ЕД5132);
- телефон керівника (ЕД5134);

- телефон бухгалтерії (ЕД5135);
- телефон відділу кадрів (ЕД5136);
- І.Б. Прізвище керівника (ЕД5190);
- І.Б. Прізвище голови профкому (ЕД5191).

У одного підприємства може бути кілька заявників (0 ..\*).

Заявник має характеристики:

- Прізвище заявника (ЕД5139);
- Ім'я заявника (ЕД5140).
- По батькові заявника (ЕД5141).
- Дата народження заявника (ЕД5142).
- Адреса заявника (ЕД5143).
- Стан заявника (ЕД5318).
- Дата початку стану заявника (ЕД5319).

Заявник пов'язаний з певним Підприємством (..1). У заявника можуть бути Непрацевдатні члени сім'ї (0 ..\*). Заявник пов'язаний з певним Поданням (1). У заявника можуть бути Інші документи (0 ..\*), які підтверджують стаж.У заявника може бути Трудова книжка (0 .. 1).

Непрацевдатні члени сім'ї заявника мають характеристики:

- П.І.Б. члена сім'ї (ЕД5145).
- Сімейні стосунки з годувальником (ЕД5146).
- Дата народження (ЕД5147).

Непрацевдатні члени сім'ї належать Заявнику (1).

Інший документ має характеристики:

- Документ, що підтверджує стаж (ЕД5165).
- Років стажу за іншими документами (ЕД5167).
- Місяців стажу за іншими документами (ЕД5168).
- Днів стажу за іншими документами (ЕД5169).
- Номер запису за порядком (ЕД5321).
- Коректувальних днів (ЕД5212).

Пільговий стаж має характеристики:

- Номер пільгового стажу (ЕД5185).
- Років пільгового стажу (ЕД5161).
- Місяців пільгового стажу (ЕД5162).
- Днів пільгового стажу (ЕД5163).
- Пільговий стаж пов'язаний із записом у трудовій книжці (..1).
- Загальний пільговий стаж має характеристики:
- Номер пільгового стажу (ЕД5185).
- Років загального пільгового стажу (ЕД5312).
- Місяців загального пільгового стажу (ЕД5313).
- Днів загального пільгового стажу (ЕД5314).
- Загальний пільговий стаж пов'язаний з Поданням (..1).

Трудова книжка має характеристики:

- Дата заповнення трудової книжки (ЕД5149).
- Років загального стажу роботи за трудовою книжкою (ЕД5309).
- Місяців загального стажу роботи за трудовою книжкою (ЕД5310).
- Днів загального стажу роботи за трудовою книжкою (ЕД5311).

Трудова книжка пов'язана з Заявником (..1).

Трудова книжка може мати запис в трудовій книжці (0 ..\*).

Тривалість стажу роботи книжка має характеристики:

- Років стажу (ЕД5157).
- Місяців стажу (ЕД5158).
- Днів стажу (ЕД5159).
- Коректувальних днів (ЕД5212).

Тривалість стажу роботи відноситься до Запису в трудовій книжці (..1).

Подання має характеристики:

- Дата формування подання (ЕД5236).
- Номер подання (ЕД5237).
- Підраховано за записами (ЕД5192).

- Дата підписання (ЕД5181).
- Стаття закону (ЕД5184).
- Обраний номер списку (ЕД5320).
- Розділ списку (ЕД5186).
- Пункт (підпункт) списку (ЕД5303).
- Підрозділ списку (ЕД5188).
- Років загальної тривалості стажу роботи за іншими документами (ЕД5306).
- Місяців загальної тривалості стажу роботи за іншими документами (ЕД5307).
- Днів загальної тривалості стажу роботи за іншими документами (ЕД5308).
- Років загального стажу (ЕД5315).
- Місяців загального стажу (ЕД5316).
- Днів загального стажу (ЕД5317).

У подання є Заявник (..1).

Подання пов'язано з Органом, що призначає пенсію (..1).

У подання може бути Загальний пільговий стаж (0 .. 2).

Запис трудової книжки має характеристики:

- Номер прийому (ЕД5304).
- Рік прийому (ЕД5151).
- Місяць прийому (ЕД5152).
- Число прийому (ЕД5153).
- Відомості по рядку запису прийому (ЕД5154).
- Підстава рядка запису прийому (ЕД5155).
- Номер звільнення (ЕД5305).
- Рік звільнення (ЕД5195).
- Місяць звільнення (ЕД5196).
- Число звільнення (ЕД5197).

- Відомості по рядку запису звільнення (ЕД5154).
- Підстава рядка запису звільнення (ЕД5155).
- Конкретний запис у трудовій книжці може бути тільки в одній Трудовий

книжці (..1).

У запису трудової книжки може бути Пільговий стаж (0 .. 2).

У запису трудової книжки може бути Тривалість стажу роботи (0 .. 1).

### **3.2 Детальні вимоги за варіантами використання**

Кожна вимога має унікальний ідентифікатор записаний в такій формі:  
<пріоритет\_ВВ><номер\_ВВ> - <номер\_детальної\_вимоги>.

#### **Виконання авторизації в системі:**

Нормальний напрямок:

Система запрошує логін і пароль. Користувач вводить логін і пароль.

Ця система відображає головну екранну форму

Розширення:

- 1) Система повідомляє, що введений логін і пароль невірний.
- 2) Повернення до пункту 1.

#### **Створення нового подання:**

Користувач вносить наступну інформації за новим поданням:

Заявник:

- Прізвище заявника (ЕД5139);
- Ім'я заявника (ЕД5140);
- По батькові заявника (ЕД5141);
- Дата народження заявника (ЕД5142);
- Адреса заявника (ЕД5143);
- Стан заявника (ЕД5318);
- Дата початку стану заявника (ЕД5319).

- Непрацевдатні члени сім'ї заявника:
- П.І.Б. члена сім'ї (ЕД5145);
- Сімейні стосунки з годувальником (ЕД5146);
- Дата народження (ЕД5147).

Трудова книжка заявника:

- Дата заповнення трудової книжки (ЕД5149)

Запис трудової книжки:

- Рік прийому (ЕД5151);
- Місяць прийому (ЕД5152);
- Число прийому (ЕД5153)
- Відомості по рядку запису прийому (ЕД5154);
- Підстава рядка запису прийому (ЕД5155);
- Рік звільнення (ЕД5195);
- Місяць звільнення (ЕД5196);
- Число звільнення (ЕД5197).
- Відомості по рядку запису звільнення (ЕД5154);
- Підстава рядка запису звільнення (ЕД5155);

Тривалість стажу роботи:

- Коректувальних днів (ЕД5212).

Пільговий стаж:

- Номер пільгового стажу (ЕД5185);
- Років пільгового стажу (ЕД5161);
- Місяців пільгового стажу (ЕД5162);
- Днів пільгового стажу (ЕД5163).

Інший документ:

- Документ, що підтверджує стаж (ЕД5165);
- Років стажу за іншими документами (ЕД5167);
- Місяців стажу за іншими документами (ЕД5168);
- Днів стажу за іншими документами (ЕД5169);

- Коректувальних днів (ЕД5212).

Подання:

- Підраховано за записами (ЕД5192);
- Дата підписання (ЕД5181);
- Стаття закону (ЕД5184);
- Обраний номер списку (ЕД5320);
- Розділ списку (ЕД5186);
- Пункт (підпункт) списку (ЕД5303);
- Підрозділ списку (ЕД5188).

**Система автоматично вносить наступну інформацію:**

Орган, що призначає пенсію (СД5118):

- Назва органу, що призначає пенсію (ЕД5119);
- Місто органу, що призначає пенсію (ЕД5120) (не тільки Сєвєродонецьк);
- Область органу, що призначає пенсію (ЕД5121).

Підприємство:

- Назва підприємства (ЕД5127);
- Індекс (ЕД5129);
- Населений пункт (ЕД5130);
- Вулиця (ЕД5131);
- Номер будинку (ЕД5132);
- Телефон керівника (ЕД5134);
- Телефон бухгалтерії (ЕД5135);
- Телефон відділу кадрів (ЕД5136);
- І.О. Прізвище керівника (ЕД5190);
- І.О. Прізвище голови профкому (ЕД5191).

Тривалість стажу роботи:

- Років стажу (ЕД5157);
- Місяців стажу (ЕД5158);

- Днів стажу (ЕД5159).

**Подання:**

- Дата формування подання (ЕД5236);
- Номер подання (ЕД5237);
- Років загальної тривалості стажу роботи за іншими документами (ЕД5306);
- Місяців загальної тривалості стажу роботи за іншими документами (ЕД5307);
- Днів загальної тривалості стажу роботи за іншими документами (ЕД5308);
- Років загального стажу (ЕД5315);
- Місяців загального стажу (ЕД5316);
- Днів загального стажу (ЕД5317).

**Загальний пільговий стаж:**

- Номер пільгового стажу (ЕД5185);
- Років загального пільгового стажу (ЕД5312);
- Місяців загального пільгового стажу (ЕД5313);
- Днів загального пільгового стажу (ЕД5314).

**Трудова книжка:**

- Років загального стажу роботи за трудовою книжкою (ЕД5309);
- Місяців загального стажу роботи за трудовою книжкою (ЕД5310);
- Днів загального стажу роботи за трудовою книжкою (ЕД5311).

Необхідно передбачити можливість ручної зміни інформації занесеною системою і перерахунку системою розрахункових значень.

**Розрахунок стажу:**

При розрахунку керуватися таким правилом:

Різниця між звільненням і прийомом (ЕД5215) = Дата прийому (ЕД5151.ЕД5152.ЕД5153) - (Дата звільнення (ЕД5195.ЕД5196.ЕД5197)



1) Розрахувати тривалість стажу роботи по внесеній Даті прийому та Даті звільнення.

Якщо Різниця між звільненням і прийомом (ЕД5215) <0то:

Загальна тривалість стажу роботи (ЕД5157.ЕД5158.ЕД5159) = Дата звільнення (ЕД5195.ЕД5196.ЕД5197) - Дата прийому (ЕД5151.ЕД5152.ЕД5153) Різниця між звільненням і прийомом (ЕД5215) інакше:

Загальна тривалість стажу роботи (СД5156 ЕД.ЕД.ЕД) = Дата звільнення (ЕД5195.ЕД5196.ЕД5197) - Дата прийому (ЕД5151.ЕД5152.ЕД5153)

Необхідно передбачити можливість відмови користувача від розрахунків тривалості стажу роботи з конкретного запису.

2) Розрахувати Загальну тривалість стажу роботи за іншими документами щодо внесеної тривалості стажу роботи за іншими документами.

Загальна тривалість стажу роботи за іншими документами (ЕД5306.ЕД5307.ЕД5308) =  $\Sigma$  (тривалість стажу роботи за іншими документами (ЕД5167.ЕД5168.ЕД5169) коректувальних днів (ЕД5212))

3) Розрахувати Загальний стаж роботи за трудовою книжкою по розрахованій тривалості стажу роботи на кожному робочому місці.

Загальний стаж роботи за трудовою книжкою (ЕД5309.ЕД5310.ЕД5311) =  $\Sigma$  (тривалість стажу роботи (ЕД5157.ЕД5158.ЕД5159) коректувальних днів (ЕД5212))

4) Розрахувати Загальний пільговий стаж згідно кожного номера списку за введеною тривалістю пільгового стажу роботи.

Розрахунки необхідно проводити по кожному списку окремо. Елемент Номер списку (ЕД5185) на сьогоднішній день може приймати два значення (1 і 2).

Загальний пільговий стаж (ЕД5312.ЕД5313.ЕД5314) =  $\Sigma$  (тривалість пільгового стажу роботи (ЕД5161.ЕД5162.ЕД5163))

Отримані результати необхідно видавати по кожному списку окремо.

5) Розрахувати Загальний стаж роботи за розрахованим Загальним стажем роботи за трудовою книжкою і Загальною тривалістю стажу роботи за іншими документами.

Загальний стаж роботи (ЕД5315.ЕД5316.ЕД5317) = Загальний стаж роботи за трудовою книжкою (ЕД5309.ЕД5310.ЕД5311) Загальна тривалість стажу роботи за іншими документами (ЕД5306.ЕД5307.ЕД5308)

#### **Зміна інформації за поданням:**

Система повинна дозволяти проводити зміни всіх внесених реквізитів за поданням.

#### **Видалення подання:**

Система повинна володіти функціональністю видалення вибраного подання.

#### **Пошук подання:**

Система повинна дозволяти проводити пошук подання за наступними реквізитами:

Подання:

- Номер подання (ЕД5237);
- Дата формування подання (ЕД5236).

Заявник:

- Прізвище заявника (ЕД5139);
- Ім'я заявника (ЕД5140);
- По батькові заявника (ЕД5141);
- Дата народження заявника (ЕД5142).

#### **Формування документа «Подання»**

Документ "Подання для призначення пенсії" має форму, представлену в Додатку А і складається з наступних структур і елементів даних:

<P1> - Орган, що призначає пенсію - Назва органу, що призначає пенсію (ЕД5119);

<P2> - Орган, що призначає пенсію - Місто органу, що призначає пенсію (ЕД5120);

<P3> - Орган, що призначає пенсію - Область органу, що призначає пенсію (ЕД5121);

<P4> - Подання - Дата формування подання (ЕД5236);

- <P5> - Подання - Номер подання (ЕД5237);
- <P6> - Підприємство - Назва підприємства (ЕД5127);
- <P7> - Підприємство - Індекс (ЕД5129);
- <P8> - Підприємство - Населений пункт (ЕД5130);
- <P9> - Підприємство - Вулиця (ЕД5131);
- <P10> - Підприємство - Номер будинку (ЕД5132);
- <P11> - Підприємство - Телефон керівника (ЕД5134);
- <P12> - Підприємство - Телефон бухгалтерії (ЕД5135);
- <P13> - Підприємство - Телефон відділу кадрів (ЕД5136);
- <P14> - Заявник - Прізвище заявника (ЕД5139);
- <P15> - Заявник - Ім'я заявника (ЕД5140);
- <P16> - Заявник - По-батькові заявника (ЕД5141);
- <P17> - Заявник - Дата народження заявника (ЕД5142);
- <P18> - Заявник - Адреса заявника (ЕД5143);
- <P19> - Порядковий номер рядка в таблиці документа;
- <P20> - Непрацевдатні члени сім'ї заявника - П.І.Б. члена сім'ї (ЕД5145);
- <P21> - Непрацевдатні члени сім'ї заявника - Сімейні відносини з годувальником (ЕД5146);
- <P22> - Непрацевдатні члени сім'ї заявника - Дата народження (ЕД5147);
- <P23> - Стаж роботи за трудовою книжкою заявника - Дата заповнення трудової книжки (ЕД5149);
- <P24> - Порядковий номер рядка в таблиці документа. Може приймати значення елементів:
- Інший документ - Номер запису по порядку (ЕД5321);
  - Запис трудової книжки - Номер прийому (ЕД5304);
  - Запис трудової книжки - Номер звільнення (ЕД5305).
- <P25> - Запис за трудової книжки - Рік прийому (ЕД5151);
- <P26> - Запис за трудової книжки - Місяць прийому (ЕД5152);
- <P27> - Запис потрудової книжки - Число прийому (ЕД5153);

- <P28> - Запис за трудової книжки - Відомості по рядку запису прийому (ЕД5154);
- <P28a> - Запис за трудової книжки - Відомості по рядку запису звільнення (ЕД5154);
- <P29> - Запис за трудової книжки - Підстава рядка запису прийому (ЕД5155);
- <P29> - Запис за трудової книжки - Підстава рядка запису звільнення (ЕД5155);
- <P30> - Тривалість стажу роботи - Років стажу (ЕД5157);
- <P31> - Тривалість стажу роботи - Місяців стажу (ЕД5158);
- <P32> - Тривалість стажу роботи - Днів стажу (ЕД5159);
- <P33> - Тривалість стажу роботи - коректувальних днів (ЕД5212);
- <P33a> - Інший документ - коректувальних днів (ЕД5212);
- <P34> - Пільговий стаж - Років пільгового стажу (ЕД5161);
- <P34a> - Пільговий стаж - Місяців пільгового стажу (ЕД5162);
- <P34б> - Пільговий стаж - Днів пільгового стажу (ЕД5163);
- <P35> - Запис трудової книжки - Рік звільнення (ЕД5195);
- <P36> - Запис трудової книжки - Місяць звільнення (ЕД5196);
- <P37> - Запис трудової книжки - Число звільнення (ЕД5197);
- <P38> - Інший документ - Документ, що підтверджує стаж (ЕД5165);
- <P39> - Інший документ - Років стажу за іншими документами (ЕД5167);
- <P39a> - Представлення - Років загальної тривалості стажу роботи за іншими документами (ЕД5306);
- <P40> - Інший документ - Місяців стажу за іншими документами (ЕД5168);
- <P40a> - Представлення - Місяців загальної тривалості стажу роботи за іншими документами (ЕД5307);
- <P41> - Інший документ - Днів стажу за іншими документами (ЕД5169);
- <P41a> - Представлення - Днів загальної тривалості стажу роботи за іншими документами (ЕД5308);
- <P42> - Подання - Стаття закону (ЕД5184);

<P43> - Подання - Обраний номер списку (ЕД5320);

<P44> - Подання - Розділ списку (ЕД5186);

<P45> - Подання - Підрозділ списку (ЕД5188);

<P46> - Подання - Пункт (підпункт) списку (ЕД5303);

<P47> - Загальний пільговий стаж - Років загального пільгового стажу (ЕД5312);

<P47а> - Загальний пільговий стаж - Місяців загального пільгового стажу (ЕД5313);

<P47б> - Загальний пільговий стаж - Днів загального пільгового стажу (ЕД5314);

<P48> - Подання - Підраховано за записами (ЕД5192);

<P49> - Підприємство - І.Б. Прізвище керівника (ЕД5190);

<P50> - Підприємство - І.Б. Прізвище голови профкому (ЕД5191);

<P51> - Подання - Дата підписання (ЕД5181);

<P52> - Пільговий стаж - Номер пільгового списку (ЕД5185);

<P53> - Заявник - Дата початку стану заявника (ЕД5319). Заповнюється залежно від стану заявника (ЕД5318):

- якщо стан заявника 1 - «Інвалідність», то Дата початку стану заявника (ЕД5319) вноситься в першому рядку;

- якщо стан заявника 2 - «Смерть годувальника», то Дата початку стану заявника (ЕД5319) вноситься в другому рядку;

- якщо стан заявника 3 - «Припинення роботи, що дає право на пенсію за вислугоюроків» та Дата початку стану заявника (ЕД5319);

<P54> - Представлення - Років загального стажу (ЕД5315);

<P54а> - Представлення - Місяців загального стажу (ЕД5316);

<P54б> - Представлення - Днів загального стажу (ЕД5317);

<P55> - Трудова книжка - Років загального стажу роботи за трудовою книжкою (ЕД5309);

<P55a> - Трудова книжка - Місяців загального стажу роботи за трудовою книжкою (ЕД5310);

<P55б> - Трудова книжка - Днів загального стажу роботи за трудовою книжкою (ЕД5311).

### 3.3 Опис бази даних

Дана база даних призначена для зберігання інформації про подання, заявника, підприємства і т.д. А також для групування всіх даних у таблиці і встановлення зв'язків між ними [12]. Всі необхідні дані описані в таблицях 3.1 – 3.12.

Таблиця 3.1 – Пенсійний орган () (додаток В, оператор 1953-2008)

PK	Name	Найменування	Тип	NotNull	Длина
True	PENS_ORGAN_ID	ПК пенсійний орган	integer	True	
False	NAZVANIE	Назва	varchar	True	100
False	NAS_PUNKT	Населений пункт	varchar	True	50
False	OBLAST	Область	varchar	True	50

#### Обмеження

Найменування	Тип	Стовпець
PK_PENS_ORGAN	Public	PENS_ORGAN_ID

#### Зв'язок з іншими таблицями:

Стовпець	Асоціація
(PENS_ORGAN_ID =	0..* PREDSTAV.FK_PREDSTAV_PENS_ORGAN
PENS_ORGAN_ID)	1 PENS_ORGAN.PK_PENS_ORGAN

Таблиця 3.2 – Підприємство (PREDPR) (додаток В, оператор 2010-2100)

PK	Name	Найменування	Тип	NotNull	Длина
True	PREDPR_ID	ПК підприємство	integer	True	
False	NAZVANIE	Назва	varchar	True	100
False	INDEKS	Індекс	integer	False	
False	NAS_PUNKT	Населений пункт	varchar	True	50
False	ULICA	Вулиця	varchar	False	50
False	DOM	Номер будинку	varchar	False	4
False	TEL_RUK	Телефон керівника	varchar	False	17
False	TEL_BUH	Телефон бухгалтерії	varchar	False	17
False	TEL_O_K	Телефон відділу кадрів	varchar	False	17
False	RUKOVODITEL	І.Б. Прізвище керівника	varchar	False	50
False	PROFSOUZ	І.Б. Прізвище голови профспілкового комітету	varchar	False	50

## Обмеження:

Найменування	Тип	Стовпець
PK_PREDPR	Public	PREDPR_ID

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(PREDPR_ID = PREDPR_ID)	0..* ZAJAVITEL.FK_ZAJAVITEL_PREDPR 1 PREDPR.PK_PREDPR

Таблиця 3.3 – Заявник (ZAJAVITEL) (додаток В, оператор 111-120)

PK	Name	Найменування	Тип	Not Null	Длина
True	ZAJAVITEL_ID	ПК Заявник	integer	True	
False	FAMILIJA	Прізвище	varchar	True	50
False	IMJA	Ім'я	varchar	True	50
False	OTCHESTVO	Побатькові	varchar	True	50
False	DATA_ROJD	Дата народження	date	True	

## Продовження таблиці 3.3

False	ADRES	Адреса	varchar	True	100
False	DATE_SOSTOYANIE	Дата стану	date	False	
False	SOST_ZAJAVITEL_ID	ПК стану заявника	integer	False	
False	PREDPR_ID	ПК підприємство	integer	False	

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(ZAJAVITEL_ID = ZAJAVITEL_ID)	0..1 TRUD_KNIGA.FK_TRUD_KNIGA_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL
(SOST_ZAJAVITEL_ID = SOST_ZAJAVITEL_ID)	0..* ZAJAVITEL.FK_ZAJAVITEL_SOST_ZAJAVITEL 1 SOST_ZAJAVITEL.PK_SOST_ZAJAVITEL
(ZAJAVITEL_ID = ZAJAVITEL_ID)	1 PREDSTAV.FK_PREDSTAV_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL
(PREDPR_ID = PREDPR_ID)	0..* ZAJAVITEL.FK_ZAJAVITEL_PREDPR 1 PREDPR.PK_PREDPR
(ZAJAVITEL_ID = ZAJAVITEL_ID)	0..* OTHER_DOC.FK_OTHER_DOC_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL
(ZAJAVITEL_ID = ZAJAVITEL_ID)	0..* NETRUD.FK_NETRUD_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL

## Обмеження

Найменування	Тип	Стовпець
PK_ZAJAVITEL	Public	ZAJAVITEL_ID
FK_ZAJAVITEL_PREDPR	Public	PREDPR_ID
FK_ZAJAVITEL_SOST_ZAJAVITEL	Public	SOST_ZAJAVITEL_ID

Таблиця 3.4 – Состояние заявителя (SOST\_ZAJAVITEL) (додаток В, оператор 471-476)

PK	Name	Найменування	Тип	NotNull	Довжина
True	SOST_ZAJAVITEL_ID	ПК стан заявника	integer	True	
False	SOSTOYANIE	Стан заявника	varchar	True	50



## Обмеження:

Найменування	Тип	Стовпець
PK_SOST_ZAJAVITEL	Public	SOST_ZAJAVITEL_ID

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(SOST_ZAJAVITEL_ID = SOST_ZAJAVITEL_ID)	0..* ZAJAVITEL.FK_ZAJAVITEL_SOST_ZAJAVITEL 1 SOST_ZAJAVITEL.PK_SOST_ZAJAVITEL

Таблиця 3.5 – Інші документи (OTHER\_DOC) (додаток В, оператор 700-775)

PK	Name	Найменування	Тип	NotNull	Длина
True	OTHER_DOC_ID	ПК інший документ	integer	True	
False	N_DOC	Порядковий номер запису	integer	True	
False	DOCUMENT	Документ	varchar	True	100
False	LET	Років стажу по іншому документу	integer	False	
False	MES	Місяців стажу за іншим документом	integer	False	
False	DNEJ	Днів стажу за іншим документом	integer	False	
False	KOR_DAY	Коректувальних днів	integer	False	
False	ZAJAVITEL_ID	ПК заявник	integer	True	

## Обмеження:

Найменування	Тип	Стовпець
PK_OTHER_DOC	Public	OTHER_DOC_ID
FK_OTHER_DOC_ZAJAVITEL	Public	ZAJAVITEL_ID

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(ZAJAVITEL_ID = ZAJAVITEL_ID)	0..* OTHER_DOC.FK_OTHER_DOC_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL

Таблиця 3.6 – Непрацевдатні члени сім'ї (NETRUD) (додаток В, оператор 1892-1951)

PK	Name	Найменування	Тип	NotN	Довжин а
True	NETRUD_ID	ПК непрацевдатний член сім'ї	integer	True	
False	F_I_O	П.І.Б.	varchar	True	100
False	RODSTVEN_SVAZ	Родинні відносини з годувальником	varchar	True	150
False	DATA_ROJD	Дата народження	date	True	
False	ZAJAVITEL_ID	ПК заявник	integer	True	

Обмеження:

Найменування	Тип	Стовпець
PK_NETRUD	Public	NETRUD_ID
FK_NETRUD_ZAJAVITEL	Public	ZAJAVITEL_ID

Зв'язок з іншими таблицями:

Стовпець	Асоціація
(ZAJAVITEL_ID = ZAJAVITEL_ID)	0..* NETRUD.FK_NETRUD_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL

Таблиця 3.7 - Подання (PREDSTAV) (додаток В, оператор 122-698)

PK	Name	Найменування	Тип	Not Null	Длина
True	PREDSTAV_ID	ПК подання	integer	True	
False	DATE_PREDST	Дата формування подання	date	False	
False	NOMER_PREDST	Номер подання	integer	True	
False	LGOT_STATJA_ZAKONA	Стаття закону пільгового стажу	varchar	False	50
False	LGOT_NOMER_SPISKA	Обраний номер списку	integer	False	
False	LGOT_RAZDEL_SPISKA	Розділ пільгового списку	varchar	False	50
False	LGOT_PODRAZDEL_SPISKA	Підрозділ пільгового списку	integer	False	
False	LGOT_PUNKT_STAJA	Пункт (підпункт) пільгового стажу	integer	False	
False	LGOT_PODS_PO_ZAPIS	Підраховано за записами	varchar	False	50
False	DATA_PODPISANIA	Дата підписання заявником	date	False	

## Продовження таблиці 3.7

PK	Name	Найменування	Тип	Not Null	Длина
False	SUM_OTHER_LET	Років загальної тривалості стажу роботи за іншими документами	integer	False	
False	SUM_OTHER_MES	Місяців загальної тривалості стажу роботи за іншими документами	integer	False	
False	SUM_OTHER_DAY	Днів загальної тривалості стажу роботи за іншими документами	integer	False	
False	SUM_LET	Років загальної тривалості стажу	integer	False	
False	SUM_MES	Місяців загальної тривалості стажу	integer	False	
False	SUM_DAY	Днів загальної тривалості стажу	integer	False	
False	PENS_ORGAN_ID	ПК Пенсійний орган	integer	False	
False	SUM_MES	ПК заявник	integer	True	

## Обмеження:

Найменування	Тип	Стовпець
PK_PREDSTAV	Public	PREDSTAV_ID
FK_PREDSTAV_PENS_ORGAN	Public	PENS_ORGAN_ID
FK_PREDSTAV_ZAJAVITEL	Public	ZAJAVITEL_ID

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(PREDSTAV_ID = PREDSTAV_ID)	0..2 SUM_LGOT_STAJ.FK_SUM_LGOT_STAJ_PREDSTAV 1 PREDSTAV.PK_PREDSTAV
(ZAJAVITEL_ID = ZAJAVITEL_ID)	1 PREDSTAV.FK_PREDSTAV_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL
(PENS_ORGAN_ID = PENS_ORGAN_ID)	0..* PREDSTAV.FK_PREDSTAV_PENS_ORGAN 1 PENS_ORGAN.PK_PENS_ORGAN

Таблиця 3.8 – Сума пільгового стажу (SUM\_LGOT\_STAJ) (додаток В, оператор 1695-1800)

PK	Name	Найменування	Тип	NotNull
True	SUM_LGOT_STAJ_ID	ПК сума пільгового стажу	integer	True
False	NOMER_LGOTNOGO_SPISKA	Номер пільгового списку	integer	True
False	SUM_LET_LGOT_STAJ	Років загального пільгового стажу	integer	True
False	SUM_MES_LGOT_STAJ	Місяців загального пільгового стажу	integer	True
False	SUM_DAY_LGOT_STAJ	Днів загального пільгового стажу	integer	True
False	PREDSTAV_ID	ПК Представлення	integer	True

Обмеження:

Найменування	Тип	Стовпець
PK_SUM_LGOT_STAJ	Public	SUM_LGOT_STAJ_ID
FK_SUM_LGOT_STAJ_PREDSTAV	Public	PREDSTAV_ID

Зв'язок з іншими таблицями:

Стовпець	Асоціація
(PREDSTAV_ID = PREDSTAV_ID)	0..2 SUM_LGOT_STAJ.FK_SUM_LGOT_STAJ_PREDSTAV 1 PREDSTAV.PK_PREDSTAV

Таблиця 3.9 – Трудова книжка (TRUD\_KNIGA) (додаток В, оператор 1094-1889)

PK	Name	Найменування	Тип	NotNull
True	TRUD_KNIGA_ID	ПК трудова книжка	integer	True
False	DATA_ZAPOLN	Дата заповнення	date	True
False	SUM_LET	Років загального стажу за трудовою книжкою	integer	False
False	SUM_MES	Місяців загального стажу за трудовою книжкою	integer	False
False	SUM_DAY	Днів загального стажу за трудовою книжкою	integer	False
False	ZAJAVITEL_ID	ПК заявник	integer	True

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(TRUD_KNIGA_ID = TRUD_KNIGA_ID)	0..* ZAPIS_TRUD.FK_ZAPIS_TRUD_TRUD_KNIGA 1 TRUD_KNIGA.PK_TRUD_KNIGA
(ZAJAVITEL_ID = ZAJAVITEL_ID)	0..1 TRUD_KNIGA.FK_TRUD_KNIGA_ZAJAVITEL 1 ZAJAVITEL.PK_ZAJAVITEL

## Обмеження:

Найменування	Тип	Стовпець
PK_TRUD_KNIGA	Public	TRUD_KNIGA_ID
FK_TRUD_KNIGA_ZAJAVITEL	Public	ZAJAVITEL_ID

Таблиця 3.10 – Запис в трудовій книжці (додаток В, оператор 1094-1800)

Name	Найменування	Тип	Not Null	Длина
ZAPIS_TRUD_ID	ПК запис трудової	integer	True	
N_PRIEMA	Номер прийому	integer	False	
GOD_PRIEMA	Рік прийому	integer	False	
MES_PRIEMA	Місяць прийому	integer	False	
CISLO_PRIEMA	Число прийому	integer	False	
SVED_PO_STROKE_PRIEM	Відомості по рядку запису прийому	varchar	False	400
OSNOVANIE_STROKI_PRIEM	Основа рядку запису прийому	varchar	False	100
N_UVOLN	Номер звільнення	integer	False	
GOD_UVOLN	Рік звільнення	integer	False	
MES_UVOLN	Місяць звільнення	integer	False	
CISLO_UVOLN	Число звільнення	integer	False	
SVED_PO_STROKE_UVOL	Відомості по рядку запису звільнення	varchar	False	400
OSNOVANIE_STROKI_UVOL	Основа рядку запису звільнення	varchar	False	100
TRUD_KNIGA_ID	ПК трудова книжка	integer	True	
ISP_RASCH	Використовувати в розрахунку	smallint	True	

## Обмеження

Найменування	Тип	Стовпець
PK_ZAPIS_TRUD	Public	ZAPIS_TRUD_ID
FK_ZAPIS_TRUD_TRUD_KNIGA	Public	TRUD_KNIGA_ID

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(TRUD_KNIGA_ID = TRUD_KNIGA_ID)	0..* ZAPIS_TRUD.FK_ZAPIS_TRUD_TRUD_KNIGA 1 TRUD_KNIGA.PK_TRUD_KNIGA
(ZAPIS_TRUD_ID = ZAPIS_TRUD_ID)	0..2 LGOT_STAJ.FK_LGOT_STAJ_ZAPIS_TRUD 1 ZAPIS_TRUD.PK_ZAPIS_TRUD
(ZAPIS_TRUD_ID = ZAPIS_TRUD_ID)	0..1 DLIT_STAJ_RABOT.FK_DLIT_STAJ_RABOT_ZAPIS_TRUD 1 ZAPIS_TRUD.PK_ZAPIS_TRUD

Таблиця 3.11 – Пільговий стаж (додаток В, оператор 1695-1800)

PK	Name	Найменування	Тип	NotNull
True	LGOT_STAJ_ID	ПК пільговий стаж	integer	True
False	NOMER_LGOTNOGO_SPISKA	Номер пільгового списку	integer	True
False	LET	Років пільгового стажу	integer	True
False	MES	Місяців пільгового стажу	integer	True
False	DNEJ	Днів пільгового стажу	integer	True
False	ZAPIS_TRUD_ID	ПК запис трудової	integer	True

## Обмеження:

Найменування	Тип	Стовпець
PK_LGOT_STAJ	Public	LGOT_STAJ_ID
FK_LGOT_STAJ_ZAPIS_TRUD	Public	ZAPIS_TRUD_ID

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(ZAPIS_TRUD_ID = ZAPIS_TRUD_ID)	0..2 LGOT_STAJ.FK_LGOT_STAJ_ZAPIS_TRUD 1 ZAPIS_TRUD.PK_ZAPIS_TRUD

Таблиця 3.12 - Тривалість стажу роботи (додаток В, оператор 1422-1618)

PK	Name	Найменування	Тип	Not Null
True	DLIT_STAJ_RABOT_ID	ПК тривалість стажу роботи	integer	True
False	LET	Років стажу	integer	True
False	MES	Місяців стажу	integer	True
False	DNEJ	Днів стажу	integer	True
PK	Name	Найменування	Тип	Not Null
False	KOR_DAY	Коректувальних днів	integer	False
False	ZAPIS_TRUD_ID	ПК запис трудової	integer	True

## Обмеження:

Найменування	Тип	Стовпець
PK_DLIT_STAJ_RABOT	Public	DLIT_STAJ_RABOT_ID
FK_DLIT_STAJ_RABOT_ZAPIS_TRUD	Public	ZAPIS_TRUD_ID

## Зв'язок з іншими таблицями:

Стовпець	Асоціація
(ZAPIS_TRUD_ID =	0..1 DLIT_STAJ_RABOT.FK_DLIT_STAJ_RABOT_ZAPIS_TRUD
ZAPIS_TRUD_ID)	1 ZAPIS_TRUD.PK_ZAPIS_TRUD

### 3.4 Алгоритм розробки програмного забезпечення

#### 3.4.1 Ескізи екранної форми

У даному розділі представлені ескізи екранної форми програмного забезпечення, що розробляється, в якості доповнення до самого алгоритму програми. Також буде прокоментовано призначення кнопок[11] (додаток В, оператор 122-698).

На рисунку 3.2 представлений загальний зовнішній вигляд програми.

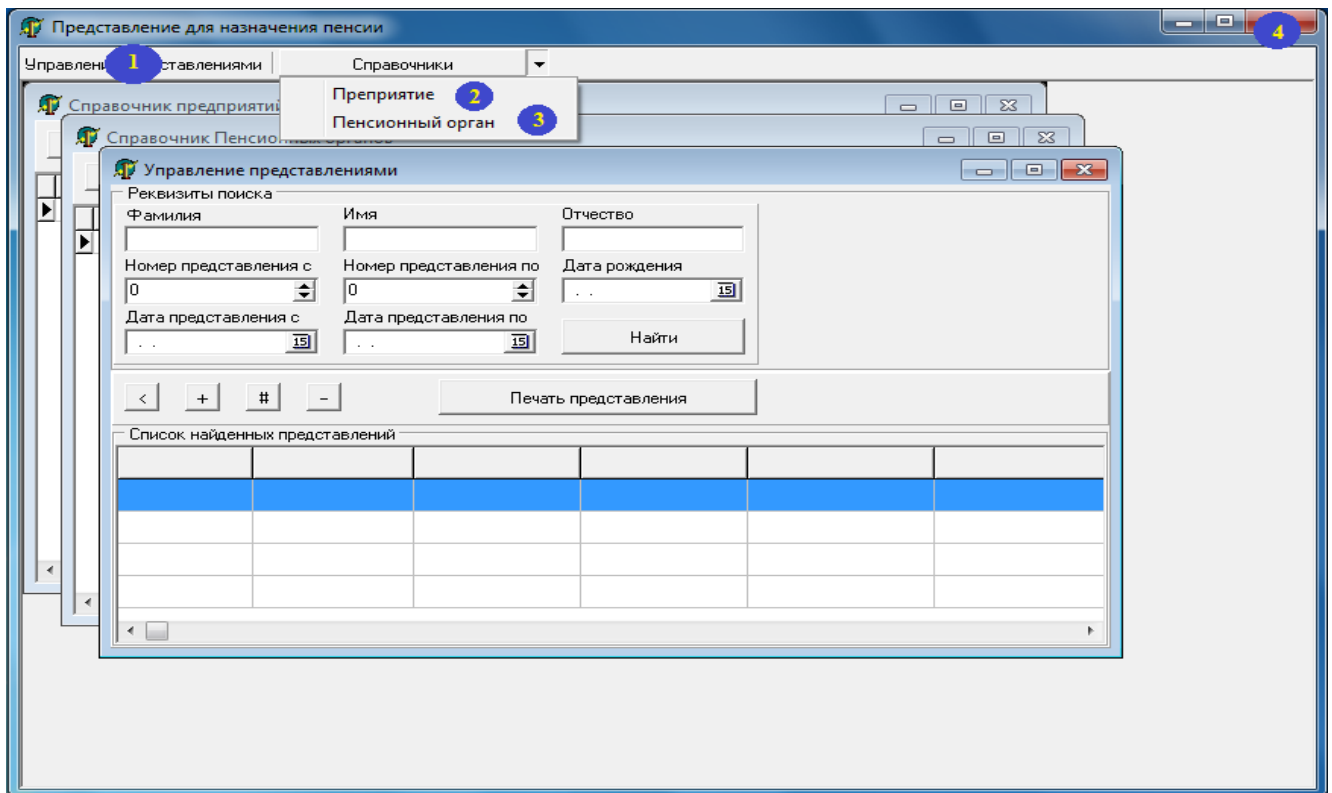


Рисунок 3.2 – Подання для нарахування пенсій

На рисунку 3.3 можна побачити зовнішній вигляд вікна «Довідник підприємств». У цей довідник будуть додаватися всі підприємства, на яких працювали службовці(додаток В, оператор 2010-2100).



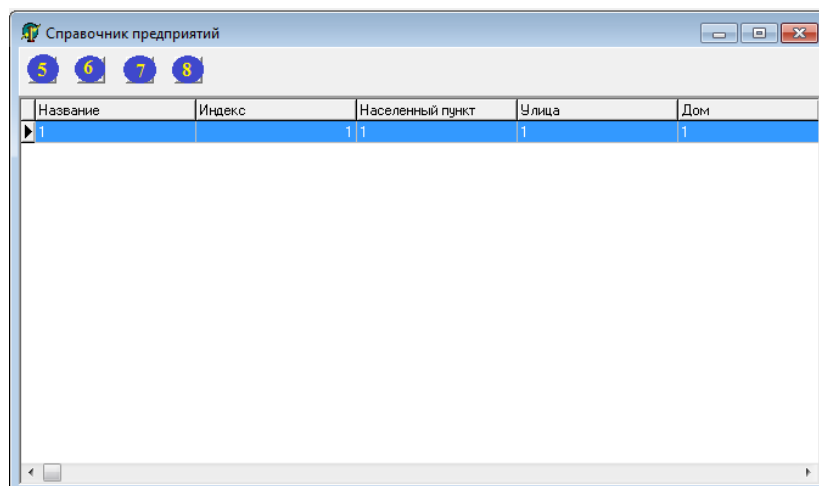


Рисунок 3.3 – Довідник підприємств

На рисунку 3.4 представлена екранна форма Довідника Пенсійних органів (додаток В, оператор 1953-2008).

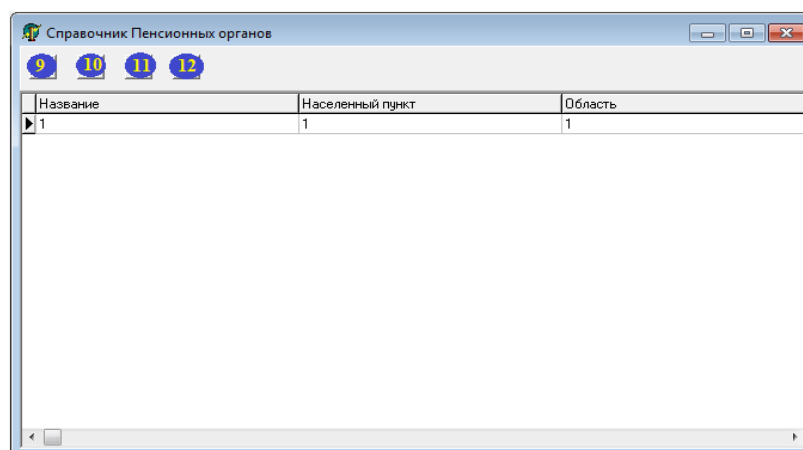


Рисунок 3.4 – Довідник Довідника Пенсійних органів

Рисунок 3.5 містить екранну форму, яка називається Управління поданнями.

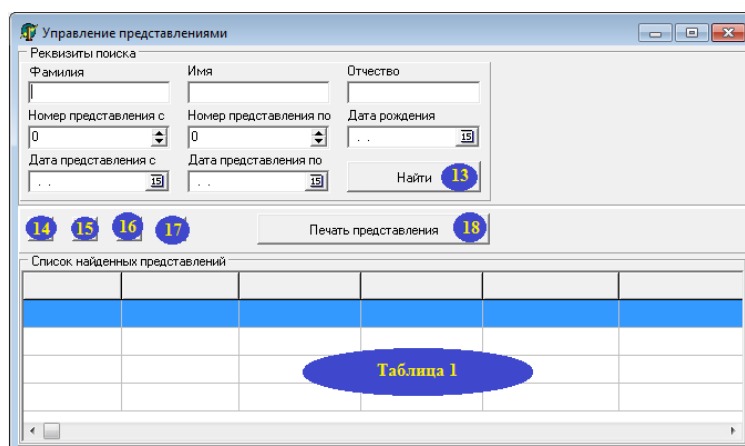


Рисунок 3.5 – Управління поданнями

На рисунку 3.6 представлена екранна форма створення нового подання (додаток В, оператор 122-698).

Рисунок 3.6 – Створення нового подання

Також при нарахуванні пенсії враховуються Непрацевдатні члени сім'ї. Цей момент зазначений на рисунку 3.7 (додаток В, оператор 1892-1951).

Рисунок 3.7 – Непрацевдатні члени сім'ї

На рисунку 3.8 представлено облік трудового стажу за іншими документами заявника (додаток В, оператор 700-1092).

Порядковый номер записи	Документ	Лет стажа по другому док.	Месяцев стажа по другому док.	Дней стажа по другому док.	Корректировочный день

Рисунок 3.8 – Трудовой стаж за іншими документами заявника

Екранна форма стажу роботи за трудовою книжкою представлена на рисунку 3.9 (додаток В, оператор 1094-1889).

Номер приёма	Год приема	Месяц приема	Число приема	Номер увольнения	Год увольнения	Месяц увольнения	Число увольнения
1	3	3	3	2	3	3	3

Длительность стажа  
 Лет: 3, Месяцев: 3, Дней: 0  
 Корректировочные дни: 3

Номер льготы | Лет льготы | Месяцев ль | Дней льготы  
 1 | 2 | 2 | 2  
 2 | 1 | 1 | 1

Сведения по приему  
 Основание приема  
 Сведения по увольнению  
 Основание увольнения

Рисунок 3.9 – Стаж роботи за трудовою книжкою

Далі представлено пояснення до призначення кнопок:

Кнопка 1 – Управління поданнями. При натисненні на цю кнопку з'являється вікно «Управління поданнями».

Кнопка 2 – Підприємство. При натисненні відкривається Довідник підприємств.

Кнопка 3 – Пенсійний орган. При натисненні відкривається Довідник Пенсійних органів.

Кнопка 4 - Вихід.

Кнопка 5 - Згорнути Довідник підприємств.

Кнопка 6 - Додавання нового підприємства.

Кнопка 7 - Зміна інформації по підприємству.

Кнопка 8 - Видалення запису про підприємство.

Кнопка 9-Згорнути Довідник Пенсійних органів.

Кнопка 10 - Додавання нового пенсійного органу.

Кнопка 11 - Зміна інформації про пенсійні органи.

Кнопка 12 - Видалення запису про пенсійні органи.

Кнопка 13 - Знайти подання.

Кнопка 14 - Згорнути вікно «Управління поданнями».

Кнопка 15-Додавання нового подання.

Кнопка 16 - Зміна інформації про подання.

Кнопка 17 - Видалення подання.

Кнопка 18 - Друк подання.

Кнопка 19 - Непрацездатні члени сім'ї.

Кнопка 20 - Стаж роботи за трудовою книжкою.

Кнопка 21 - Стаж за іншими документами.

Кнопка 22 - Ок.

Кнопка 23 - Скасування.

Кнопка 24-Згорнути вікно «Додавання нового непрацездатного члена сім'ї».

Кнопка 25 - Додавання нового непрацездатного члена сім'ї.

Кнопка 26 - Зміна інформації про непрацездатним членам сім'ї.

Кнопка 27 - Видалення непрацездатного члена сім'ї.

Кнопка 28 - Згорнути вікно «Трудовий стаж за іншими документами заявника».

Кнопка 29 - Додати новий документ заявника.

Кнопка 30 - Змінити інформацію про документ.

Кнопка 31 - Видалити інформацію про документ заявника.

Кнопка 32 – Згорнути вікно «Стаж роботи за трудовою книжкою».

Кнопка 33 – Додати запис до трудової книжки.

Кнопка 34 – Змінити запис у трудовій книжці.

Кнопка 35 –Видалення запису по трудовій книжці.

Кнопка 36 - Розрахунок стажу роботи.

### **3.4.2 Алгоритм роботи програми**

На рисунку 3.10 представлено схему алгоритму роботи програмного забезпечення.

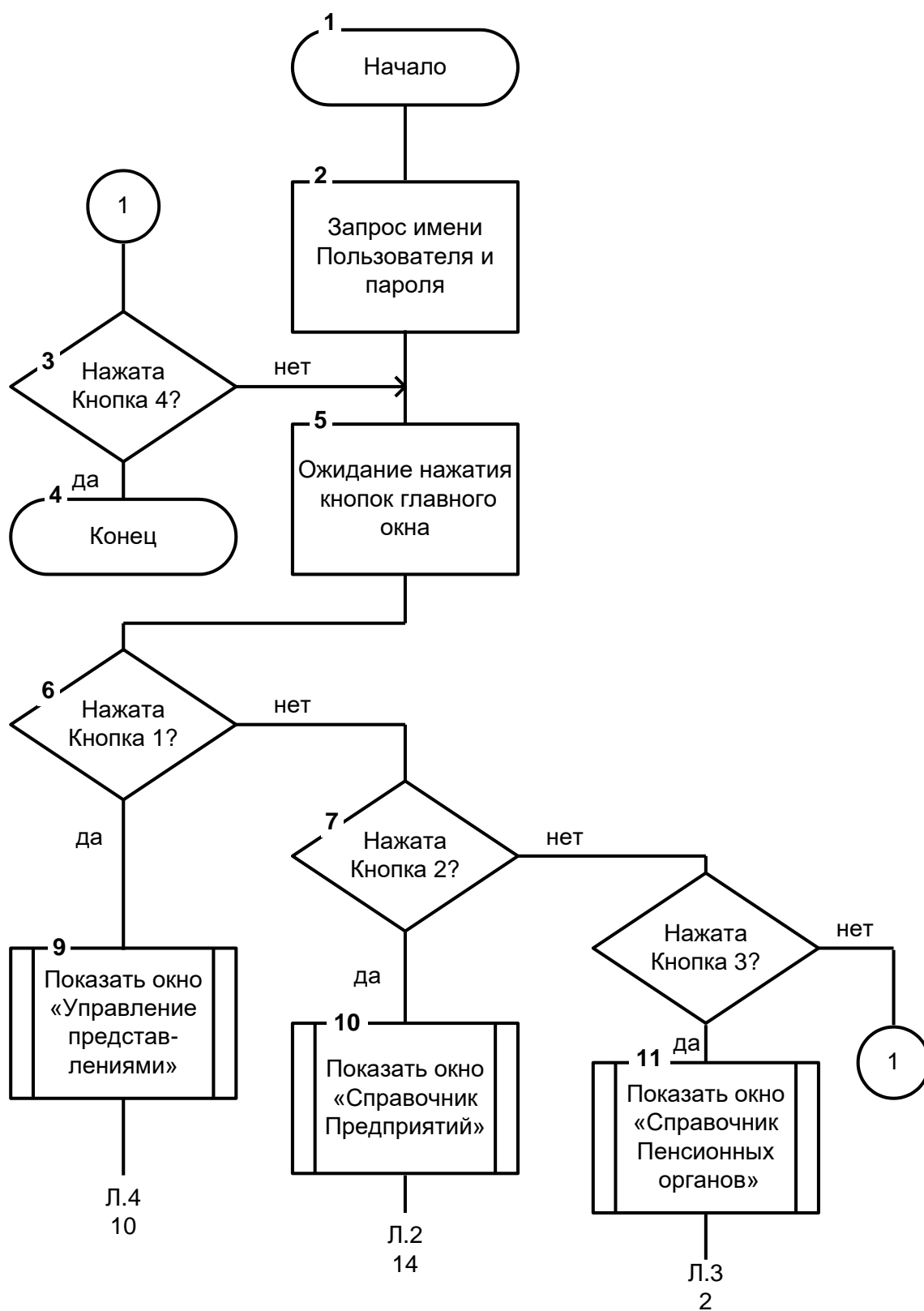


Рисунок 3.10 – Схема алгоритму. Аркуш 1

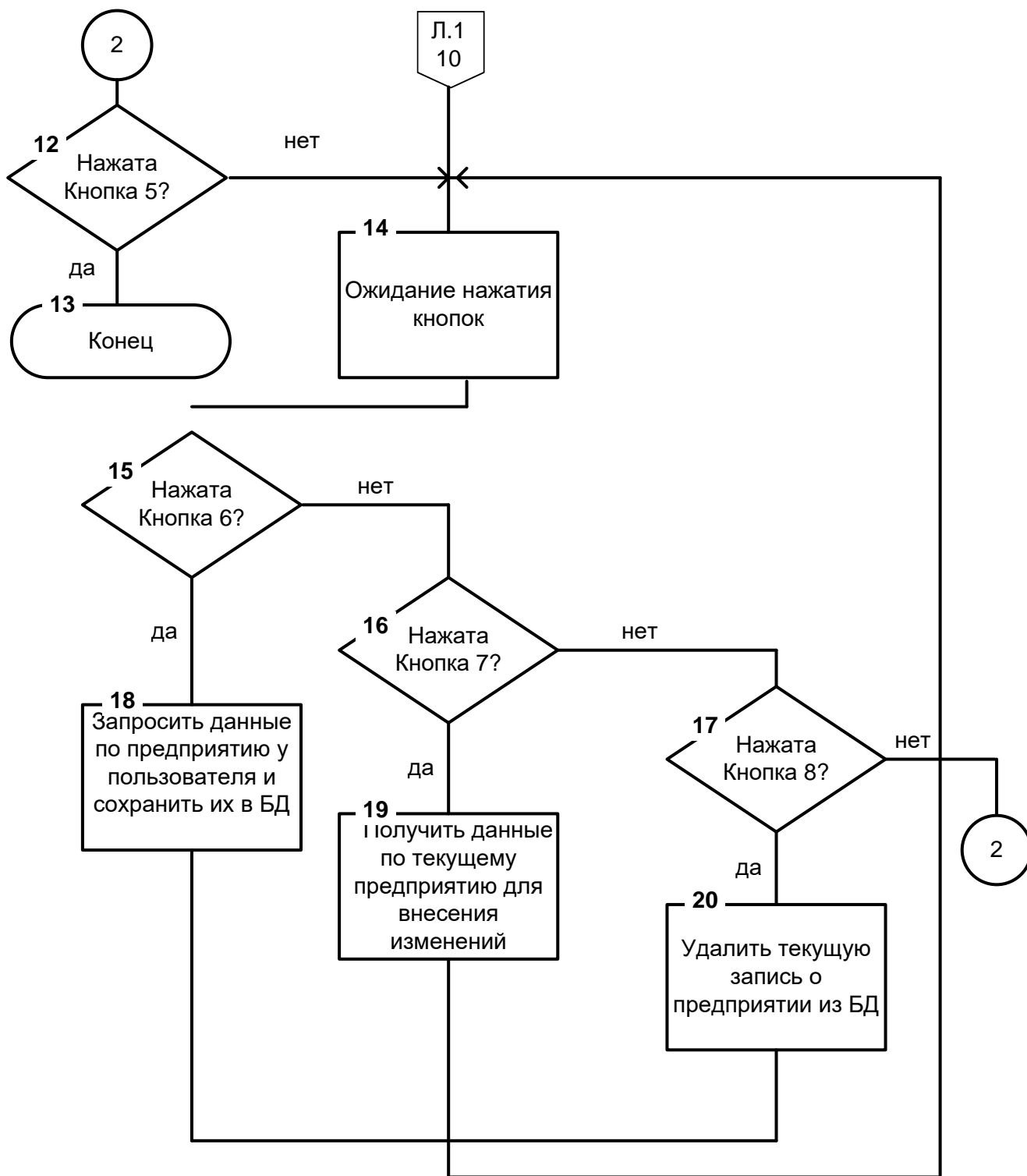


Рисунок 3.10. Аркуш 2

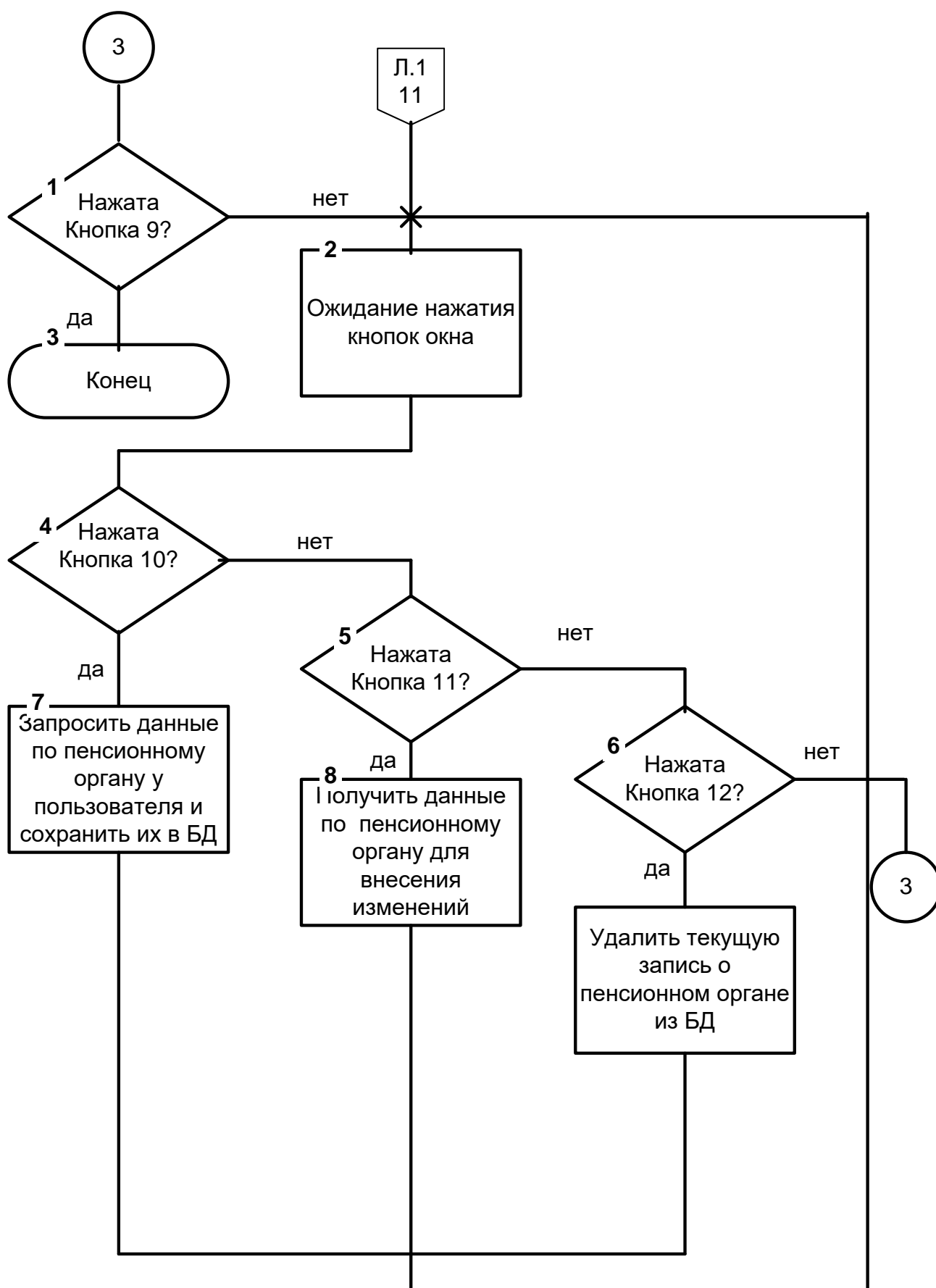


Рисунок 3.10. Аркуш 3



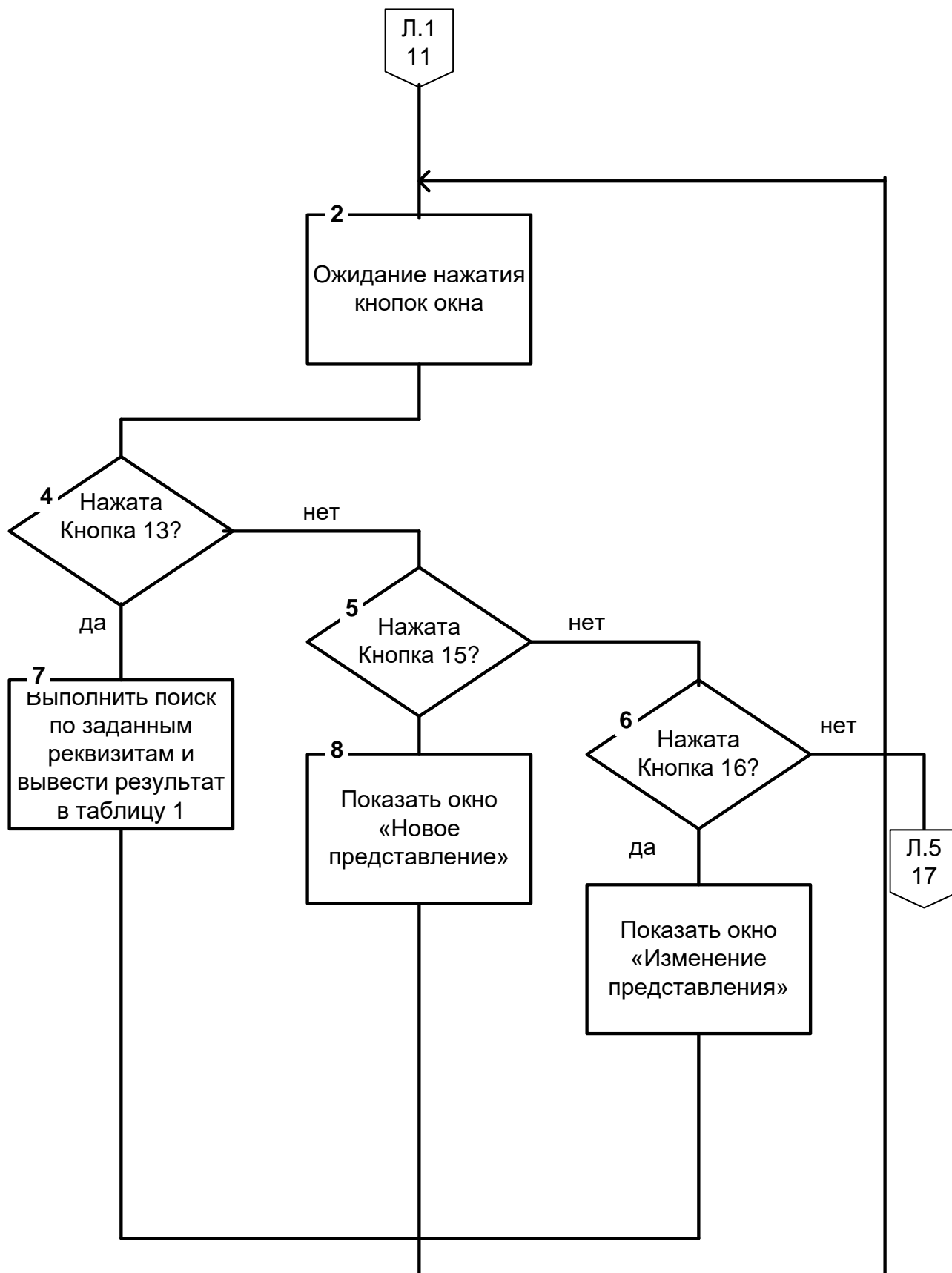


Рисунок 3.10. Аркуш 4

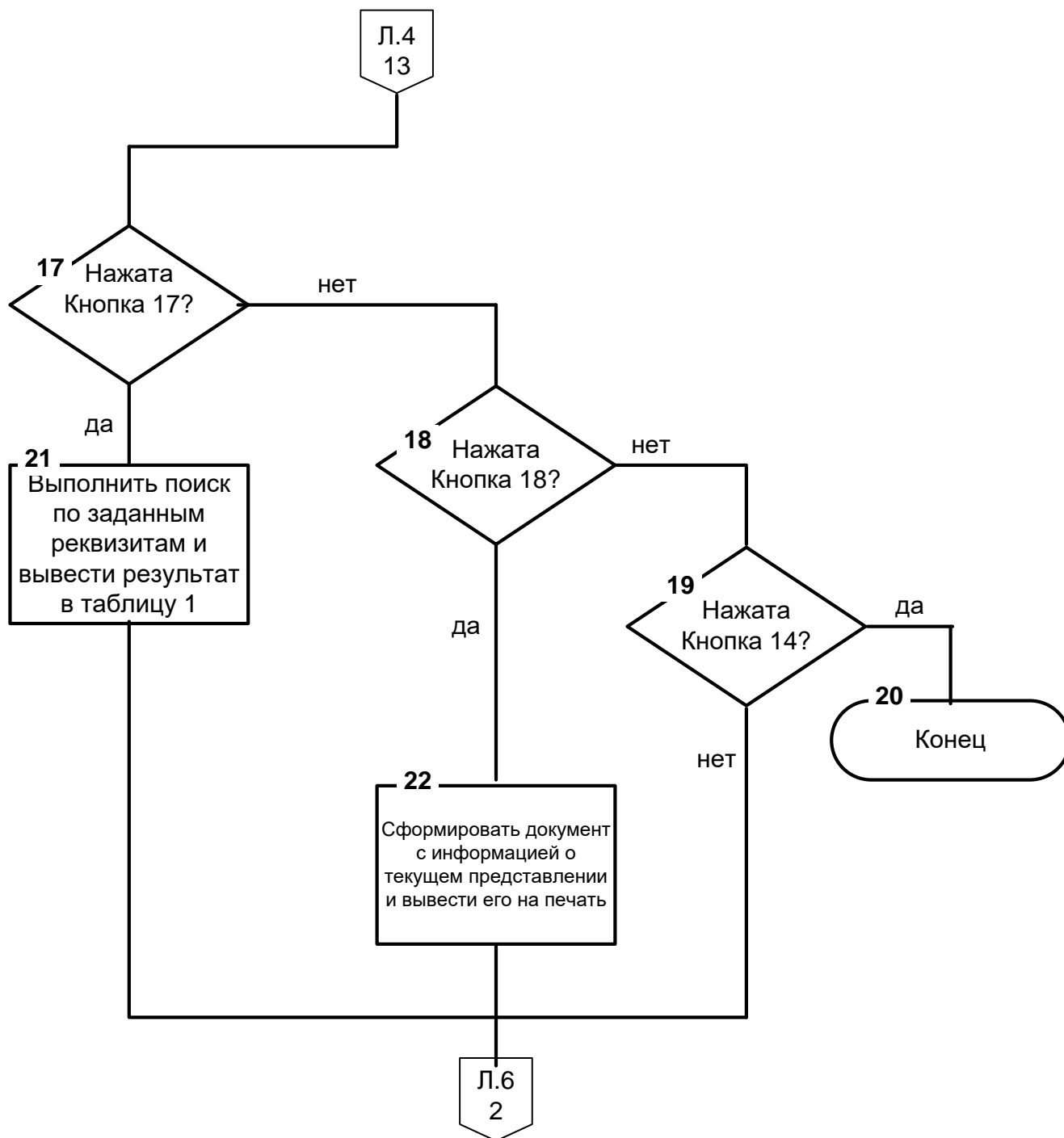


Рисунок 3.10. Аркуш 5

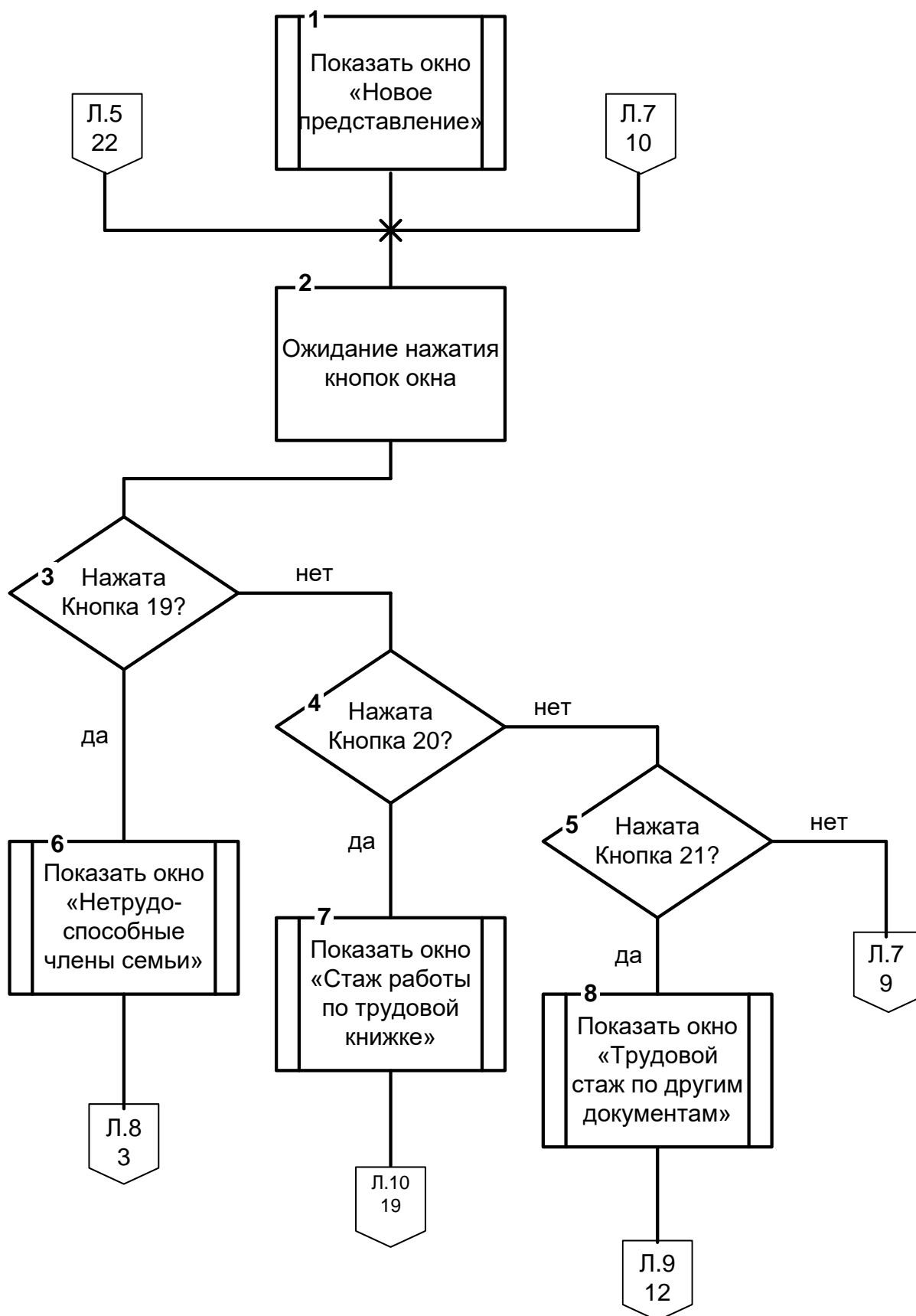


Рисунок 3.10. Аркуш 6

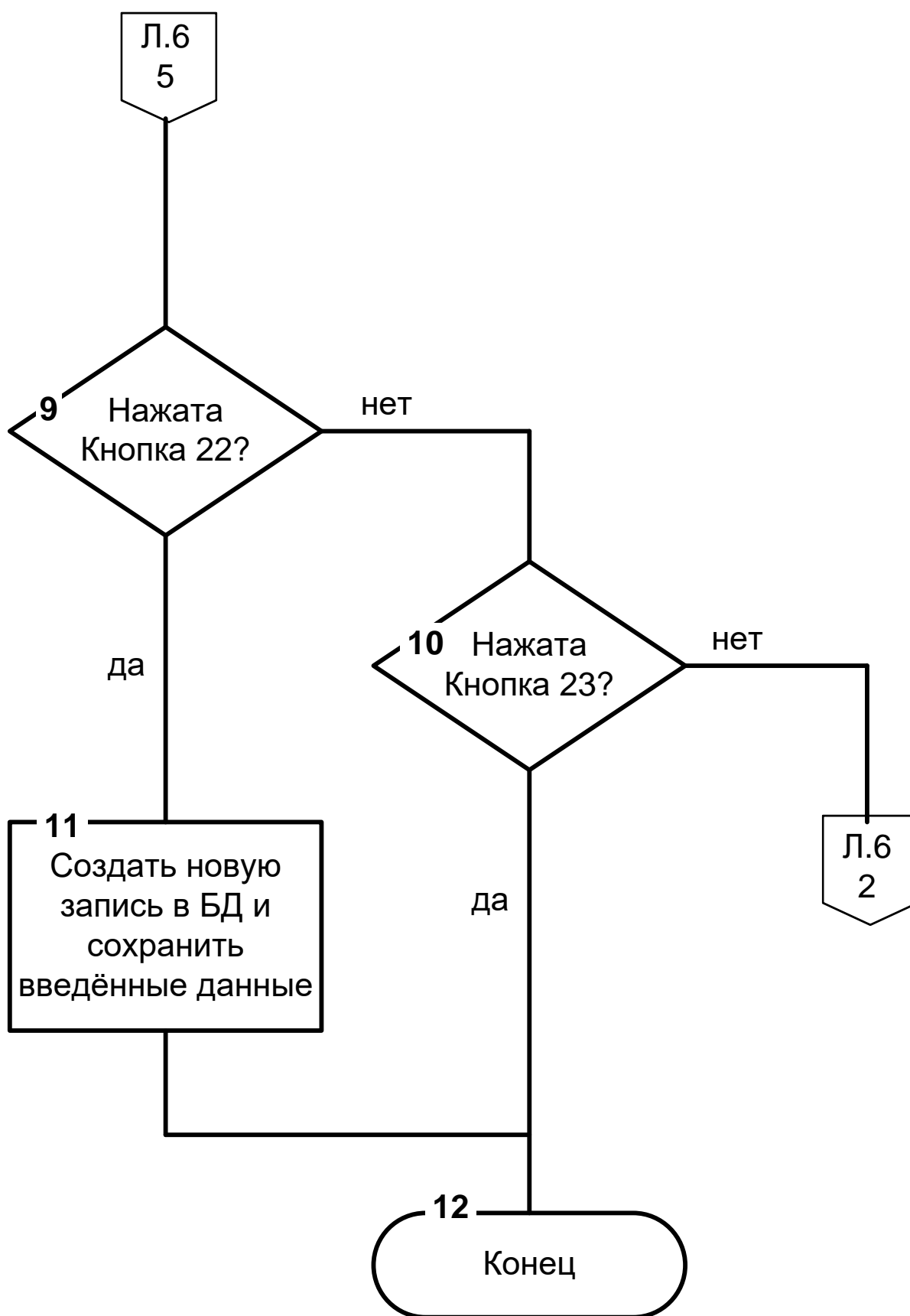


Рисунок 3.10. Аркуш 7

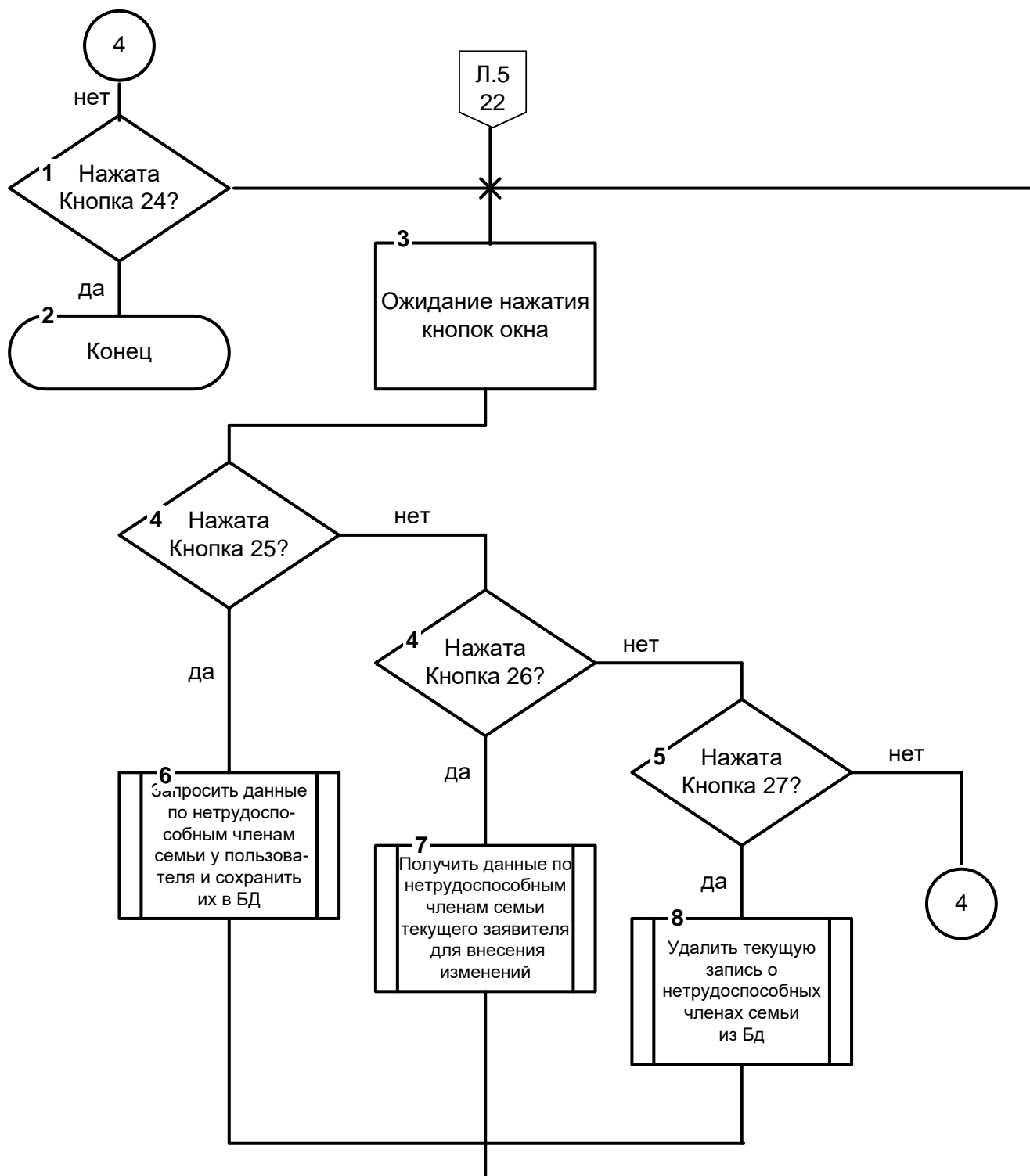


Рисунок 3.10. Аркуш 8

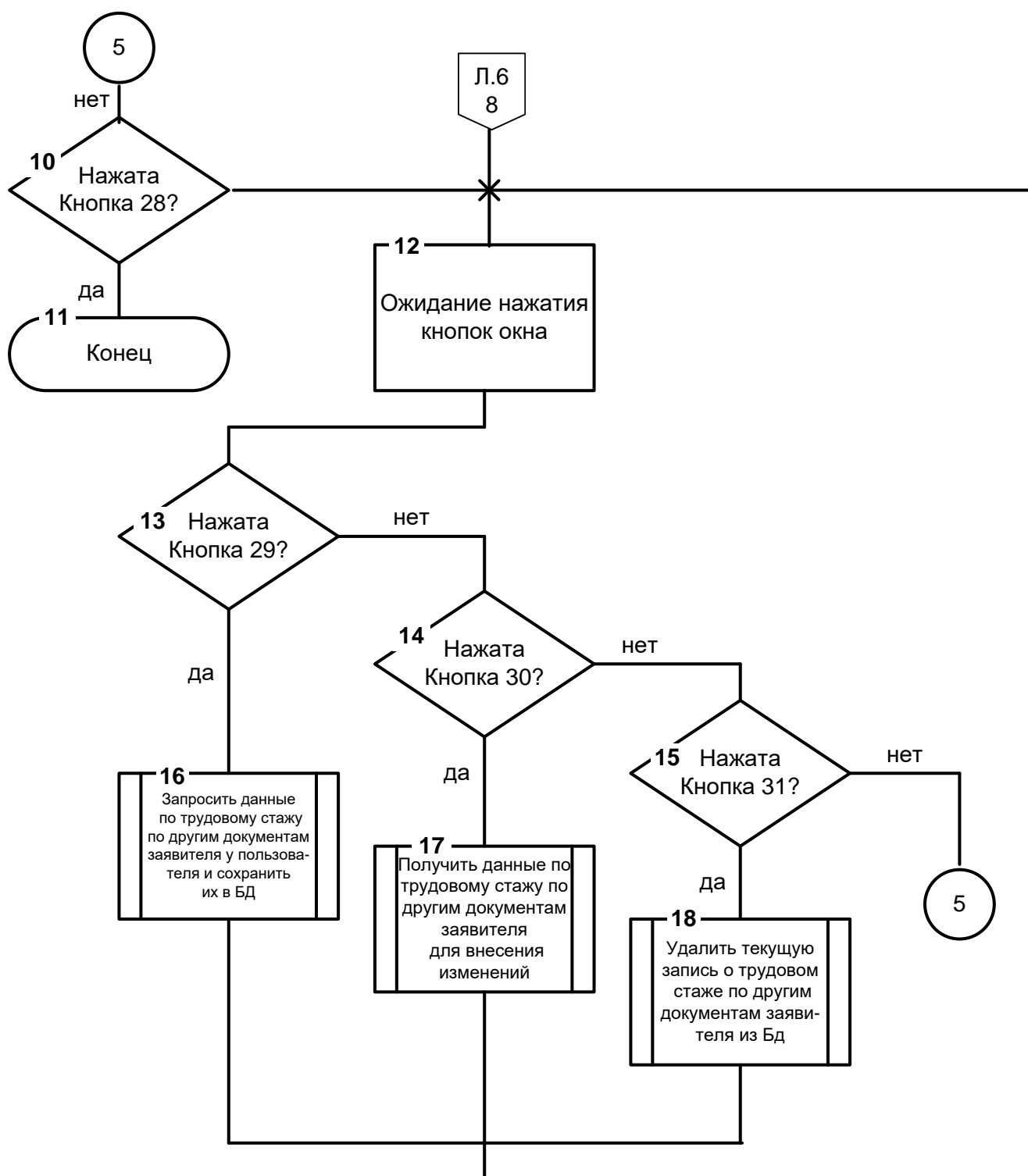


Рисунок 3.10. Аркуш 9

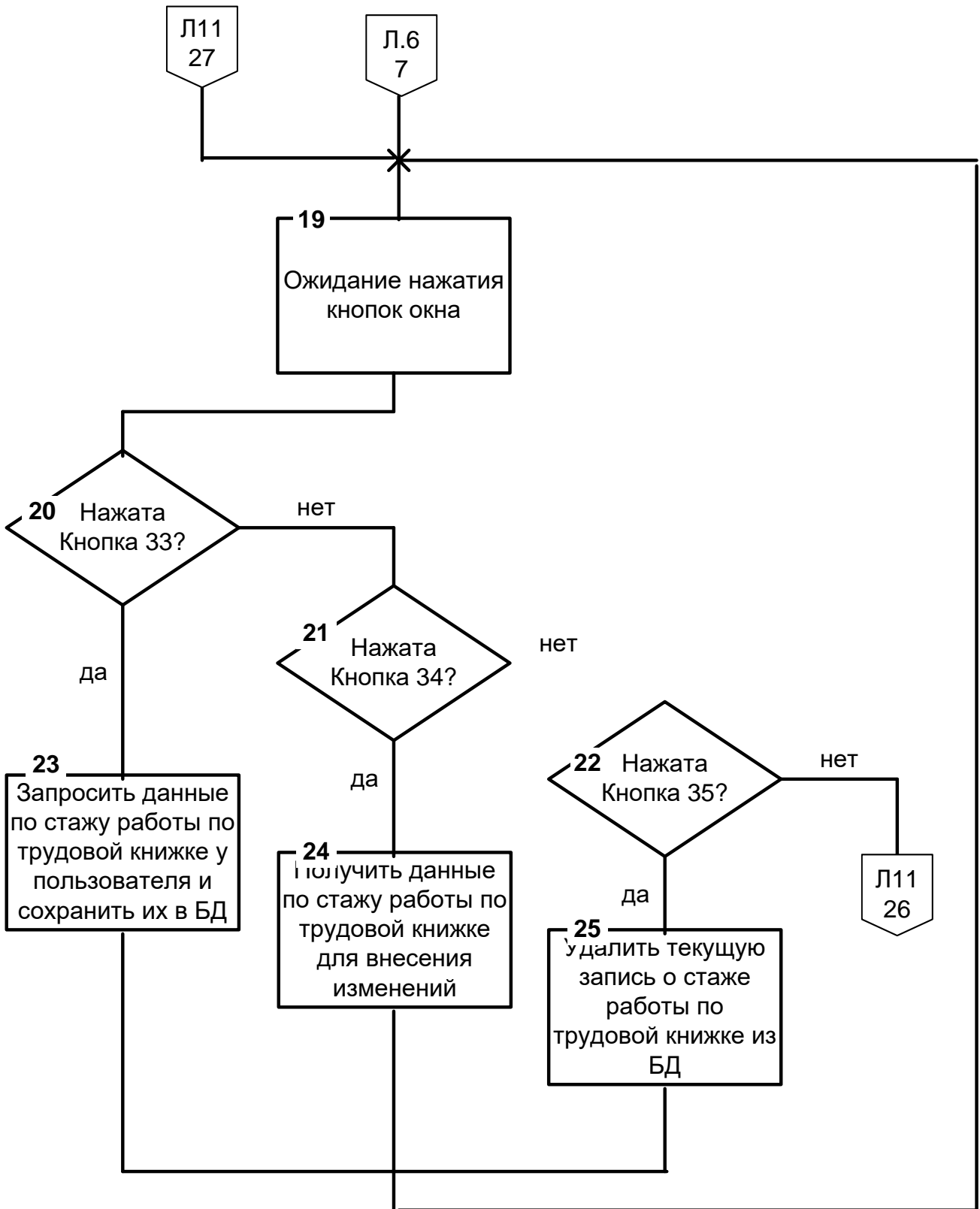


Рисунок 3.10. Аркуш 10

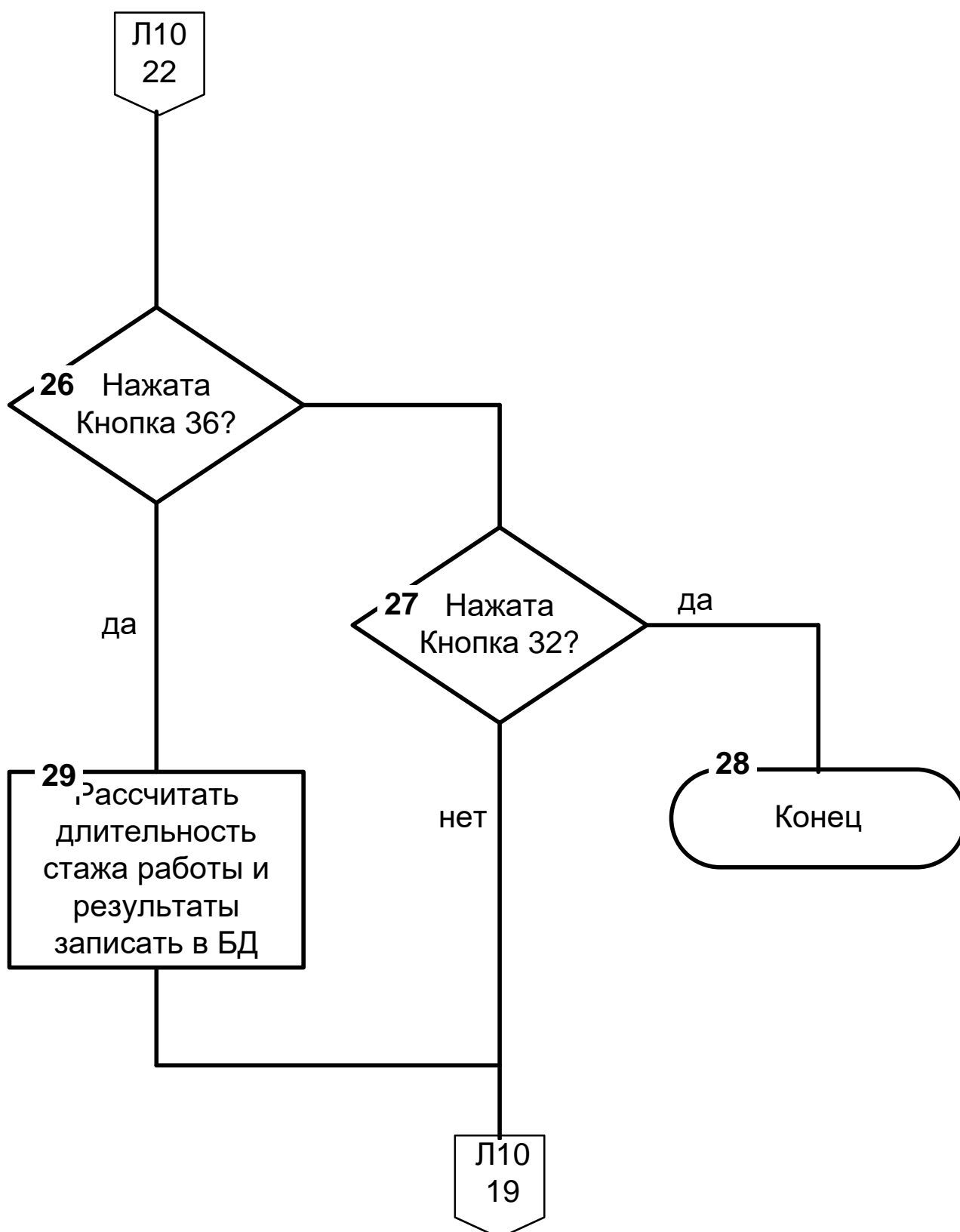


Рисунок 3.10. Аркуш 11



### 3.4.3 Структура програми

Програма має модульну структуру. Кожен модуль являє собою діалогове вікно зі своїм набором функцій. Взаємозв'язки між діалоговими вікнами програми показані на рисунку 3.11.

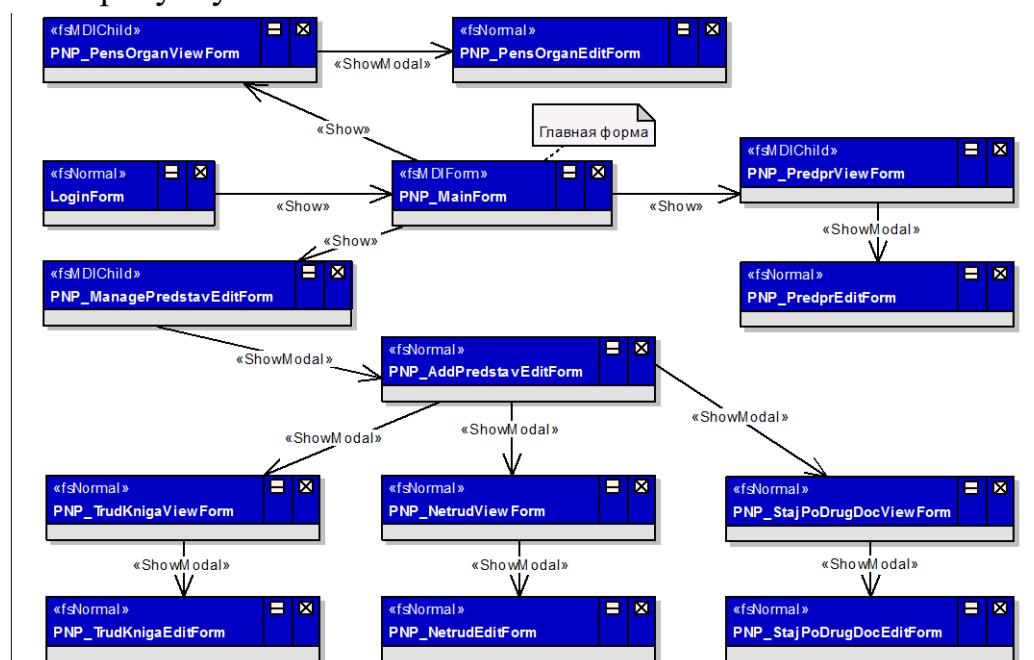


Рисунок 3.11 - Взаємозв'язки між діалоговими вікнами програми

В таблиці 3.13 описані форми програми.

Таблиця 3.13 - Форми програми

Імена форм	Тип форм	Заголовки форм
LoginForm	«fsNormal»	Авторизація
PNP_MainForm	«fsMDIForm»	Подання для нарахування пенсій
PNP_AddPredstavEditForm	«fsNormal»	Додавання/Зміна подання
PNP_ManagePredstavEditForm	«fsMDIChild»	Управління поданнями
PNP_NetrudEditForm	«fsNormal»	Додавання/Зміна інформації щодо непрацездатного члена сім'ї
PNP_NetrudViewForm	«fsNormal»	Непрацездатні члени сім'ї
PNP_PensOrganEditForm	«fsNormal»	Додавання/Зміна інформації про пенсійний орган
PNP_PensOrganViewForm	«fsMDIChild»	Довідник Пенсійних органів
PNP_PredprEditForm	«fsNormal»	Додавання/Зміна інформації про підприємство
PNP_PredprViewForm	«fsMDIChild»	Довідник підприємств
PNP_StajPoDrugDocEditForm	«fsNormal»	Додавання/Зміна запису за іншими документами
PNP_StajPoDrugDocViewForm	«fsNormal»	Трудовий стаж за іншими документами заявника
PNP_TrudKnigaEditForm	«fsNormal»	Додавання/Зміна запису трудової книжки
PNP_TrudKnigaViewForm	«fsNormal»	Стаж роботи за трудовою книжкою

Модулі PNP\_PensOrganEditForm, PNP\_PredprEditForm, PNP\_AddPredstavEditForm, PNP\_NetrudEditForm, PNP\_StajPoDrugDocEditForm, PNP\_TrudKnigaEditForm є нащадками базового класу TPNP\_EditFormBase. Дані модулі, призначені для додавання або редагування записів таблиць бази даних.

Модулі PNP\_PensOrganViewForm, PNP\_PredprViewForm, PNP\_StajPoDrugDocViewForm, PNP\_NetrudViewForm, PNP\_TrudKnigaViewForm, PNP\_ManagePredstavEditForm (рисунок 3.11) для відображення значень полів таблиць бази даних і виклику відповідних діалогових вікон для додавання або редагування записів таблиць бази даних [11].

На рисунку 3.11 зображена форма «LoginForm»: при запуску програми PNP виконати читання логіна із ini - файлу, відобразити форму «LoginForm» із заголовком «Авторизація». При натисканні «ОК» виконати підключення до сервера БД, записати логін в ini - файл, відобразити форму PNP\_MAIN\_FORM» із заголовком «Подання для нарахування пенсій».

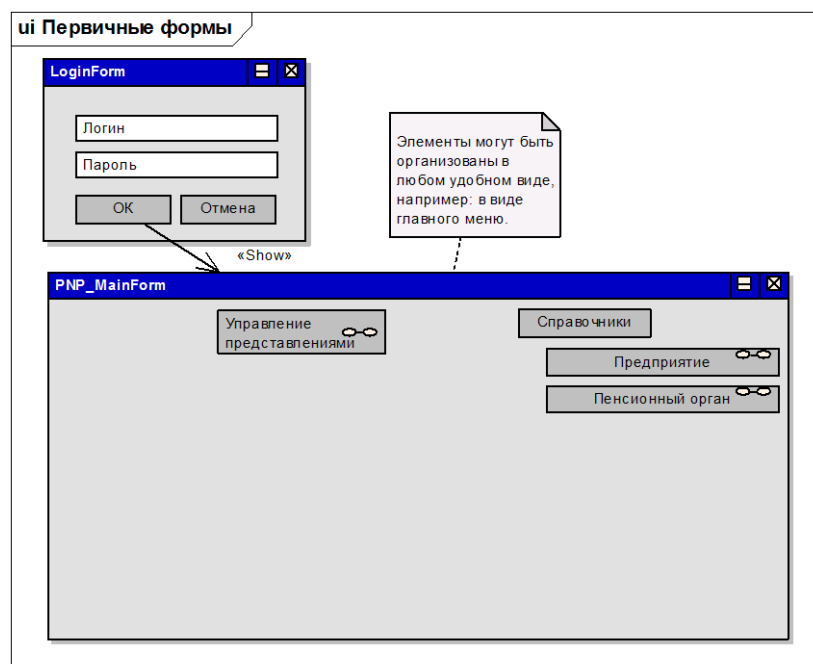


Рисунок 3.11 - Форма «PNP\_PredprViewForm»

На рисунку 3.12 зображена форма «PNP\_PredprViewForm»:

- В таблиці відображається результат виконання запроса SQL1.
- По кнопці «+» відображається форма «PNP\_PredprEditForm» із

заголовком «Додавання нового підприємства», якщо результат форми «ОК» - виконується додавання нового підприємства.

- По кнопці «#» відображається форма «PNP\_PredprEditForm» із заголовком «Зміна інформації по підприємству» з інформацією по поточному запису, якщо результат форми «ОК» - виконується Зміна інформації по підприємству [13].

- По кнопці «-» відображається діалогове вікно «Підтвердження», якщо результат «ОК» - виконати Видалення інформації в довіднику підприємств (додаток В, оператор 1-120).

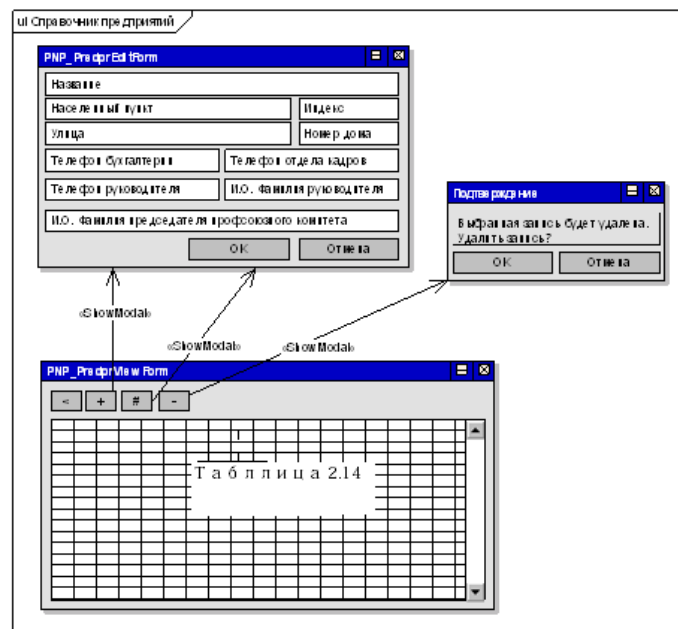


Рисунок 3.12 - Форма «PNP\_PredprViewForm» (додаток В, оператор 2010-2100)

При натисненні кнопки «<» необхідно закрити форму. При закритті форми управління необхідно повернути формі, що викликала.

На рисунку 3.13 зображена форма «PNP\_PensOrganEditForm».

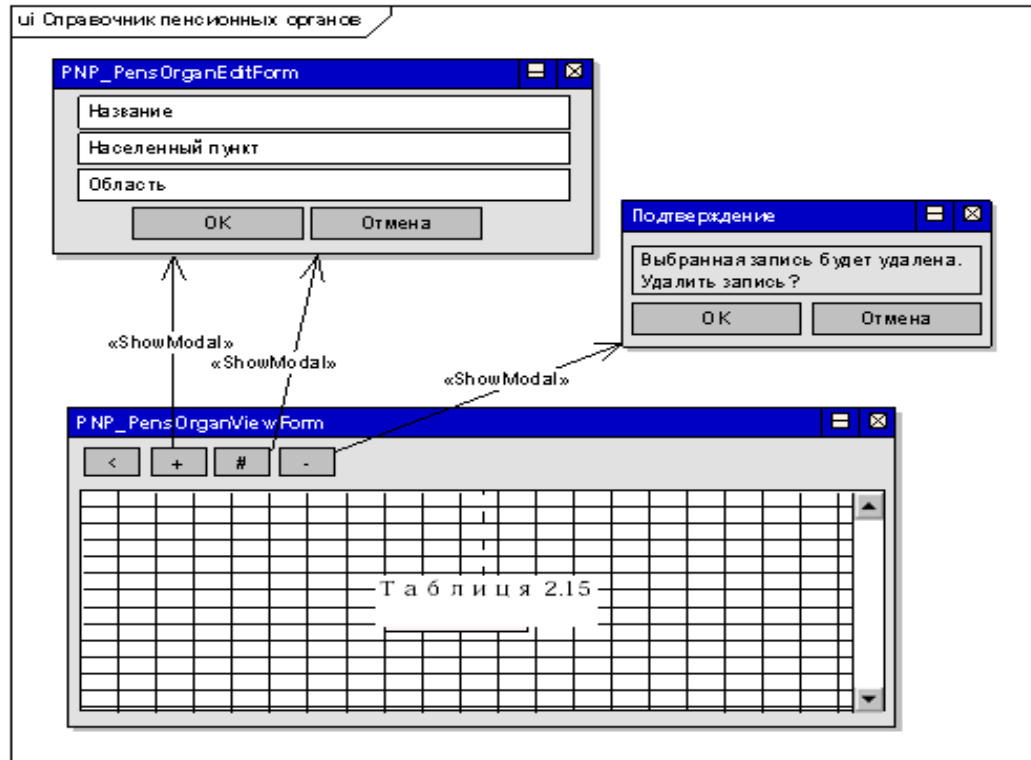


Рисунок 3.13 - Форма «PNP\_PensOrganEditForm»

Форма «PNP\_PensOrganEditForm» (додаток В, оператор 1953-2008):

- В таблиці відображається результат виконання запису SQL2.
- По кнопці «+» відображається форма «PNP\_PensOrganEditForm» із заголовком «Додавання пенсійного органу». Якщо результат форми «ОК» – виконати додавання пенсійного органу [14].
- По кнопці «#» відображається форма «PNP\_PensOrganEditForm» із заголовком «Зміна інформації щодо пенсійного органу» з інформацією щодо поточного запису в відповідній таблиці, якщо результат форми «ОК» - виконати зміну інформації щодо пенсійного органу. Реквізити пенсійного органу з форми записати в таблицю 3.1.
- По кнопці «-» відображається діалогове вікно "Підтвердження", якщо результат «ОК» виконати видалення пенсійного органу поточного запису в таблиці 3.1.
- По кнопці «<» необхідно закрити форму. Після закриття форми управління необхідно повернути формі, що викликала.

На рисунку 3.14 зображена форма «PNP\_MnagePredstavEditForm».

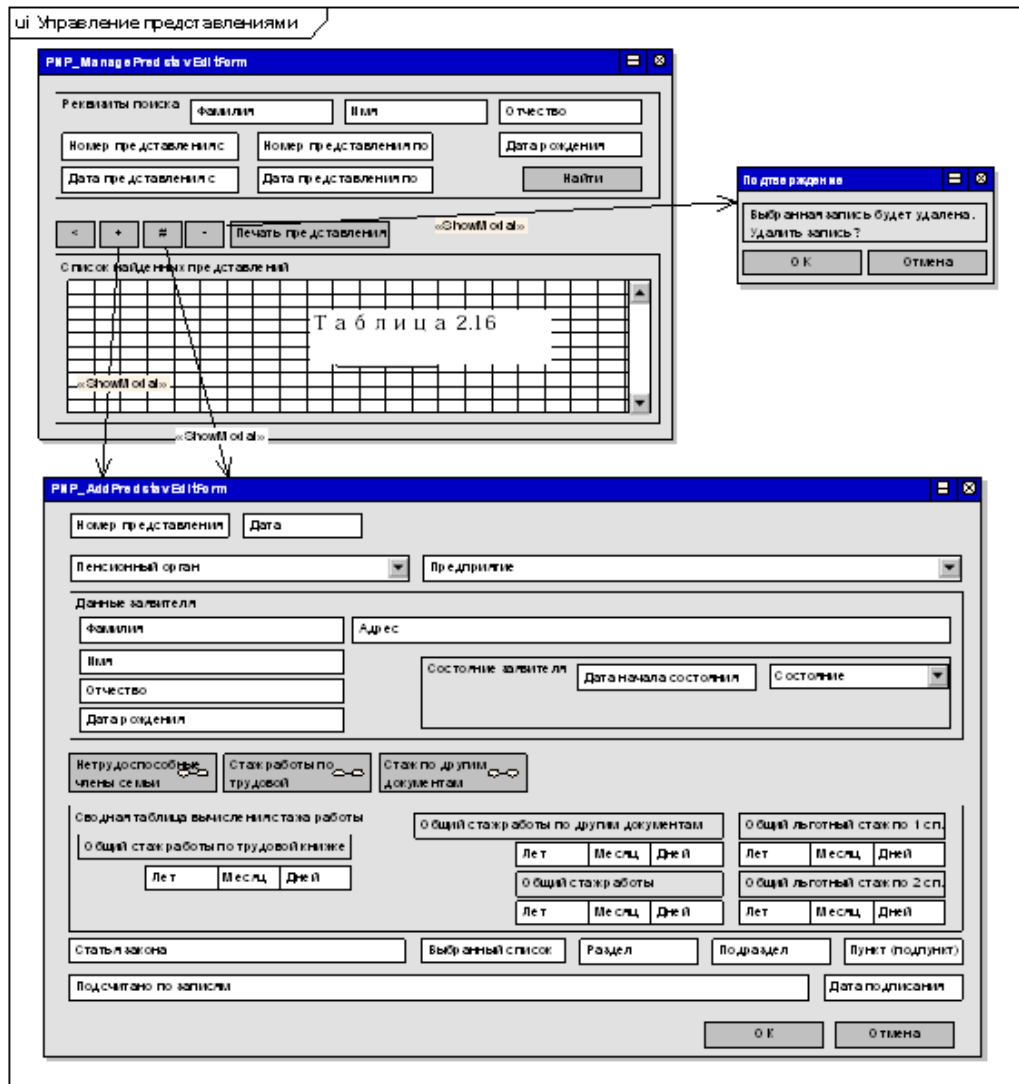


Рисунок 3.14 - Форма «PNP\_MnagePredstavEditForm»

Форма «PNP\_MnagePredstavEditForm» (додаток В, оператор 122-698):

- По кнопці «Знайти» виконується пошук подання: виконати SQL запит:  
 select p.nomer\_predst, p.date\_predst, z.familija, z.imja, z.otchestvo, z.data\_rojd  
 from zajavitel z join predstav p on z.zajavitel\_id = p.zajavitel\_id  
 where (p.nomer\_predst between: id1 and: id2) and (p.date\_predst between: id3 and:  
 id4) and (z.familija =: id5) and (z.imja =: id6) and (z.otchestvo =: id7 ) and (z.data\_rojd =:  
 id8)

Параметру запиту id присвоїти значення відповідних полів з форми.

Якщо який-небудь з параметрів id запиту відсутній, тоді його в запит не

включати. Результат виконання відобразити в таблиці 2.16. Параметри описані в таблиці 3.14 .

Таблиця 3.14 - Параметри

Параметр	Присвоюється значення з поля форми
id1	Номер подання з
id2	Номер подання по Якщо відсутній прирівняти до id1
id3	Дата подання з
id4	Дата подання по Якщо відсутній прирівняти до id3
id5	Прізвище
id6	Ім'я
id7	Побатькові
id8	Дата народження

- По кнопці «+» виконується створення подання:

Визначити номер останнього подання: виконати SQL запит:  
selectmax (Predst) asmax\_nFromPREDSTAV

Потім присвоїти змінній MAX значення результату SQL запиту (max\_n).

Створити запис у таблиці PREDSTAV.

Записати в PREDSTAV.NOMER\_PREDST значення MAX 1.

Записати в PREDSTAV.DATE\_PREDST поточну дату у форматі dd.mm.yyyu.

Після виконання відобразити форму «PNP\_AddPredstavEditForm» із заголовком «Нове подання», якщо результат форми «ОК» виконати збереження створеного подання:

Створити запис в таблиці 3.3, реквізити заявника записати з форми до таблиці 3.3.

Реквізити подання записати з форми в таблицю 3.7.

- По кнопці «#» відображається форма «PNP\_AddPredstavEditForm» із заголовком «Зміна подання» з інформацією за поточним записом в таблиці 2.16, якщо результат

форми «ОК» - виконати збереження зміненого подання:

1) Реквізити заявника записати з форми в таблицю 3.3.

2) Реквізити подання записати з форми в таблицю 3.7.

- По кнопці «-» відображається діалогове вікно «Підтвердження», якщо результат «ОК» виконати видалення подання:

1) Знайти в таблиці 3.3 запис з ZAJAVITEL\_ID рівний PREDSTAV.ZAJAVITEL\_ID поточного подання.

2) Видалити знайдену запис в таблиці 3.3.

- По кнопці «Непрацевдатні члени сім'ї» відображається форма «PNP\_NetrudViewForm» із заголовком «Непрацевдатні члени сім'ї».

- По кнопці «Стаж роботи за трудовою» відображається форма «PNP\_TrudKnigaViewForm» із заголовком «Стаж роботи за трудовою книжкою».

- По кнопці «Стаж за іншими документами» відображається форма «PNP\_StajPoDrugDocViewForm» із заголовком «Стаж роботи за іншими документами».

- По кнопці «Друк подання» виконується формування документа «Подання для нарахування пенсій». Документ має форму, представлену в додатках А, та формується за заданим поданням, згідно наступного алгоритму:

- Заповнити поля документа:

<P4> – PREDSTAV.DATE\_PREDST;

<P5> – PREDSTAV.NOMER\_PREDST;

<P42> – PREDSTAV.LGOT\_STATJA\_ZAKONA;

<P43> – PREDSTAV.LGOT\_NOMER\_SPISKA;

<P44> – PREDSTAV.LGOT\_RAZDEL\_SPISKA;

<P45> – PREDSTAV.LGOT\_PODRAZDEL\_SPISKA;

<P46> – PREDSTAV.LGOT\_PUNKT\_STAJA;

<P48> – PREDSTAV.LGOT\_PODS\_PO\_ZAPIS;

<P51> – PREDSTAV..

- В таблиці 3.1 знайти запис з PENS\_ORGAN\_ID, що дорівнює PREDSTAV.PENS\_ORGAN\_ID поточного подання. Заповнити поля документу:

- <P1> – PENS\_ORGAN.NAZVANIE;
- <P2> – PENS\_ORGAN.NAS\_PUNKT;
- <P3> – PENS\_ORGAN.OBLAST.

В таблиці 3.3 знайти запис з ZAJAVITEL\_ID, що дорівнює PREDSTAV.ZAJAVITEL\_ID поточного подання. Заповнити поля документу:

- <P14> – ZAJAVITEL.FAMILIJA;
- <P15> – ZAJAVITEL.IMJA;
- <P16> – ZAJAVITEL.OTCHESTVO;
- <P17> – ZAJAVITEL.DATA\_ROJD;
- <P18> – ZAJAVITEL.ADRES.

Якщо ZAJAVITEL.SOST\_ZAJAVITEL\_ID = 1 тоді

- <P53> – ZAJAVITEL.DATE\_SOSTOYANIE

Якщо ZAJAVITEL.SOST\_ZAJAVITEL\_ID = 2 тоді

- <P53a> – ZAJAVITEL.DATE\_SOSTOYANIE

Якщо ZAJAVITEL.SOST\_ZAJAVITEL\_ID = 3 тоді

- <P53б> – ZAJAVITEL.DATE\_SOSTOYANIE

В таблиці 3.2 знайти запис з \_ID, що дорівнює ZAJAVITEL.\_ID поточного заявника. Заповнити поля документу:

- <P6> – PREDPR.NAZVANIE;
- <P7> – PREDPR.INDEKS;
- <P8> – PREDPR.NAS\_PUNKT;
- <P9> – PREDPR.ULICA;
- <P10> – PREDPR.DOM;
- <P11> – PREDPR.TEL\_RUK;
- <P12> – PREDPR.TEL\_BUH;
- <P13> – PREDPR.TEL\_O\_K;
- <P49> – PREDPR.RUKOVODITEL;
- <P50> – PREDPR.PROFSOUZ.



В таблиці 3.6 знайти запис з ZAJAVITEL\_ID, що дорівнює ZAJAVITEL.ZAJAVITEL\_ID поточного заявника. Заповнити поля документа:

<P19> – Порядковий номер рядку в таблиці документа (присвоює система);

<P20> – NETRUD.F\_I\_O;

<P21> – NETRUD.RODSTVEN\_SVAZ;

<P22> – NETRUD.DATA\_ROJD.

В таблиці 3.9 знайти запис з ZAJAVITEL\_ID, що дорівнює ZAJAVITEL.ZAJAVITEL\_ID поточного заявника. Заповнити поля документа:

<P23> – TRUD\_KNIGA.DATA\_ZAPOLN.

В таблиці 3.10 знайти запис з , що дорівнює .поточного рядку трудової книги, відсортувати записи по .та ZAPIS\_TRUD.N\_UVOLN. Заповнити поля документа:

<P24> – ZAPIS\_TRUD.N\_PRIEMA;

<P24a> – ZAPIS\_TRUD.N\_UVOLN;

<P25> – ZAPIS\_TRUD.GOD\_PRIEMA;

<P35> – ZAPIS\_TRUD.GOD\_UVOLN;

<P26> – ZAPIS\_TRUD.MES\_PRIEMA;

<P36> – ZAPIS\_TRUD.MES\_UVOLN;

<P27> – ZAPIS\_TRUD.CISLO\_PRIEMA;

<P37> – ZAPIS\_TRUD.CISLO\_UVOLN;

<P28> – ZAPIS\_TRUD.SVED\_PO\_STROKE\_PRIEM;

<P28a> – ZAPIS\_TRUD.SVED\_PO\_STROKE\_UVOL;

<P29> – ZAPIS\_TRUD.OSNOVANIE\_STROKI\_PRIEM;

<P29a> – ZAPIS\_TRUD.OSNOVANIE\_STROKI\_UVOL.

В таблиці 3.5 знайти запис з , що дорівнює ZAJAVITEL.ZAJAVITEL\_ID поточного заявника. Заповнити поля документа:

<P246> – OTHER\_DOC.N\_DOC;

<P38> – OTHER\_DOC.DOCUMENT.

По кнопці «<>» закрити форму.

Під час закриття форми управління повернути формі, що викликала.

Форма «PNP\_NetrudViewForm» (рис.3.15): (додаток В, оператор 1892-1951).

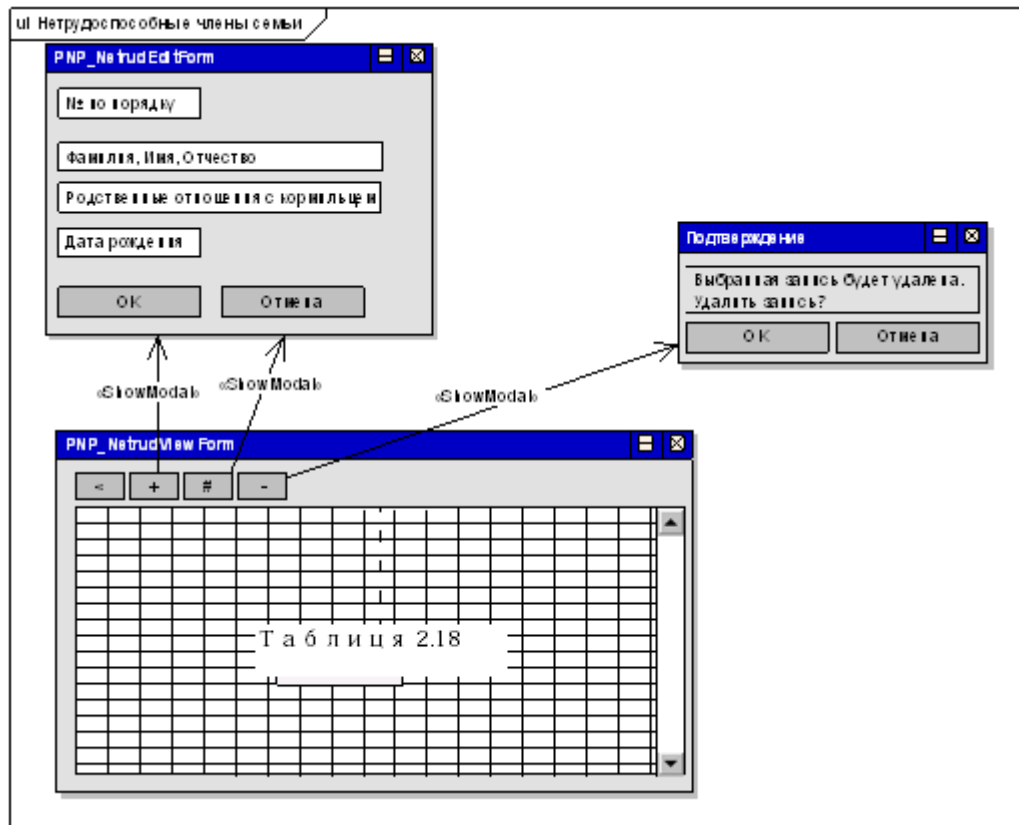


Рисунок 3.15 - Форма «PNP\_NetrudViewForm»

В таблиці 2.18 (рис.3.15) відображається результат виконання запиту SQL3.

По кнопці «+» відображається форма «PNP\_NetrudEditForm» із заголовком «Новий непрацездатний член сім'ї заявника», якщо результат форми «ОК» - виконати створення нового непрацездатного члена сім'ї заявника: Створити запис в таблиці 3.6, реквізити непрацездатного члена сім'ї записати з форми в таблицю 3.6.

По кнопці «#» відображається форма «PNP\_NetrudEditForm» із заголовком «Зміна інформації про непрацездатного члена сім'ї заявника» з інформацією за поточним записом в таблиці 2.18 (рис.3.15), якщо результат форми «ОК» - виконати зміну інформації про непрацездатного члена сім'ї заявника: реквізити непрацездатного члена сім'ї записати з форми в таблицю 3.6.

По кнопці «-» відобразити діалогове вікно «Підтвердження», якщо результат «ОК» виконати видалення інформації про непрацездатного члена сім'ї: виконати видалення поточного запису в таблиці 3.6.

По кнопці «<>» закрити форму.

При закритті форми управління повернути формі, що викликала.

На рисунку 3.16 зображена форма «PNP\_TrudKnigaViewForm». На рисунку 3.17 зображено додавання до запису у трудову книжку (додаток В, оператор 1094-1889).

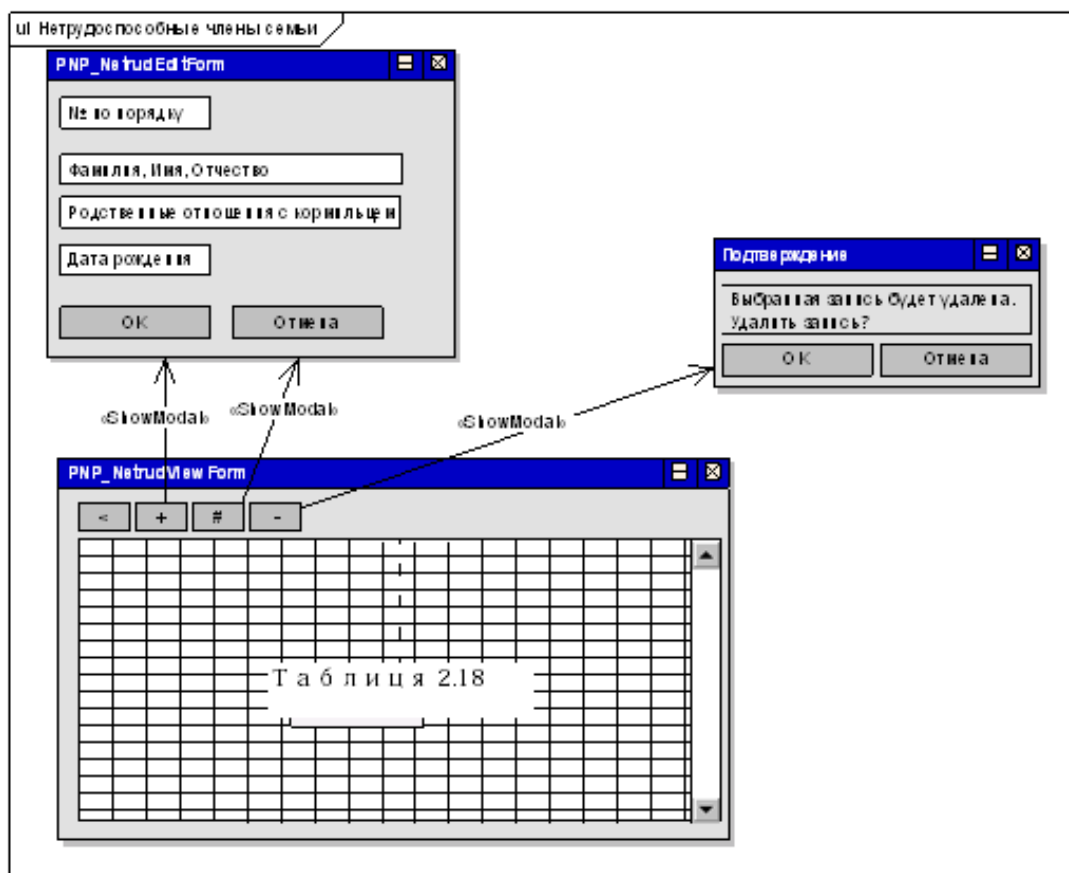


Рисунок 3.16 - Форма «PNP\_TrudKnigaViewForm»

Рисунок 3.17 – Додавання запису у трудову книжку

Перед відображенням форми виконується відображення інформації за трудовою книжкою: знайти в таблиці 3.9 запис з ZAJAVITEL\_ID, який дорівнює ZAJAVITEL.ZAJAVITEL\_ID поточного заявника. Якщо запис не знайдений, тоді створити запис в таблиці 3.9.

Результати запиту SQL4 відобразити на формі «PNP\_TrudKnigaViewForm».

У таблиці відображаються поля:

ZAPIS\_TRUD.N\_PRIEMA, ZAPIS\_TRUD.GOD\_PRIMA,  
 ZAPIS\_TRUD.MES\_PRIEMA, ZAPIS\_TRUD.CISLO\_PRIEMA,  
 ZAPIS\_TRUD.N\_UVOLN,  
 ZAPIS\_TRUD.GOD\_UVOLN,ZAPIS\_TRUD.MES\_UVOLN,  
 ZAPIS\_TRUD.CISLO\_UVOLN.

Нижче таблиці в полях форми відображається:

Відомості з прийому – ZAPIS\_TRUD.SVED\_PO\_STROKE\_PRIEM.

Підстава прийому – ZAPIS\_TRUD.OSNOVANIE\_STROKI\_PRIEM.

Відомості про звільнення – ZAPIS\_TRUD.SVED\_PO\_STROKE\_UVOLN.

Підстава звільнення – ZAPIS\_TRUD.OSNOVANIE\_STROKI\_UVOLN.

Праворуч від таблиці в полях форми відображається:

Тривалість стажу – Років – DLIT\_STAJ\_RABOT.LET.

Тривалість стажу – Місяць – DLIT\_STAJ\_RABOT.MES.

Тривалість стажу – Днів – DLIT\_STAJ\_RABOT.DNEJ.

Тривалість стажу – Коректувальні дні – DLIT\_STAJ\_RABOT.KOR\_DAY.

При переміщенні по таблиці 2.18 (рис.3.16), параметр id запиту SQL5 повинен прирівнюватися ZAPIS\_TRUD\_ID поточного запису в таблиці 2.18 (рис.3.16).

По кнопці «+» виконується додавання запису за трудовою книжкою. Визначається номер останнього запису. Для цього виконати SQL запит:

```
Select case when max (coalesce (N_PRIEMA, 0))>max (coalesce (N_UVOLN, 0))
thenmax (N_PRIEMA) else max (N_UVOLN)
endasmax_n
```

from ZAPIS\_TRUD where ZAPIS\_TRUD.trud\_kniga\_id =: id

параметр запиту id привласнити поточному запису таблиці 3.9.

Потім присвоїти змінній MAX значення результату SQL запиту (max\_n).

Створити запис в таблиці 3.10.

Після виконання відображається форма «PNP\_TrudKnigaEditForm» із заголовком «Додавання запису до трудової книжки», якщо результат форми «ОК» - виконати збереження запису трудової книжки. Реквізити стажу роботи за трудовою книжкою записати з форми в таблицю. Якщо у працівника є пільговий стаж, тоді:

Якщо Пільговий стаж по 1 списку, створити запис в таблиці 3.11.

Якщо Пільговий стаж по 2 списку, створити запис в таблиці 3.11.

Якщо є тривалість стажу роботи, тоді створити запис у таблиці 3.12.

По кнопці «#» відображається форма «PNP\_TrudKnigaEditForm» із заголовком «Зміна запису трудової книжки», якщо результат форми «ОК» - виконати зміну запису трудової книжки. Реквізити стажу роботи за трудовою книжкою записати з форми в таблицю 3.10. Якщо у працівника є пільговий стаж, тоді:

Якщо Пільговий стаж по 1 списку, знайти запис в таблиці 3.11 з рівним поточним записем трудової книжки та LGOT\_STAJ. = 1:

Якщо запис знайдено, тоді записати в таблицю 3.11.

Якщо Пільговий стаж по 2 списку, знайти запис в таблиці 3.11 з рівним поточним записем трудової книжки та LGOT\_STAJ. = 2:

Якщо запис знайдено, тоді записати в таблицю 3.11. Якщо запис не знайдено, тоді створити запис в таблиці 3.11.

Якщо є тривалість стажу роботи, тоді знайти запис в таблиці 3.12 з рівним поточним записем трудової книжки:

Якщо запис знайдено, тоді записати в таблицю 3.12.

Якщо запис не знайдено, тоді створити запис в таблиці 3.12.

По кнопці «-» відобразити діалогове вікно «Підтвердження», якщо результат «ОК» виконати видалення запису в трудовій книжці, видалення поточного запису в таблиці 3.10.

По кнопці «Розрахувати стаж» виконати розрахунок тривалості стажу роботи за записом трудової:

1) Вибрати всі записи таблиці ZAPIS\_TRUD з рівним. поточним записем трудової книжки.

2) Ввести змінні:  $NOM = 1$ ,  $GOD\_U = 0$ ,  $MES\_U = 0$ ,  $CISLO\_U = 0$ .

3) Знайти запис з  $ZAPIS\_TRUD. = NOM$ .

Перевірити:

1) Якщо  $ZAPIS\_TRUD.ISP\_RASCH = 0$ , тоді розрахувати  $NOM = NOM + 1$  і перейти на пункт 3.

2) Якщо  $ZAPIS\_TRUD. ISP\_RASCH = 1$ , тоді заносимо  $ZAPIS\_TRUD.GOD\_PRIEMA$  в  $GOD\_P$ ; заносимо  $ZAPIS\_TRUD.MES\_PRIEMA$  в  $MES\_P$ ; заносимо  $ZAPIS\_TRUD.CISLO\_PRIEMA$  в  $CISLO\_P$ ; розраховуємо  $NOM = NOM + 1$ .

3) Якщо запис не знайдено припинити виконання алгоритму.

4) Розрахувати DPU:

$$DPU = (GOD\_P \cdot 360 + MES\_P \cdot 30 + CISLO\_P) - (GOD\_U \cdot 360 + MES\_U \cdot 30 + CISLO\_U + 1).$$

5) Знайти запис  $z = NOM$ .

Якщо запис  $z = NOM$  знайдена, тоді присвоїти  $GOD\_U = ZAPIS\_TRUD.UVOLN$ ,

присвоїти  $MES\_U = ZAPIS\_TRUD.UVOLN$ , присвоїти  $CISLO\_U = ZAPIS\_TRUD.CISLO\_UVOLN$ , присвоїти  $NOM = NOM + 1$ .

Якщо запис  $z = NOM$  не знайдений, знайти запис  $z = NOM$ .

Якщо запис  $z = NOM$  знайдений, тоді  $NOM = NOM + 1$ , перехід на пункт 5.

Якщо запис  $z = NOM$  не знайдений, тоді необхідно видати користувачу повідомлення «Порушена цілісність нумерації записів». Розрахунок припинити.

6) Розрахувати наступні значення:

$$DUDPU = CISLO\_U + DPU - CISLO\_P.$$

$DUPD = CISLO\_U - CISLO\_P.$

$DUPM = MES\_U - MES\_P.$

$DUPM1 = (MES\_U - 1) - MES\_P.$

7) Розрахувати тривалості стажу по трудовій книжці:

Розрахунок тривалості днів:

Якщо  $(DPU < 0)$  і  $(DUDPU \geq 0)$ , тоді  $DAY = DUDPU.$

Якщо  $(DPU < 0)$  і  $(DUDPU < 0)$ , тоді  $DAY = DUDPU + 30.$

Якщо  $(DPU > 0)$  і  $(DUPD \geq 0)$ , тоді  $DAY = DUPD.$

Якщо  $(DPU > 0)$  і  $(DUPD < 0)$ , тоді  $DAY = DUPD + 30.$

Розрахунок тривалості місяців:

Якщо  $(DPU < 0)$  і  $(DUDPU \geq 0)$  і  $(DUPM \geq 0)$ , тоді  $MES = DUPM.$

Якщо  $(DPU < 0)$  і  $(DUDPU \geq 0)$  і  $(DUPM < 0)$ , тоді  $MES = DUPM + 12.$

Якщо  $(DPU < 0)$  і  $(DUDPU < 0)$  і  $(DUPM1 \geq 0)$ , тоді  $MES = DUPM1.$

Якщо  $(DPU < 0)$  і  $(DUDPU < 0)$  і  $(DUPM1 < 0)$ , тоді  $MES = DUPM1 + 12.$

Якщо  $(DPU > 0)$  і  $(DUPD \geq 0)$  і  $(DUPM \geq 0)$ , тоді  $MES = DUPM.$

Якщо  $(DPU > 0)$  і  $(DUPD \geq 0)$  і  $(DUPM < 0)$ , тоді  $MES = DUPM + 12.$

Якщо  $(DPU > 0)$  і  $(DUPD < 0)$  і  $(DUPM1 \geq 0)$ , тоді  $MES = DUPM1.$

Якщо  $(DPU > 0)$  і  $(DUPD < 0)$  і  $(DUPM1 < 0)$ , тоді  $MES = DUPM1 + 12.$

Розрахунок тривалості років:

$LET = ((GOD\_U \cdot 360 + MES\_U \cdot 30 + CISLO\_U) - (GOD\_P \cdot 360 + MES\_P \cdot 30 + CISLO\_P)) / 360$

LET = ціла частина от LET.

Знайти  $DLIT\_STAJ\_RABOT.ZAPIS\_TRUD\_ID$  рівний

$ZAPIS\_TRUD.ZAPIS\_TRUD\_ID$  поточного запису трудової книжки.

Якщо запис знайдено, тоді перезаписати значення в таблиці 3.12.

Якщо запис не знайдено, тоді створити запис в таблиці 3.12.

Перейти на пункт 3.

По кнопці «<>» закрити форму.

На рисунку 3.17 зображена форма «PNP\_TrudKnigaViewForm».

Checkbox «Використовувати в розрахунок» повинен бути пов'язаний з таблицею полем ISP\_RASCH:

Якщо Checkbox = true, тоді в. ISP\_RASCH записати значення «1».

Якщо Checkbox = false, тоді в. ISP\_RASCH записати значення «0».

Якщо Checkbox = false, тоді рядки «Відомості щодо звільнення», «Заснування звільнення» мають бути заборонені на редагування.

При додаванні запису до трудової книжки (при виклику форми PNP\_TrudKnigaEditForm для нового запису) поле Checkbox повинно бути true за замовчуванням.

На рисунку 3.18 зображена форма «PNP\_StajPoDrugDocViewForm».

У таблиці 2.19 (рис.3.18) відображається результат виконання запиту SQL6.

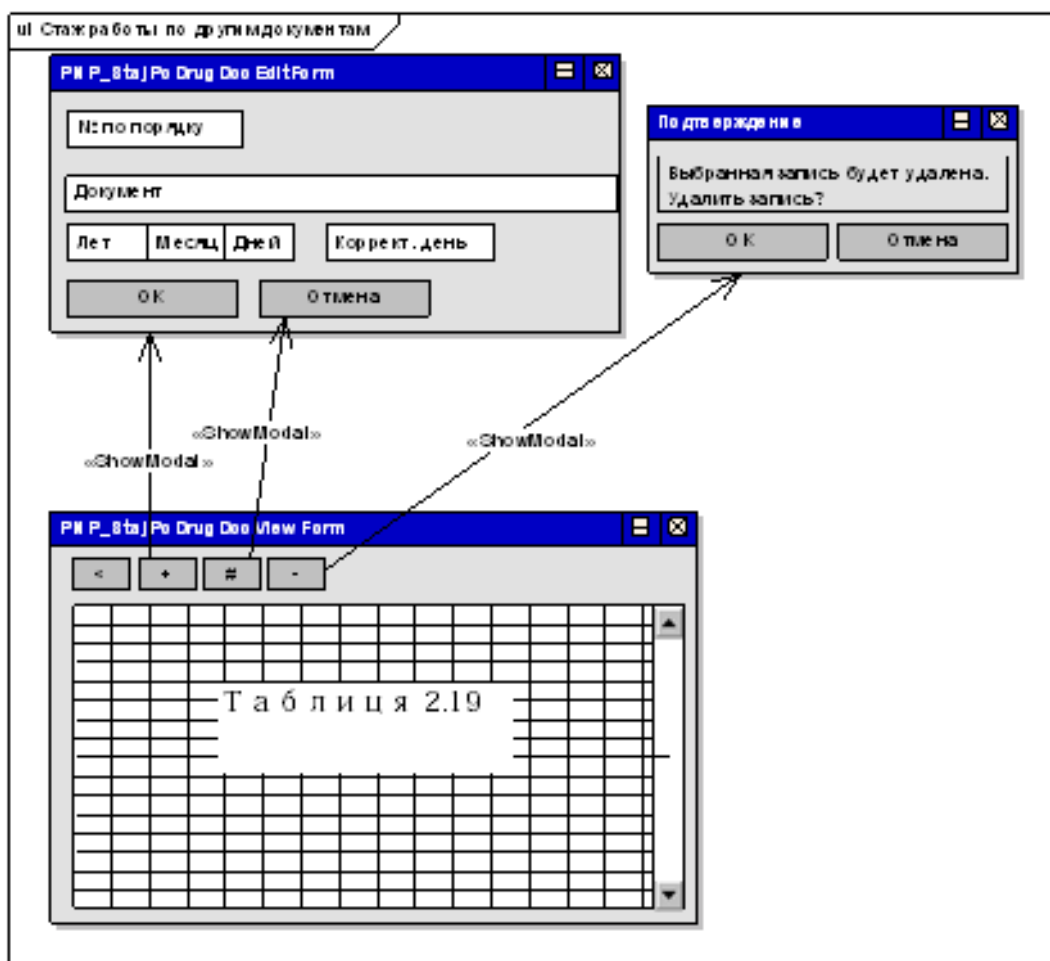


Рисунок 3.18 - Форма «PNP\_StajPoDrugDocViewForm» (додаток В, оператор 700-1092)



По кнопці «+» виконати додавання іншого документа заявника. Визначити номер останнього запису:

Виконати SQL запит:

```
Select case when max (>)>max (N_UVOLN) then max (>) else max (N_UVOLN) end
as max_n from where ZAPIS_TRUD.TRUD_KNIGA_ID =: id
```

параметр запиту id привласнити поточному запису таблиці 3.9.

2) Присвоїти змінній MAX значення результату SQL запиту (max\_n).

3) Створити запис у таблиці 3.5.

4) Записати в OTHER\_DOC.DOC значення MAX + 1.

Після виконання відобразити форму «PNP\_StajPoDrugDocEditForm» із заголовком «Додавання іншого документа заявника». Якщо результат форми «ОК» виконати збереження іншого документа заявника. Реквізити іншого документа з форми записати в таблицю 3.5.

По кнопці «#» відобразити форму «PNP\_StajPoDrugDocEditForm» із заголовком «Зміна інформації по іншому документу» з інформацією за поточним записом в таблиці 2.19. Якщо результат форми «ОК» - виконати збереження змін іншого документа заявника. Реквізити іншого документа з форми записати в таблицю 3.5.

По кнопці «-» відобразити діалогове вікно «Підтвердження». Якщо результат «ОК» виконати видалення іншого документа заявника. Виконати видалення поточного запису в таблиці 3.5.

По кнопці «<» закрити форму.

При закритті форми виконати Розрахунок загальної тривалості стажу роботи за іншими документами заявника:

1) Вибрати всі записи таблиці OTHER\_DOC, в яких ZAJAVITEL\_ID дорівнює ZAJAVITEL.ZAJAVITEL\_ID поточного заявника.

2) Перейти на перший запис. Створити змінні:

LET = 0; MES = 0; DAY = 0.

3) Присвоїти змінним наступні значення:

LET: = LET + OTHER\_DOC.LET

MES: = MES + OTHER\_DOC.MES

DAY: = DAY + OTHER\_DOC.DNEJ + OTHER\_DOC.AY

4) Якщо поточний запис не останній, тоді перейти на наступний запис і повернутися на пункт 3.

5) Розрахувати тривалість:

A1 = ціла частина від ділення (DAY/30)

A2 = ціла частина від ділення ((MES + A1) / 12)

DAY = DAY - A1 • 30

MES = (MES + A1) - A2 • 12

LET = LET + A2

6) Знайти в таблиці PREDSTAV запис, в якій ZAJAVITEL\_ID дорівнює ZAJAVITEL.ZAJAVITEL\_ID поточного заявника.

7) Записати значення в таблицю 3.7.

Потім слід виконати розрахунок загального стажу роботи:

Виконати SQL запит:

```
select t.sum_let, t.sum_mes, t.sum_day, o.sum_other_let, o.sum_other_mes,
o.sum_other_day from predstavo join      (trud_knigat join zajavitelzont.zajavitel_id
= z.zajavitel_id) ono.zajavitel_id = z.zajavitel_id where z.zajavitel_id =: id
```

параметр запиту id привласнити ZAJAVITEL\_ID поточного заявника.

Перейти на перший запис. Створити змінні:

LET = 0; MES = 0; DAY = 0.

Присвоїти змінним наступні значення:

LET: = TRUD\_KNIGA.LET + PREDSTAV\_SUM\_OTHER\_LET

MES: = TRUD\_KNIGA.MES + PREDSTAV\_SUM\_OTHER\_MES

DAY: = TRUD\_KNIGA.DAY + PREDSTAV\_SUM\_OTHER\_DAY

Розрахувати тривалість:

A1 = ціла частина від ділення (DAY/30)

A2 = ціла частина від ділення ((MES + A1) / 12)

$$\text{DAY} = \text{DAY} - A1 \cdot 30$$

$$\text{MES} = (\text{MES} + A1) - A2 \cdot 12$$

$$\text{LET} = \text{LET} + A2$$

5) Знайти в таблиці 3.7 запис, в якому ZAJAVITEL\_ID дорівнює ZAJAVITEL.ZAJAVITEL\_ID поточного заявника.

6) Записати значення в таблицю 3.7.

Після виконання управління повернути форми, що викликала.

### 3.4.4 Основні функції програми

Основні функції класу TPNP\_EditFormBase:

1) Функція SetTableName призначена для завдання імені таблиці бази даних, з якою пов'язане діалогове вікно. Всі дані, введені на діалоговому вікні, будуть зберігатися у вказану таблицю.

Об'ява функції класі:

```
procedure SetTableName (value: String);
```

Параметри:

value - ім'я таблиці

Функція SetID\_Field призначена для завдання імені ключового поля в таблиці, ім'я якого було призначено в функції SetTableName.

Об'ява функції класі:

```
procedure SetID_Field (value: String);
```

Параметри:

value - ім'я ключового поля

Функція SetID призначена для завдання значення ключового поля, ім'я якого було призначено в функції SetID\_Field.

Оголошення функції класі:

```
procedure SetID (value: Integer);
```

Параметри:

value - значення ключового поля

Функція FillFieldDataArray призначена для створення масиву відповідностей між полями таблиці, ім'я якої було призначено в функції SetTableName, та елементами введення діалогового вікна. Ця функція є абстрактною, що означає відсутність її реалізації в базовому класі. Ця функція повинна бути реалізована в дочірньому класі. Оголошення функції в класі:

```
procedure FillFieldDataArray; virtual; abstract;
```

Функція ApplyData призначена для запису введених даних на діалог в таблицю, ім'я якої було призначено в функції SetTableName. Відповідність між полями таблиці і елементами введення діалогу має бути призначено в функції FillFieldDataArray.

Оголошення функції в класі:

```
function ApplyData: Integer;
```

Результатом роботи функції є ціле число. Функція повертає -1 у випадку помилки, будь-яке інше значення, що повертається означає, що функція відпрацювала без помилок.

Функція SetNewRecord призначена для завдання режиму роботи діалогового вікна.

Об'ява функції у класі [15]:

```
procedure SetNewRecord (value: Boolean); virtual;
```

Параметри:

value - ознака створення нового запису в таблиці, ім'я якого було призначено в функції SetTableName. Можливі значення:

True - діалог працює в режимі створення нового запису;

False - діалог працює в режимі редагування існуючого запису.

Функція SetComponentText призначена для завдання значення елемента введення діалогового вікна.

Об'ява функції у класі:

```
procedure SetComponentText (Index: Integer; Value: String);
```

Параметри:

Index – індексу масивів ідповідностей між полями таблиці, ім'я якої було призначено в функції SetTableName, та елементами введення діалогового вікна;

Value - значення, яке потрібно присвоїти елементу введення, ім'я якого зазначено в комірці масиву з індексом Index.

Функція ComponentText призначена для отримання значення елемента вводу діалогового вікна [16].

Об'явафункціїукласі:

```
function ComponentText (Index: Integer): String;
```

Параметри:

Index – індексу масивів відповідностей між полями таблиці, ім'я якої було призначено в функції SetTableName, та елементами введення діалогового вікна.

Результатом роботи функції є значення рядка елемента вводу, ім'я якого зазначено в комірці масиву з індексом Index.

Функція MaxID призначена для пошуку максимального значення поля в таблиці.

Об'ява функції у класі:

```
function MaxID (TbName: String; IDFieldName: String): Integer;
```

Параметри:

TbName - ім'я таблиці, в якій необхідно виконати пошук;

IDFieldName - ім'я параметра таблиці TbName, максимальне значення якого необхідно знайти.

Результатом роботи функції є ціле число. Функція повертає -1 у випадку помилки, будь-яке інше значення, що повертається означає, що функція відпрацювала без помилок.

Основні функції класів TPNP\_PensOrganViewForm, TPNP\_PredprViewForm, TPNP\_StajPoDrugDocViewForm, TPNP\_NetrudViewForm:

Функція SpeedButton1Click є обробником події натискання кнопки і призначена для закриття діалогового вікна з передачею управління викликати

діалогове вікно.

Об'ява функції у класі:

```
procedure SpeedButton1Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція SpeedButton2Click є обробником події натискання кнопки і призначена для виклику діалогового вікна створення запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton2Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція SpeedButton3Click є обробником події натискання кнопки і призначена для виклику діалогового вікна редагування запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton3Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція SpeedButton4Click є обробником події натискання кнопки і призначена для видалення запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton4Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція FormatTable призначена для форматування поля відображення значень полів таблиці бази даних.

Оголошення функції в класі:

```
procedure FormatTable;
```

Функція FormShow є обробником події відображення діалогового вікна і призначена для завдання початкових значень локальних змінних даного діалогового

вікна.

Об'ява функції у класі:

```
procedure FormShow (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція FormClose є обробником події закриття діалогового вікна і призначена для виконання дій по завершенню роботи діалогового вікна (фінальні обчислення з занесенням результатів у базу даних, звільнення пам'яті, закриття з'єднання з сервером баз даних) [17].

Об'ява функції у класі:

```
procedure FormClose (Sender: TObject; var Action: TCloseAction);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія;

Action - тип закриття діалогового вікна (нічого не робити, приховати, мінімізувати, видалити з пам'яті). Використовується «сховати».

Основні функції класу TPNP\_ManagePredstavEditForm:

Функція SpeedButton1Click є обробником події натискання кнопки і призначена для закриття діалогового вікна з передачею управління викликати діалогове вікно.

Об'ява функції у класі:

```
procedure SpeedButton1Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція SpeedButton2Click є обробником події натискання кнопки і призначена для виклику діалогового вікна створення запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton2Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція `SpeedButton3Click` є обробником події натискання кнопки і призначена для виклику діалогового вікна редагування запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton3Click (Sender: TObject);
```

Параметри:

`Sender` - вказівник на екземпляр класу, від якого прийшла подія.

Функція `SpeedButton4Click` є обробником події натискання кнопки і призначена для видалення запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton4Click (Sender: TObject);
```

Параметри:

`Sender` - вказівник на екземпляр класу, від якого прийшла подія.

Функція `bSearchClick` є обробником події натискання кнопки і призначена для виконання пошуку записів в таблиці бази даних по вказаним критеріям.

Об'ява функції у класі::

```
Procedure bSearchClick (Sender: TObject);
```

Параметри:

`Sender` - вказівник на екземпляр класу, від якого прийшла подія.

Функція `bPrintClick` є обробником події натискання кнопки і призначена для формування документа у форматі MS Word для його подальшої відправки на друк.

Об'ява функції у класі:

```
procedure bPrintClick (Sender: TObject);
```

Параметри:

`Sender` - вказівник на екземпляр класу, від якого прийшла подія.

Основні функції класу `TPNP_TrudKnigaViewForm`:

Функція `SpeedButton1Click` є обробником події натискання кнопки і призначена для закриття діалогового вікна з передачею управління викликати діалогове вікно.

Об'ява функції у класі:



```
procedure SpeedButton1Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція SpeedButton2Click є обробником події натискання кнопки і призначена для виклику діалогового вікна створення запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton2Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція SpeedButton3Click є обробником події натискання кнопки і призначена для виклику діалогового вікна редагування запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton3Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція SpeedButton4Click є обробником події натискання кнопки і призначена для видалення запису таблиці бази даних.

Об'ява функції у класі:

```
procedure SpeedButton4Click (Sender: TObject);
```

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція UpDateTables призначена для оновлення даних у полях відображення значень полів таблиці бази даних.

Оголошення функції в класі:

```
procedure UpDateTables;
```

Функція bCalculationClick є обробником події натискання кнопки і призначена для виконання обчислень за розрахунок стажу та занесення результатів в базу даних.

Об'ява функції у класі:

`procedure CalculationClick (Sender: TObject);`

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція `FormShow` є обробником події відображення діалогового вікна і призначена для завдання початкових значень локальних змінних даного діалогового вікна.

Об'ява функції у класі:

`procedure FormShow (Sender: TObject);`

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія.

Функція `FormClose` є обробником події закриття діалогового вікна і призначена для виконання дій по завершенню роботи діалогового вікна (фінальні обчислення з занесенням результатів у базу даних, звільнення пам'яті, закриття з'єднання з сервером баз даних).

Об'ява функції у класі:

`procedure FormClose (Sender: TObject; var Action: TCloseAction);`

Параметри:

Sender - вказівник на екземпляр класу, від якого прийшла подія;

Action - тип закриття діалогового вікна (нічого не робити, приховати, мінімізувати, видалити з пам'яті). Використовується «сховати».

### 3.5 Запити SQL

Нижче наведені основні SQL– запити;

SQL1 – Довідник підприємств:

```
Select a.nazvanie, a.indeks, a.nas_punkt, a.ulica, a.dom from predpr a
```

SQL2 – Довідник пенсійних органів:

```
Select a.nazvanie, a.nas_punkt, a.oblast from pens_organ a
```

SQL3 – Непрацездатні члени сім'ї:

```
Select a.f_I_o, a.rodstven_svaz, a.data_rojd from netrud a where
netrud.zajavitel_id = :id
```

Параметр запису id присвоїти ZAJAVITEL\_ID поточного запису таблиці ZAJAVITEL.

SQL4 – Стаж роботи за трудовою книжкою:

```
selectt.data_zapoln, z.n_priema, z.god_priema, z.mes_priema, z.cislo_priema,
z.sved_po_stroke_priem, z.osnovanie_stroki_priem,z.n_uvoln, z.god_uvoln, z.mes_uvoln,
z.cislo_uvoln, z.sved_po_stroke_uvol, z.osnovanie_stroki_uvol,dl.let, dl.mes, dl.dnej,
dl.kor_day
```

```
fromdlit_staj_rabot dl right outer join(trud_kniga t joinzapis_trud z on
t.trud_kniga_id=z.trud_kniga_id)on dl.zapis_trud_id=z.zapis_trud_id
```

```
wheret.zajavitel_id = :id
```

параметр запису id привласнити ZAJAVITEL\_ID поточного запису таблиці ZAJAVITEL.

SQL5 – Тривалість пільгового стажу:

```
select l.nomer_lgotnogo_spiska, l.let, l.mes, l.dnej from lgot_staj l where
l.zapis_trud_id= :id
```

параметр запису id привласнити ZAPIS\_TRUD\_ID поточного запису таблиці ZAPIS\_TRUD.

SQL6 – Трудовий стаж за іншими документами:

```
Select a.n_doc, a.document, a.let, a.mes, a.dnej, a.kor_day
```

from other\_doc a where other\_doc.zajavitel\_id = :id

параметр запису id привласнити ZAJAVITEL\_ID поточного запису таблиці ZAJAVITEL.

## **4. ОХОРОНА ПРАЦІ**

В даному розділі проведено аналіз потенційних небезпечних та шкідливих виробничих факторів, причин пожеж. Розглянуті заходи, які дозволяють забезпечити гігієну праці і виробничу санітарію. На підставі аналізу розроблені заходи з техніки безпеки та рекомендації з пожежної профілактики.

Завданням даної роботи бакалавра було створення програмного забезпечення для розрахунку стажу роботи та нарахування пенсій. Так як в процесі проектування використовувалося комп'ютерне обладнання, то аналіз потенційно небезпечних і шкідливих виробничих чинників виконується для персонального комп'ютера, на якому буде виконуватися розробка.

### **4.1 Загальні питання з охорони праці**

Умови праці на робочому місці, безпека технологічних процесів, машин, механізмів, устаткування та інших засобів виробництва, стан засобів колективного та індивідуального захисту, що використовуються працівником, а також санітарно-побутові умови повинні відповідати вимогам нормативних актів про охорону праці. В законі України «Про охорону праці» визначається, що охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

### **4.2 Аналіз стану умов праці**

Робота над створенням програмного забезпечення для розрахунку стажу роботи та нарахування пенсій проходитиме в приміщенні багатоквартирного будинку. Для даної роботи достатньо однієї людини, для якої надано робоче місце зі стаціонарним комп'ютером.

#### 4.2.1 Вимоги до приміщень

Геометричні розміри приміщення зазначені в табл. 4.1.

Таблиця 4.1 – Розміри приміщення

Найменування	Значення
Довжина, м	5
Ширина, м	5
Висота, м	3
Площа, м <sup>2</sup>	25
Об'єм, м <sup>3</sup>	75

Згідно з [14] розмір площі для одного робочого місця оператора персонального комп'ютера має бути не менше 6 кв. м, а об'єм — не менше 20 куб. м. Отже, дане приміщення цілком відповідає зазначеним нормам.

#### 4.2.2 Вимоги до організації місця праці

При порівнянні відповідності характеристик робочого місця нормативним основні вимоги до організації робочого місця за [12] (табл. 4.2) і відповідними фактичними значеннями для робочого місця, констатуємо повну відповідність.

Таблиця 4.2 - Характеристики робочого місця

Найменування параметра	Фактичне значення	Нормативне значення
Висота робочої поверхні, мм	770	680 ÷ 800
Висота простору для ніг, мм	750	не менше 600
Ширина простору для ніг, мм	550	не менше 500
Глибина простору для ніг, мм	700	не менше 650
Висота поверхні сидіння, мм	450	400 ÷ 500
Ширина сидіння, мм	450	не менше 400
Глибина сидіння, мм	470	не менше 400
Висота поверхні спинки, мм	400	не менше 300
Ширина опорної поверхні спинки, мм	500	не менше 380
Радіус кривини спинки в горизонтальній площині, мм	400	400
Відстань від очей до екрану дисплея, мм	750	700 ÷ 800

### **4.2.3 Навантаження та напруженість процесу праці**

Під час виконання випускної роботи бакалавра:

за фізичним навантаженням робота відноситься до категорії легкі роботи (Ia), її виконують сидячи з періодичним ходінням. Щодо характеру організування виконання дипломної роботи, то він підпадає під нав'язаний режим, оскільки певні розділи роботи необхідно виконати у встановлені конкретні терміни.

Рекомендовано застосування екранних фільтрів, локальних світлофільтрів (засобів індивідуального захисту очей) та інших засобів захисту, а також інші профілактичні заходи наведені в [12].

Роботу за дипломним проектом визнано, таку, що займає 50% часу робочого дня та за восьмигодинної робочої зміни рекомендовано встановити додаткові регламентовані перерви - для розробників програм тривалістю 15 хв. через кожную годину роботи;

## **4.3. Виробнича санітарія**

На підставі аналізу небезпечних та шкідливих факторів при виробництві (експлуатації), пожежної безпеки можуть бути надалі вирішені питання необхідності забезпечення працюючих достатньою кількістю освітлення, вентиляції повітря, організації заземлення, тощо.

### **4.3.1. Аналіз небезпечних та шкідливих факторів при виробництві (експлуатації) виробу**

Аналіз небезпечних та шкідливих виробничих факторів виконується у табличній формі (табл. 4.3). Роботу, пов'язану з ЕОП з ВДТ, у тому числі на тих, які мають робочі місця, обладнані ЕОМ з ВДТ і ПП, виконують із забезпеченням

виконання НПАОП 0.00.-1.28-10 [20], які встановлюють вимоги безпеки до обладнання робочих місць, до роботи із застосуванням ЕОМ з ВДТ і ПП.

Основними робочими характеристиками персонального комп'ютера є:

- робоча напруга  $U=+220\text{В} \pm 5\%$ ;
- робочий струм  $I=2\text{А}$ ;
- споживана потужність  $P=350\text{ Вт}$ .

Таблиця 4.3 – Аналіз небезпечних і шкідливих виробничих факторів

Небезпечні і шкідливі виробничі фактори	Джерела факторів (види робіт)	Кількісна оцінка	Нормативні документи
<b>фізичні</b>			
підвищена температура поверхонь обладнання	експлуатація ЕОМ, принтерів, сканерів чи/або серверного обладнання для роботи	2	[14]
підвищений рівень іонізуючого випромінювання в робочій зоні	-//-	2	[14] [24]
підвищений рівень електромагнітного випромінювання	-//-	2	[24]
підвищений рівень напруги електричної мережі, замикання якої може відбутися через тіло людини	-//-	4	[25] [14]
підвищений рівень статичної електрики	-//-	2	[25]
підвищена напруженість електричного поля	-//-	2	[24]
підвищена напруженість магнітного поля	-//-	2	[24]
недостатність природного світла	порушення умов праці (вимог до приміщень)	2	[11]
недостатнє освітлення робочої зони	порушення гігієнічних параметрів виробничого середовища	3	[11]
підвищена яскравість світла	порушення умов праці (організації місця праці- налагодження моніторів)	1	[12]
понижена контрастність	-//-	1	[12]



Продовження таблиці 4.3

<b><i>психофізіологічні:</i></b>			
- нервово-психічна перевантаження (розумове, перенапруження аналізаторів-зорових)	- пошук інформації для постановки теми; - пошук та аналіз аналогів і літератури; - пошук наявних технологій, моделювання та аналіз алгоритмів; - виконання роботи за темою диплома, тестування; - оформлення роботи	4	[26] [12]
- фізичні (статичне – сидіння)	порушення умов праці (організації місця праці - сидіння користувача) та організації робочого часу - безпервна робота)	2	[26] [12]

Робочі місця мають відповідати вимогам Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин, затверджених постановою Головного державного санітарного лікаря України від 10.12.98 N 7 [12].

#### **4.3.2 Пожежна безпека**

Небезпека розвитку пожежі на обчислювальному центрі обумовлюється застосуванням розгалужених систем електроживлення ЕОМ, вентиляції і кондиціонування.

Запобігти утворенню горючого середовища (замінити горючі речовини і матеріали на негорючі і важкогорючі) не надається технічно можливим. Тому проектом передбачаються способи і засоби запобігання утворення (або внесення) в горюче середовище джерел запалювання, таких як:

- 1) застосування електроустаткування, відповідної пожежонебезпечної і вибухонебезпечної зонам відповідно до ПУЕ;
- 2) застосування в конструкції швидкодійних засобів захисного відключення можливих джерел запалення;
- 3) виключення можливості появи іскрового розряду в горючому середовищі з енергією, рівної і вище мінімальної енергії запалення.

Згідно [18] таке приміщення, площею 25 м<sup>2</sup>, відноситься до категорії "В" (пожежонебезпечної) та для протипожежного захисту в ньому проектом передбачено устаткування автоматичною пожежною сигналізацією із застосуванням датчиків-сповіщувачів РІД-1 (сповіщувач димовий ізоляційний) в кількості 1 шт., і застосуванням первинних засобів пожежогасіння.

Простори усередині приміщень в межах, яких можуть утворюватися або знаходитися пожежонебезпечні речовини і матеріали відповідно до [18] відносяться до пожежонебезпечної зони класу П-Па. Це обумовлено тим, що в приміщенні знаходяться тверді горючі та важкозаймисті речовини та матеріали. Приміщенню, у якому розташоване робоче місце, присвоюється II ступень вогнестійкості.

Продуктами згорання, що виділяються на пожежі, є: окис вуглецю; сірчистий газ; окис азоту; синильна кислота; акромін; фосген; хлор і ін. При горінні пластмас, окрім звичних продуктів згорання, виділяються різні продукти термічного розкладання: хлорангідридні кислоти, формальдегіди, хлористий водень, фосген, синильна кислота, аміак, фенол, ацетон, стирол [10].

### **4.3.3 Електробезпека**

На робочому місці виконуються наступні вимоги електробезпеки: ПК, периферійні пристрої та устаткування для обслуговування, електропроводи і кабелі за виконанням та ступенем захисту відповідають класу зони за ПУЕ (правила улаштування електроустановок), мають апаратуру захисту від струму короткого замикання та інших аварійних режимів. Лінія електромережі для живлення ПК, периферійних пристроїв і устаткування для обслуговування, виконана як окрема групова три- провідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників. Нульовий захисний провідник використовується для заземлення (занулення) електроприймачів. Штепсельні з'єднання та електророзетки крім контактів фазового та нульового робочого провідників мають спеціальні контакти для підключення нульового захисного

провідника. Електромережа штепсельних розеток для живлення персональних ПК укладено по підлозі поруч зі стінами відповідно до затвердженого плану розміщення обладнання та технічних характеристик обладнання. Металеві труби та гнучкі металеві рукави заземлені. Захисне заземлення включає в себе заземлюючих пристроїв і провідник, який з'єднує заземлюючий пристрій з обладнанням, яке заземлюється - заземлюючий провідник.

#### 4.4 Гігієнічні вимоги до параметрів виробничого середовища

##### 4.4.1 Мікроклімат

Мікроклімат робочих приміщень – це клімат внутрішнього середовища цих приміщень, що визначається діючої на організм людини з'єднанням температури, вологості, швидкості переміщення повітря. Оптимальні значення для температури, відносної вологості й рухливості повітря для зазначеного робочого місця відповідають [13] і наведені в табл. 4.4:

Таблиця 4.4 – Норми мікроклімату робочої зони об'єкту

Період року	Категорія робіт	Температура С <sup>0</sup>	Відносна вологість %	Швидкість руху повітря, м/с
Холодна	легка-1 а	22 - 24	40 – 60	0,1
Тепла	легка-1 а	23 - 25	40 – 60	0,1

У приміщенні на робочому місці забезпечуються оптимальні значення параметрів мікроклімату. Дане приміщення обладнане системою опалення, кондиціонування повітря. Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

#### 4.4.2 Освітлення

Світло є природною умовою існування людини. Воно впливає на стан вищих психічних функцій і фізіологічні процеси в організмі. Хороше освітлення діє тонізуюче, створює гарний настрій, покращує протікання основних процесів вищої нервової діяльності.

У приміщенні, де розташовані ЕОМ передбачається природне бічне освітлення, рівень якого відповідає [11]. Джерелом природного освітлення є сонячне світло. Регулярно повинен проводитися контроль освітленості, який підтверджує, що рівень освітленості задовольняє ДБН і для даного приміщення в світлий час доби достатньо природного освітлення.

*Розрахунок освітлення.*

Для виробничих та адміністративних приміщень світловий коефіцієнт приймається не менше  $1/8$ , в побутових –  $1/10$ :

$$S_b = \left( \frac{1}{5} \div \frac{1}{10} \right) \cdot S_n, \quad (4.1)$$

де  $S_b$  – площа віконних прорізів,  $m^2$ ;

$S_n$  – площа підлоги,  $m^2$ .

$$S_n = a \cdot b = 5 \cdot 5 = 25 \text{ м}^2,$$

$$S = 1/8 \cdot 25 = 3,125 \text{ м}^2.$$

Приймаємо 2 вікна площею  $S=1,6 \text{ м}^2$  кожне.

Розрахунок штучного освітлення виробляється по коефіцієнтах використання світлового потоку, яким визначається потік, необхідний для створення заданої освітленості при загальному рівномірному освітленні.

Розрахунок кількості світильників  $n$  виробляється по формулі (4.2):

$$n = \frac{E \cdot S \cdot Z \cdot K}{F \cdot U \cdot M}, \quad (4.2)$$

де  $E$  – нормована освітленість робочої поверхні, визначається нормами – 300 лк;

$S$  – освітлювана площа, м<sup>2</sup>;  $S = 25$  м<sup>2</sup>;

$Z$  – поправочний коефіцієнт світильника ( $Z = 1,15$  для ламп розжарювання та ДРЛ;  $Z = 1,1$  для люмінесцентних ламп) приймаємо рівним 1,1;

$K$  – коефіцієнт запасу, що враховує зниження освітленості в процесі експлуатації – 1,5;

$U$  – коефіцієнт використання, залежний від типу світильника, показника індексу приміщення і т.п. – 0,575

$M$  – число люмінесцентних ламп в світильнику – 2;

$F$  – світловий потік лампи – 3332 лм (для ПСО 30-2).

Підставивши числові значення у формулу (4.2), отримуємо:

$$n = \frac{(300 \times 25 \times 1,1 \times 1,5)}{3332 \times 0,575 \times 2} = 3,23$$

Приймаємо освітлювальну установку, яка складається з 1-го світильника, який складаються з 60 -ти світлодіодів загальною потужністю 34 Вт, напругою – 220 В.

#### 4.4.3 Шум

Рівень шуму, що супроводжує роботу користувачів персональних комп'ютерів (зумовлений як роботою системних блоків, клавіатури, так і друкуванням на принтерах, а також зовнішніми чинниками), коливається у межах 50–65 дБА [13]. У залах опрацювання інформації та комп'ютерного набору рівні шуму не повинні перевищувати 65 дБА.

#### **4.4.4 Вентилювання**

У приміщенні, де знаходяться ЕОМ, повітрообмін реалізується за допомогою природної організованої вентиляції (вентиляційні шахти), тобто при  $V$  приміщення  $> 40$  м<sup>3</sup> на одного працюючого допускається природна вентиляція. Цей метод забезпечує приток потрібної кількості свіжого повітря, що визначається в СНіП.

Також має здійснюватися провітрювання приміщення, в залежності від погодних умов, тривалість повинна бути не менше 10 хв. Найкращий обмін повітря здійснюється при наскрізному провітрюванні.

#### **4.5 Заходи з організації виробничого середовища та попередження виникнення надзвичайних ситуацій**

Відповідно до санітарно-гігієнічних нормативів та правил експлуатації обладнання передбачено наступні заходи безпеки під час експлуатації персонального комп'ютера та периферійних пристроїв:

- правильне організування місця праці та дотримання оптимальних режимів праці та відпочинку під час роботи з ПК;
- експлуатацію сертифікованого обладнання;
- дотримання заходів електробезпеки;
- забезпечення оптимальних параметрів мікроклімату;
- забезпечення раціонального освітлення місця праці (освітленість робочого місця не перевищувала  $2/3$  нормальної освітленості приміщення);
- облаштовуючи приміщення для роботи з ПК, потрібно передбачити припливно-витяжну вентиляцію або кондиціонування повітря.

Крім того, потрібно дотримуватися правил безпеки під час експлуатації інших електричних приладів та вимоги безпеки при надзвичайних ситуаціях.

### Розрахунок захисного заземлення (забезпечення електробезпеки будівлі).

Згідно з класифікацією приміщень за ступенем небезпеки ураження електричним струмом [19], приміщення в якому проводяться всі роботи відноситься до першого класу (без підвищеної небезпеки). Коефіцієнт використання вертикальних заземлювачів  $\eta_v$  в залежності від розміщення заземлювачів та їх кількості знаходиться в межах 0,4...0,99. Взаємну екрануючу дію горизонтального заземлювача (з'єднувальної смуги) враховують за допомогою коефіцієнта використання горизонтального заземлювача  $\eta_c$ .

Послідовність розрахунку.

1) Визначається необхідний опір штучних заземлювачів  $R_{шт.з.}$ :

$$R_{шт.з.} = \frac{R_d \cdot R_{пр.з.}}{R_{пр.з.} - R_d}, \quad (4.3)$$

де  $R_{пр.з.}$  – опір природних заземлювачів;  $R_d$  – допустимий опір заземлення. Якщо природні заземлювачі відсутні, то  $R_{шт.з.} = R_d$ .

Підставивши числові значення у формулу (4.3), отримуємо:

$$2) R_{шт.з.} = \frac{4 \cdot 40}{40 - 4} \approx 4 \text{ Ом}$$

3) Опір заземлення в значній мірі залежить від питомого опору ґрунту  $\rho$ , Ом·м. Приблизне значення питомого опору глини приймаємо  $\rho = 40$  Ом·м (табличне значення).

4) Розрахунковий питомий опір ґрунту,  $\rho_{розр.}$ , Ом·м, визначається відповідно для вертикальних заземлювачів  $\rho_{розр.в.}$ , і горизонтальних  $\rho_{розр.г.}$ , Ом·м за формулою:

$$\rho_{розр.} = \psi \cdot \rho, \quad (4.4)$$

де  $\psi$  – коефіцієнт сезонності для вертикальних заземлювачів I кліматичної зони з нормальною вологістю землі, приймається для вертикальних заземлювачів  $\rho_{розр.в.} = 1,7$  і горизонтальних  $\rho_{розр.г.} = 5,5$  Ом·м.

$$\rho_{\text{розр.в}} = 1,7 \cdot 40 = 68 \text{ Ом}\cdot\text{м}$$

$$\rho_{\text{розр.г}} = 5,5 \cdot 40 = 220 \text{ Ом}\cdot\text{м}$$

5) Розраховується опір розтікання струму вертикального заземлювача  $R_{\text{в}}$ , Ом, за формулою (4.5).

$$R_{\text{в}} = \frac{\rho_{\text{розр.в}}}{2 \cdot \pi \cdot l_{\text{в}}} \cdot \left( \ln \frac{2 \cdot l_{\text{в}}}{d_{\text{ст}}} + \frac{1}{2} \cdot \ln \frac{4 \cdot t + l_{\text{в}}}{4 \cdot t - l_{\text{в}}} \right), \quad 4.5)$$

де  $l_{\text{в}}$  – довжина вертикального заземлювача (для труб - 2–3 м;  $l_{\text{в}}=3$  м);

$d_{\text{ст}}$  – діаметр стержня (для труб - 0,03–0,05 м;  $d_{\text{ст}}=0,05$  м);

$t$  – відстань від поверхні землі до середини заземлювача, яка визначається за формулою (4.6):

$$t = h_{\text{в}} + \frac{l_{\text{в}}}{2}, \quad 4.6)$$

де  $h_{\text{в}}$  – глибина закладання вертикальних заземлювачів (0,8 м); тоді

$$t = 0,8 + \frac{3}{2} = 2,3 \text{ м}$$

$$R_{\text{в}} = \frac{68}{2 \cdot \pi \cdot 3} \cdot \left( \ln \frac{2 \cdot 3}{0,05} + \frac{1}{2} \cdot \ln \frac{4 \cdot 2,3 + 3}{4 \cdot 2,3 - 3} \right) = 18,5 \text{ Ом}$$

б) Визначається теоретична кількість вертикальних заземлювачів  $n$  штук, без урахування коефіцієнта використання  $\eta_{\text{в}}$ :

$$n = \frac{2 \cdot R_{\text{в}}}{R_{\text{д}}} = \frac{2 \cdot 18,5}{4} = 9,25 \quad 4.7)$$

$I$  визначається коефіцієнт використання вертикальних електродів групового заземлювача без врахування впливу з'єднувальної стрічки  $\eta_{\text{в}} = 0,57$  (табличне значення).

7) Визначається необхідна кількість вертикальних заземлювачів з урахуванням коефіцієнта використання  $n_{\text{в}}$ , шт:



$$n_B = \frac{2 \cdot R_B}{R_d \cdot \eta_B} = \frac{2 \cdot 18,5}{4 \cdot 0,57} = 16,2 \approx 16 \quad (4.8)$$

8) Визначається довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м:

$$l_c = 1,05 \cdot L_B \cdot (n_B - 1), \quad (4.9)$$

де  $L_B$  – відстань між вертикальними заземлювачами, (прийняти за  $L_B = 3$  м);  
 $n_B$  – необхідна кількість вертикальних заземлювачів.

$$l_c = 1,05 \cdot 3 \cdot (16 - 1) \approx 48 \text{ м}$$

9) Визначається опір розтіканню струму горизонтального заземлювача (з'єднувальної стрічки)  $R_\Gamma$ , Ом:

$$R_\Gamma = \frac{\rho_{\text{розр.г}}}{2 \cdot \pi \cdot l_c} \cdot \ln \frac{2 \cdot l_c^2}{d_{\text{см}} \cdot h_\Gamma}, \quad (4.10)$$

де  $d_{\text{см}}$  – еквівалентний діаметр смуги шириною  $b$ ,  $d_{\text{см}} = 0,95b$ ,  $b = 0,15$  м;

$h_\Gamma$  – глибина закладання горизонтальних заземлювачів (0,5 м);

$l_c$  – довжина з'єднувальної стрічки горизонтального заземлювача  $l_c$ , м

$$R_\Gamma = \frac{220}{2 \cdot \pi \cdot 48} \cdot \ln \frac{2 \cdot 48^2}{0,95 \cdot 0,15 \cdot 0,5} = 8,1 \text{ Ом}$$

10) Визначається коефіцієнт використання горизонтального заземлювача  $\eta_c$  відповідно до необхідної кількості вертикальних заземлювачів  $n_B$ .

Коефіцієнт використання з'єднувальної смуги  $\eta_c = 0,3$  (табличне значення).

11) Розраховується результуючий опір заземлювального електроду з урахуванням з'єднувальної смуги:

$$R_{\text{заг}} = \frac{R_B \cdot R_\Gamma}{R_B \cdot \eta_c + R_\Gamma \cdot n_B \cdot \eta_B} \leq R_d. \quad (4.18)$$

Висновок: дане захисне заземлення буде забезпечувати електробезпеку будівлі, так як виконується умова:  $R_{\text{заг}} < 4$  Ом, а саме:

$$R_{\text{зар}} = \frac{18,5 \cdot 8,1}{18,5 \cdot 0,3 + 8,1 \cdot 16 \cdot 0,57} = 1,9 \leq R_{\text{д}}$$

1) При виникненню пожеж при роботі на ПЕОМ від таких можливими джерел запалювання як:

- іскри і дуги коротких замикань;
- перегрів провідників, резисторів та інших радіодеталей ПЕОМ, від тривалої перевантаження та наявності перехідного опору;
- іскри при розмиканні і розмиканні ланцюгів;
- розряди статичної електрики;
- необережному поводженню з вогнем, а також вибухи газо-повітряних і паро-повітряних сумішей.

#### **Висновки до розділу 4**

В результаті проведеної роботи було зроблено аналіз умов праці, шкідливих та небезпечних чинників, з якими стикається робітник. Було визначено параметри і певні характеристики приміщення для роботи над запропонованим проектом, описано, які заходи потрібно зробити для того, щоб дане приміщення відповідало необхідним нормам і було комфортним і безпечним для робітника. Приведені рекомендації щодо організації робочого місця, а також важливу інформацію щодо пожежної та електробезпеки. Були наведені розміри приміщення та наведено значення температури, вологості й рухливості повітря, необхідна кількість і потужність ламп та інші параметри, значення яких впливає на умови праці робітника, а також – наведені інструкції з охорони праці, техніки безпеки при роботі на комп'ютері.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. - М.: Финансы и статистика, 1998.–176 с.
2. Орлов, С.А. Технологии разработки программного обеспечения / С.А. Орлов–СПб.: Питер, 2002.–464 с.
3. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс / Гарсиа-Молина Г, Ульман Дж, Уидом Дж. — М.: "Вильямс", 2003. – 229 с.
4. Дейт. К. Дж. Введение в системы баз данных / К. Дж. Дейт. — "Вильямс", 2001. – 426 с.
5. Харрингтон Д. Л Проектирование реляционных баз данных. Просто и доступно / Д. Л. Харрингтон. – М.: ЛОРИ, 2000. – 277 с.
6. Коннолли Т. М, Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. М. Коннолли, К. Бегг. – М.: Издательский дом "Вильямс", 2003. – 261 с.
7. Голованов М., Веселов Е. Создание компонентов в среде Delphi. Руководство разработчика / М. Голованов. – СПб.: Издательство "ВНУ СПб", 2004. – 320 с.
8. Методичні вказівки до виконання і захисту дипломного проекту (роботи) бакалавра спеціальностей 122 "Комп'ютерні науки та інформаційні технології", 123 "Комп'ютерна інженерія", 125 "Кібербезпека" (за напрямками 6.050101 "Комп'ютерні науки", 6.050102 "Комп'ютерна інженерія", 6.170101 "Безпека інформаційних і комунікаційних систем") для здобувачів вищої освіти денної і заочної форм навчання / Уклад.: Скарга-Бандурова І.С., Барбарук В.М., Кардашук В.С. – Сєвєродонецьк: СНУ ім. В. Даля, 2018. – 60 с.
9. Методичні вказівки до виконання розділу дипломного проекту (роботи) бакалавра "Охорона праці та безпека в надзвичайних ситуаціях" (для студентів

денної та заочної форм навчання за спеціальностями 122 "Комп'ютерні науки та інформаційні технології", 123 "Комп'ютерна інженерія", 125 "Кібербезпека" (за напрямками 6.050101 "Комп'ютерні науки", 6.050102 "Комп'ютерна інженерія", 6.170101 "Безпека інформаційних і комунікаційних систем")) / Уклад.: Критська Я.О. – Під ред. Скарги-Бандурової І.С. – Сєверодонецьк: вид-во СНУ ім. В. Даля, 2017. – 71 с.

10. ГОСТ 12.1.044-89 ССБТ. Пожежовибухонебезпека речовин і матеріалів.  
Номенклатура показників і методи їх визначення

11. ДБН В.2.5-28:2015 Природне і штучне освітлення

12. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин

13. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку

14. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих

15. НПАОП 0.00-4.12-05 Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці

16. НПАОП 0.00-4.15-98 Про розробку інструкцій з охорони праці

17. НПАОП 0.00-6.03-93 Порядок опрацювання та затвердження власником нормативних актів про охорону праці

18. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою

19. НПАОП 40.1-1.01-97 Правила безопасной эксплуатации электроустановок

20. НПАОП 40.1-1.32-01 Правила устройства электроустановок.  
Электрооборудование специальных установок

21. ДСН 3.3.6.039-99 Санітарні норми виробничої загальної та локальної вібрації

22. ДСТУ ГОСТ 12.1.012-90 ССБТ. Вібраційна безпека. Загальні вимоги

23. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування

24. ГОСТ 12.1.006-84 ССБТ. Електромагнітні поля радіочастот. Загальні вимоги безпеки. Допустимі рівні на робочих місцях і вимоги до проведення контролю

25. ГОСТ 12.1.030-81 ССБТ. Електробезпечність. Захисне заземлення. Занулення

26. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно- обчислювальних машин

## ВИСНОВКИ

В данному дипломному проєкті розроблено програмне забезпечення для розрахунку стажу роботи та нарахування пенсій. Дана програма значно полегшить роботу працівників відділу кадрів, а також скоротить час виконання операцій.

В першому розділі зроблено аналіз операцій, які виконує працівник відділу кадрів до автоматизації та аналіз операцій, які буде виконувати робітник відділу кадрів після автоматизації. Сформульоване технічне завдання.

У другому розділі виконано вибір засобів розробки програмного забезпечення. Виділені важливі функції і можливості мови програмування Delphi для отримання надійного ПЗ.

У третьому розділі на основі аналізу технічного завдання, структури і діяльності працівників відділу кадрів побудована структура програмного забезпечення; база даних, яка призначена для зберігання інформації; розроблено і описано програмне забезпечення та наведені інструкції оператора.

У четвертому розділі проведений аналіз потенційних небезпечних і шкідливих факторів на робочому місці оператора ЕОМ. Описані заходи щодо техніки безпеки, заходи щодо забезпечення виробничої санітарії і гігієни праці, а також рекомендації з протипожежної безпеки. Проведено розрахунок природного та штучного освітлення.

Розробка програмного забезпечення дозволила вирішити основні проблеми, котрі потребували автоматизації: зменшення витрат значної кількості робочого часу; працівник відділу кадрів може виконувати значно більший обсяг роботи; зменшення рівня можливості здійснення помилки в оформленні необхідного документа; покращення робочих умов працівника.

Розроблена система дозволяє полегшити роботу персоналу, зменшити витрати часу та покращити якість виконуваної роботи.

Отримані результати розробки свідчать про ефективність та доцільність впровадження та використання на підприємстві розробленого ПЗ.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАНЬ

1. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. - М.: Финансы и статистика, 1998.–176 с.
2. Орлов, С.А. Технологии разработки программного обеспечения / С.А. Орлов–СПб.: Питер, 2002.–464 с.
3. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс / Гарсиа-Молина Г, Ульман Дж, Уидом Дж. — М.: "Вильямс", 2003. – 229 с.
4. Дейт. К. Дж. Введение в системы баз данных / К. Дж. Дейт. — "Вильямс", 2001. – 426 с.
5. Харрингтон Д. Л Проектирование реляционных баз данных. Просто и доступно / Д. Л. Харрингтон. – М.: ЛОРИ, 2000. – 277 с.
6. Коннолли Т. М, Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. М. Коннолли, К. Бегг. – М.: Издательский дом "Вильямс", 2003. – 261 с.
7. Голованов М., Веселов Е. Создание компонентов в среде Delphi. Руководство разработчика / М. Голованов. – СПб.: Издательство "ВНУ СПб", 2004. – 320 с.
8. Методичні вказівки до виконання і захисту дипломного проекту (роботи) бакалавра спеціальностей 122 "Комп'ютерні науки та інформаційні технології", 123 "Комп'ютерна інженерія", 125 "Кібербезпека" (за напрямами 6.050101 "Комп'ютерні науки", 6.050102 "Комп'ютерна інженерія", 6.170101 "Безпека інформаційних і комунікаційних систем") для здобувачів вищої освіти денної і заочної форм навчання / Уклад.: Скарга-Бандурова І.С., Барбарук В.М., Кардашук В.С. – Сєвєродонецьк: СНУ ім. В. Даля, 2018. – 60 с.
9. Методичні вказівки до виконання розділу дипломного проекту (роботи)

бакалавра "Охорона праці та безпека в надзвичайних ситуаціях" (для студентів денної та заочної форм навчання за спеціальностями 122 "Комп'ютерні науки та інформаційні технології", 123 "Комп'ютерна інженерія", 125 "Кібербезпека" (за напрямками 6.050101 "Комп'ютерні науки", 6.050102 "Комп'ютерна інженерія", 6.170101 "Безпека інформаційних і комунікаційних систем")) / Уклад.: Критська Я.О. – Під ред. Скарги-Бандурової І.С. – Сєверодонецьк: вид-во СНУ ім. В. Даля, 2017. – 71 с.

10. ГОСТ 12.1.044-89 ССБТ. Пожежовибухонебезпека речовин і матеріалів.

Номенклатура показників і методи їх визначення

11. ДБН В.2.5-28:2015 Природне і штучне освітлення

12. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин

13. ДСН 3.3.6.037-99 Санітарні норми виробничого шуму, ультразвуку та інфразвуку

14. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих

15. НПАОП 0.00-4.12-05 Типове положення про порядок проведення навчання і перевірки знань з питань охорони праці

16. НПАОП 0.00-4.15-98 Про розробку інструкцій з охорони праці

17. НПАОП 0.00-6.03-93 Порядок опрацювання та затвердження власником нормативних актів про охорону праці

18. НАПБ Б.03.002-2007 Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою

19. НПАОП 40.1-1.01-97 Правила безопасной эксплуатации электроустановок

20. НПАОП 40.1-1.32-01 Правила устройства электроустановок. Электрооборудование специальных установок

21. ДСН 3.3.6.039-99 Санітарні норми виробничої загальної та локальної вібрації

22. ДСТУ ГОСТ 12.1.012-90 ССБТ. Вібраційна безпека. Загальні вимоги



23. ДБН В.2.5-67:2013 Опалення, вентиляція та кондиціонування
24. ГОСТ 12.1.006-84 ССБТ. Електромагнітні поля радіочастот. Загальні вимоги безпеки. Допустимі рівні на робочих місцях і вимоги до проведення контролю
25. ГОСТ 12.1.030-81 ССБТ. Електробезпечність. Захисне заземлення.
26. НПАОП 0.00-1.28-10 Правила охорони праці під час експлуатації електронно- обчислювальних машин

## ДОДАТОК А

Порядок та оформлення документів для призначення (перерахунку) пенсій

Додаток 3

до пункту 35 Порядку та оформлення документів  
для призначення (перерахунку) пенсій відповідно  
до Закону України “Про загальнообов’язкове  
державне пенсійне страхування”До органу, що призначає пенсію, \_\_\_\_\_<P1>, м. \_\_\_\_\_<P2>  
\_\_\_\_\_<P3> обл.

## ПОДАННЯ ПРО ПРИЗНАЧЕННЯ ПЕНСІЇ

Від \_\_\_\_\_&lt;P4&gt; року № \_\_\_\_\_&lt;P5&gt;

Адміністрація (правління) спільно із профспілковим комітетом

\_\_\_\_\_&lt;P6&gt;

(найменування підприємства, установи, організації)

адреса \_\_\_\_\_&lt;P7&gt; м. \_\_\_\_\_&lt;P8&gt;, вул. \_\_\_\_\_&lt;P9&gt;, \_\_\_\_\_&lt;P10&gt;

телефони: керівника \_\_\_\_\_&lt;P11&gt;, бухгалтерії \_\_\_\_\_&lt;P12&gt;, в/кадрів \_\_\_\_\_&lt;P13&gt;

представляє до призначення пенсії за віком, по інвалідності, за вислугу років, у зв’язку з  
втратою годувальника (непотрібне закреслити)

\_\_\_\_\_&lt;P14&gt; \_\_\_\_\_&lt;P15&gt; \_\_\_\_\_&lt;P16&gt;

(прізвище, ім’я, по батькові)

\_\_\_\_\_&lt;P17&gt; року народження, який проживає за адресою: \_\_\_\_\_&lt;P18&gt;

Дата	установлення інвалідності	_____<P53>
	смерть годувальника	_____<P53>
	припинення роботи, що дає право на пенсію за віслугу років	_____<P53>

**1. Непрацездатні члени сім’ї, які перебувають (перебували) на утриманні заявника  
(померлого годувальника)**

№ з/п	Прізвище, ім’я, по батькові	Родинні відносини з годувальником	Дата народження
_____<P19>	_____<P20>	_____<P21>	_____<P22>



### 3. Трудовий стаж за іншими документами заявника (померлого годувальника)

№ запису	Документ	Тривалість стажу роботи
<P24>	<P38>	<P39>.<P40>.<P41>
		<P33a>
Разом за іншими документами		<P39a>.<P40a>.<P41a>

### 4. Зведена таблиця обчислення стажу роботи

Види стажу	За трудовою книжкою	За іншими документами	Разом
Загальний	<P55>.<P55a>.<P55b>	<P39a>.<P40a>.<P41a>	<P54>.<P54a>.<P54b>
На пільгових умовах	Сп.<P52> <P47>.<P47a>.<P47b>		
За вислугу років			
Інших			

Стаж, що дає право на пенсію на пільгових умовах, за вислугу років (непотрібне закреслити) відповідно до статті Закону \_\_\_\_\_ <P42> \_\_\_\_\_, списку № <P43> \_\_\_\_\_, розділу \_\_\_\_\_ <P44> \_\_\_\_\_, підрозділу \_\_\_\_\_ <P45> \_\_\_\_\_, пункт (підпункт) \_\_\_\_\_ <P46> \_\_\_\_\_  
 Підраховано за записами № \_\_\_\_\_ <P48> \_\_\_\_\_

М.П.

Керівник \_\_\_\_\_ <P49> \_\_\_\_\_

(прізвище)

Голова профспілкового комітету \_\_\_\_\_ <P50> \_\_\_\_\_

(прізвище)

Із поданням ознайомлений \_\_\_\_\_  
 (підпис заявника)

\_\_\_\_\_ <P51> \_\_\_\_\_  
 (дата)

## ДОДАТОК Б

## Элементи даних

<b>Код</b>	<b>Название</b>	<b>Тип</b>	<b>Длина</b>	<b>Целое</b>	<b>Дробное</b>	<b>Описание</b>
ЭД5119	Название органа, назначающего пенсию	Строка	50	0	0	Юридическое название органа, назначающий пенсию.
ЭД5120	Город органа, назначающего пенсию	Строка	50	0	0	Название города, в котором расположен орган, назначающий пенсию.
ЭД5121	Область органа, назначающего пенсию	Строка	50	0	0	Название области, в которой расположен орган, назначающий пенсию.
ЭД5127	Название предприятия	Строка	100	0	0	Название предприятия, готовящего документ.
ЭД5129	Индекс	Целое	5	0	0	Индекс предприятия, готовящего документ
ЭД5130	Населенный пункт	Строка	50	0	0	Город предприятия, готовящего документ
ЭД5131	Улица	Строка	50	0	0	Улица предприятия, готовящего документ
ЭД5132	Номер дома	Строка	4	0	0	Дом предприятия, готовящего документ
ЭД5134	Телефон руководителя	Строка	17	0	0	Телефон руководителя предприятия, готовящего документ
ЭД5135	Телефон бухгалтерии	Строка	17	0	0	Телефон бухгалтерии предприятия, готовящего документ
ЭД5136	Телефон отдела кадров	Строка	17	0	0	Телефон отдела кадров предприятия, готовящего документ
ЭД5139	Фамилия заявителя	Строка	50	0	0	Фамилия заявителя
ЭД5140	Имя заявителя	Строка	50	0	0	Имя заявителя

<b>Код</b>	<b>Название</b>	<b>Тип</b>	<b>Длина</b>	<b>Целое</b>	<b>Дробное</b>	<b>Описание</b>
ЭД5141	Отчество заявителя	Строка	50	0	0	Отчество заявителя
ЭД5142	Дата рождения заявителя	Дата	–	–	–	Дата рождения заявителя
ЭД5143	Адрес заявителя	Строка	100	0	0	Физический адрес заявителя
ЭД5145	Ф.И.О. члена семьи	Строка	100	0	0	Фамилия, Имя, Отчество члена семьи который находится (находился) на содержании заявителя (погибшего кормильца).
ЭД5146	Семейные отношения с кормильцем	Строка	150	0	0	Семейные отношения с кормильцем
ЭД5147	Дата рождения	Дата	–	–	–	Дата рождения нетрудоспособного члена семьи, который находится (находился) на содержании заявителя (умершего кормильца)
ЭД5149	Дата заполнения трудовой книжки	Дата	–	–	–	Дата начала заполнения трудовой книжки
ЭД5151	Год приема	Целое	4	0	0	Год приема заявителя на новое место работы.
ЭД5152	Месяц приема	Целое	2	0	0	Месяц приема заявителя на новое место работы.
ЭД5153	Число приема	Целое	2	0	0	Число приема заявителя на новое место работы.
ЭД5154	Сведения по строке записи	Строка	400	0	0	Сведения про принятие, перемещение, увольнение ... по работе
ЭД5155	Основание строки записи	Строка	100	0	0	На основании каких документов сделана запись по приему, увольнения, перемещению...
ЭД5157	Лет стажа	Целое	2	0	0	Количество лет стажа

<b>Код</b>	<b>Название</b>	<b>Тип</b>	<b>Длина</b>	<b>Целое</b>	<b>Дробное</b>	<b>Описание</b>
ЭД5158	Месяцев стажа	Целое	2	0	0	Количество месяцев стажа
ЭД5159	Дней стажа	Целое	2	0	0	Количество дней стажа
ЭД5161	Лет льготного стажа	Целое	2	0	0	Количество лет льготного стажа
ЭД5162	Месяцев льготного стажа	Целое	2	0	0	Количество месяцев льготного стажа
ЭД5163	Дней льготного стажа	Целое	2	0	0	Количество дней льготного стажа
ЭД5165	Документ, подтверждающий стаж	Строка	100	0	0	Название документа подтверждающего стаж
ЭД5167	Лет стажа по другим документам	Целое	2	0	0	Количество лет стажа по другим документам
ЭД5168	Месяцев стажа по другим документам	Целое	2	0	0	Количество месяцев стажа по другим документам
ЭД5169	Дней стажа по другим документам	Целое	2	0	0	Количество дней стажа по другим документам
ЭД5181	Дата подписания	Дата	–	–	–	Дата подписания представления
ЭД5184	Статья закона	Строка	50	0	0	Статья закона согласно которой стаж дает право на пенсию на льготных условиях, за выслугу лет.
ЭД5185	Номер списка	Целое	1	0	0	Номер льготного списка. Принимает значение 1 или 2.
ЭД5186	Раздел списка	Строка	4	0	0	Раздел льготного списка
ЭД5188	Подраздел списка	Целое	1	0	0	подраздел льготного списка

Код	Название	Тип	Длина	Целое	Дробное	Описание
ЭД5190	Руководитель	Строка	50	0	0	Фамилия, инициалы руководителя предприятия готовящего документ.
ЭД5191	Председатель профкома	Строка	50	0	0	Фамилия, инициалы председателя профсоюзного комитета предприятия готовящего документ.
ЭД5192	Подсчитано по записям	Строка	50	0	0	Указываются записи, по которым считался стаж работы.
ЭД5195	Год увольнения	Целое	4	0	0	Год увольнения заявителя с места работы.
ЭД5196	Месяц увольнения	Целое	2	0	0	Месяц увольнения заявителя с места работы.
ЭД5197	Число увольнения	Целое	2	0	0	Число увольнения заявителя с места работы.
ЭД5212	Корректировочные дни	Целое	1	0	0	Количество дней, на которое необходимо скорректировать стаж работы.
ЭД5215	Разница между увольнением и приемом	Целое	4	0	0	Количество дней между увольнением с предыдущего места работы и принятием на следующее место работы
ЭД5236	Дата формирования представления	Дата	–	–	–	Календарная дата формирования документа программой.
ЭД5237	Номер представления	Строка	8	0	0	Номер документа по порядку.
ЭД5241	Примечание	Строка	50	0	0	Примечание по документу
ЭД5303	Пункт (подпункт) списка	Строка	30	0	0	Пункт (подпункт) списка льготного стажа
ЭД5304	Номер приема	Целое	3	0	0	Номер приема по трудовой
ЭД5305	Номер увольнения	Целое	3	0	0	Номер увольнения по трудовой
ЭД5306	Лет общей длительности стажа работы по другим документам	Целое	2	0	0	Количество лет общей длительности стажа работы по другим документам
	Месяцев общей	Целое	2	0	0	Количество месяцев общей



Код	Название	Тип	Длина	Целое	Дробное	Описание
ЭД5307	длительности стажа работы по другим документам					длительности стажа работы по другим документам
ЭД5308	Дней общей длительности стажа работы по другим документам	Целое	2	0	0	Количество дней общей длительности стажа работы по другим документам
ЭД5309	Лет общего стажа работы по трудовой книжке	Целое	2	0	0	Количество лет общего стажа работы по трудовой книжке
ЭД5310	Месяцев общего стажа работы по трудовой книжке	Целое	2	0	0	Количество месяцев общего стажа работы по трудовой книжке
ЭД5311	Дней общего стажа работы по трудовой книжке	Целое	2	0	0	Количество дней общего стажа работы по трудовой книжке
ЭД5312	Лет общего льготного стажа	Целое	2	0	0	Количество лет общего льготного стажа
ЭД5313	Месяцев общего льготного стажа	Целое	2	0	0	Количество месяцев общего льготного стажа
ЭД5314	Дней общего льготного стажа	Целое	2	0	0	Количество дней общего льготного стажа
ЭД5315	Лет общего стажа	Целое	2	0	0	Количество лет общего стажа
ЭД5316	Месяцев общего стажа	Целое	2	0	0	Количество месяцев общего стажа
ЭД5317	Дней общего стажа	Целое	2	0	0	Количество дней общего стажа
ЭД5318	Состояние заявителя	Целое	1	0	0	1- Инвалидность; 2- Смерть кормильца; 3- Прекращение работы, что дает право на пенсию по выслуге лет
ЭД5319	Дата начала состояния заявителя	Дата	-	-	-	Дата начала одного из состояний заявителя
ЭД5320	Выбранный номер списка	Целое	1	0	0	Номер списка, который дает право на пенсию на льготных условиях
ЭД5321	Номер записи по порядку	Целое	3	0	0	Номер записи по другому документу

## ДОДАТОК В

## Лістинг програми

```

1.   unit uPNP_LoginForm;
2.   interface
3.   uses
4.   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
5.   StdCtrls, Buttons, Mask, uPNP_ConnectionForm;
6.   type
7.   TPNP_LoginForm = class(TForm)
8.   eLogin: TEdit;
9.   ePassword: TMaskEdit;
10.  BitBtn1: TBitBtn;
11.  BitBtn2: TBitBtn;
12.  Label1: TLabel;
13.  Label2: TLabel;
14.  procedure FormShow(Sender: TObject);
15.  procedure BitBtn1Click(Sender: TObject);
16.  procedure BitBtn2Click(Sender: TObject);
17.  private
18.  { Private declarations }
19.  public
20.  { Public declarations }
21.  end;
22.  var
23.  PNP_LoginForm: TPNP_LoginForm;
24.  implementation
25.  {$R *.DFM}
26.  const
27.  FileName = 'PNP.ini';
28.  procedure TPNP_LoginForm.FormShow(Sender: TObject);
29.  var
30.  S: String;
31.  T: TextFile;
32.  begin
33.  AssignFile( T, FileName );
34.  {$I-}
35.  Reset( T );
36.  {$I+}
37.  if IOResult <> 0 then
38.  begin
39.  {$I-}
40.  Rewrite( T );
41.  {$I+}
42.  if IOResult <> 0 then
43.  begin
44.  MessageDlg( 'Ошибка создания ini-файла', mError, [mbOk], 0 );
45.  Exit;
46.  end
47.  end
48.  else
49.  begin
50.  ReadLn( T, S );
51.  eLogin.Text := S;
52.  ReadLn( T, S );
53.  ePassword.Text := S;
54.  end;
55.  CloseFile( T );
56.  end;
57.  procedure TPNP_LoginForm.BitBtn1Click(Sender: TObject);
58.  var
59.  T: TextFile;
60.  ConnectionString: String;
61.  PathToDB: String;
62.  begin
63.  PathToDB := ExtractFilePath(ParamStr(0));
64.  ConnectionString := 'Provider=LCPLIBProvider.3;Free;Password=' + ePassword.Text +
65.  ';Persist Security Info=True;User ID=' + eLogin.Text +
66.  ';Location=localhost.' + PathToDB + '\D1.FDB;ctype=WIN1251;garbage_collect=True;' +
67.  'auto_commit=True;auto_commit_level=65536;bicursor=False;free_threading=True;' +
68.  'array_vt_type=True;inner_trans=True;truncate_char=True;schema_cache=1;' +
69.  'support_odbc_query=False;force_prepare_stmt=False;unicode_mode=True;' +

```

```

70. 'unicode_stmt=True;clear_param_info=True;auto_param_describe=True;' +
71. 'force_param_describe=True;named_param_prefix=::ignore_unprepare=True;' +
72. 'exec_sp_named_param=False;odbc_call_sp=2;auto_commit_ddl=0;auto_commit_ddl_level=4096;' +
73. 'rowset_vm_using=512;nested_trans=False;ado_exec_sp=2;schema_sp_params=2;' +
74. 'ignore_err_param=0;dbclient_type=ib;support_square_delim_name=0;keywords_list=0;' +
75. 'tr_abort_flags=0;array_type=1;reset_ds_tr_rule=1;numeric_i2_rules=0;' +
76. 'numeric_i4_rules=0;numeric_i8_rules=0;dbclient_lock_rules=1';
77. PNP_ConnectionForm.ADOConnection1.ConnectionString := ConnectionString;
78. try
79. PNP_ConnectionForm.ADOConnection1.Connected := True;
80. except on E : Exception do
81. begin
82. ShowMessage( E.ClassName + ' db error: ' + E.Message );
83. Halt;
84. end;
85. end;
86. AssignFile( T, FileName );
87. {$I-}
88. ReWrite( T );
89. {$I+}
90. if IOResult <> 0 then
91. begin
92. MessageDlg( 'Ошибка при записи ini-файла', mtError, [mbOk], 0 );
93. Exit;
94. end;
95. WriteLn( T, eLogin.Text );
96. WriteLn( T, ePassword.Text );
97. CloseFile( T );
98. end;
99. procedure TPNP_LoginForm.BitBtn2Click(Sender: TObject);
100. begin
101. Halt;
102. end;
103. end.
104. unit uPNP_ManagePredstavEditForm;
105. interface
106. uses
107. Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
108. StdCtrls, Spin, Buttons, ExtCtrls, Mask, ToolEdit, Grids, Db, ADODB,
109. uPNP_ConnectionForm, OPCDataAccessTypeLibrary, comobj;
110. type
111. TManagePredstavEditFormData = record
112. PREDSTAV_ID: Integer;
113. ZAJAVITEL_ID: Integer;
114. nomer_predst: String;
115. date_predst: String;
116. familija: String;
117. imja: String;
118. otchestvo: String;
119. data_rojd: String;
120. end;
121.
122. TManagePredstavEditFormArray = array of TManagePredstavEditFormData;
123. TPNP_ManagePredstavEditForm = class(TForm)
124. GroupBox1: TGroupBox;
125. Panel2: TPanel;
126. Panel1: TPanel;
127. SpeedButton1: TSpeedButton;
128. SpeedButton2: TSpeedButton;
129. SpeedButton3: TSpeedButton;
130. SpeedButton4: TSpeedButton;
131. GroupBox2: TGroupBox;
132. StringGrid1: TStringGrid;
133. Panel3: TPanel;
134. Label2: TLabel;
135. Label3: TLabel;
136. Label4: TLabel;
137. Label5: TLabel;
138. Label6: TLabel;
139. eFirstName: TEdit;
140. eSecondName: TEdit;
141. seNumberFrom: TSpinEdit;
142. seNumberTo: TSpinEdit;
143. deDateFrom: TDateEdit;
144. deDateTo: TDateEdit;
145. Label7: TLabel;
146. Label8: TLabel;
147. deBornDate: TDateEdit;

```

```

148.   bSearch: TBitBtn;
149.   Label1: TLabel;
150.   eLastName: TEdit;
151.   bPrint: TBitBtn;
152.   ADOQuery1: TADOQuery;
153.   procedure SpeedButton4Click(Sender: TObject);
154.   procedure SpeedButton3Click(Sender: TObject);
155.   procedure SpeedButton2Click(Sender: TObject);
156.   procedure SpeedButton1Click(Sender: TObject);
157.   procedure bSearchClick(Sender: TObject);
158.   procedure bPrintClick(Sender: TObject);
159.   private
160.   { Private declarations }
161.   public
162.   { Public declarations }
163.   dataArray: TManagePredstavEditFormArray;
164.   end;
165.   var
166.   PNP_ManagePredstavEditForm: TPNP_ManagePredstavEditForm;
167.   MyWnd : HWND;
168.   implementation
169.   uses uPNP_AddPredstavEditForm;
170.   {$R *.DFM}
171.   procedure TPNP_ManagePredstavEditForm.SpeedButton4Click(Sender: TObject);
172.   var
173.   Msg, SQL_str: String;
174.   begin
175.   if ( StringGrid1.Row > 0 ) and ( StringGrid1.Row < StringGrid1.RowCount ) then
176.   begin
177.   Msg := 'Выбранная запись будет удалена.' + #10#13 + 'Удалить запись?';
178.   if ( mrOK = MessageDlg( Msg, mtWarning, mbOKCancel, 0 ) ) then
179.   begin
180.   SQL_str := 'delete from ZAJAVITEL where ZAJAVITEL_ID = "' +
181.   IntToStr( dataArray[StringGrid1.Row-1].ZAJAVITEL_ID ) + "'";
182.   ADOQuery1.SQL.Clear;
183.   ADOQuery1.SQL.Append( SQL_str );
184.   try
185.   ADOQuery1.ExecSQL;
186.   except on E : Exception do
187.   begin
188.   ShowMessage( E.ClassName + ' db error: ' + E.Message );
189.   ADOQuery1.Close;
190.   Exit;
191.   end;
192.   end;
193.   end;
194.   end;
195.   end;
196.   procedure TPNP_ManagePredstavEditForm.SpeedButton3Click(Sender: TObject);
197.   begin
198.   if ( High( dataArray ) > -1 ) then
199.   begin
200.   PNP_AddPredstavEditForm.Button1.Enabled := True;
201.   PNP_AddPredstavEditForm.Button2.Enabled := True;
202.   PNP_AddPredstavEditForm.Button3.Enabled := True;
203.   PNP_AddPredstavEditForm.Caption := 'Изменение представления';
204.   PNP_AddPredstavEditForm.SetTableName( 'PREDSTAV' );
205.   PNP_AddPredstavEditForm.SetID_Field( 'PREDSTAV_ID' );
206.   PNP_AddPredstavEditForm.SetID( dataArray[StringGrid1.Row-1].PREDSTAV_ID );
207.   PNP_AddPredstavEditForm.ZAJAVITEL_ID.Text := IntToStr( dataArray[StringGrid1.Row-1].ZAJAVITEL_ID );
208.   PNP_AddPredstavEditForm.SetNewRecord( false );
209.   PNP_AddPredstavEditForm.ShowModal;
210.   end;
211.   end;
212.   procedure TPNP_ManagePredstavEditForm.SpeedButton2Click(Sender: TObject);
213.   var
214.   MAX: Integer;
215.   begin
216.   PNP_AddPredstavEditForm.Caption := 'Новое представление';
217.   MAX := PNP_AddPredstavEditForm.MaxID( 'PREDSTAV', 'NOMER_PREDST' );
218.   if ( -1 = MAX ) then Exit;
219.   Inc( MAX );
220.   PNP_AddPredstavEditForm.Button1.Enabled := False;
221.   PNP_AddPredstavEditForm.Button2.Enabled := False;
222.   PNP_AddPredstavEditForm.Button3.Enabled := False;
223.   PNP_AddPredstavEditForm.SetNewRecord( True );
224.   PNP_AddPredstavEditForm.sePredstavNumber.Value := MAX;
225.   PNP_AddPredstavEditForm.dePredstavDate.Text := DateToStr( Date );

```

```

226. PNP_AddPredstavEditForm.SetTableName( 'PREDSTAV' );
227. PNP_AddPredstavEditForm.SetID_Field( 'PREDSTAV_ID' );
228. PNP_AddPredstavEditForm.ShowModal;
229. end;
230. procedure TPNP_ManagePredstavEditForm.SpeedButton1Click(Sender: TObject);
231. begin
232. Close;
233. end;
234. procedure TPNP_ManagePredstavEditForm.bSearchClick(Sender: TObject);
235. var
236. s, SQL_str: String;
237. i: Integer;
238. begin
239. s := deDateFrom.Text;
240. try
241. StrToDate( s );
242. except
243. MessageDlg( 'Введите правильно дату представления c', mtError, [mbOK], 0 );
244. Exit;
245. end;
246. StringGrid1.RowCount := 2;
247. StringGrid1.Rows[0].Clear;
248. StringGrid1.Rows[0].Append( 'Номер представления' );
249. StringGrid1.Rows[0].Append( 'Дата представления' );
250. StringGrid1.Rows[0].Append( 'Фамилия' );
251. StringGrid1.Rows[0].Append( 'Имя' );
252. StringGrid1.Rows[0].Append( 'Отчество' );
253. StringGrid1.Rows[0].Append( 'Дата рождения' );
254. StringGrid1.Rows[1].Clear;
255. SQL_str := 'select p.PREDSTAV_ID, p.nomer_predst, p.date_predst, p.ZAJAVITEL_ID, z.familija, z.imja, z.otchestvo,
z.data_rojld ' +
256. 'from zajavitel z join predstav p on z.zajavitel_id = p.zajavitel_id ' +
257. 'where (p.nomer_predst between "' + seNumberFrom.Text + '" and ';
258. if seNumberTo.Value < seNumberFrom.Value then
259. SQL_str := SQL_str + "' + seNumberFrom.Text + '"
260. else
261. SQL_str := SQL_str + "' + seNumberTo.Text + '"';
262. SQL_str := SQL_str + ') and (p.date_predst between "' + deDateFrom.Text + '" and ';
263. s := deDateTo.Text;
264. try
265. StrToDate( s );
266. SQL_str := SQL_str + "' + s + '"';
267. except
268. SQL_str := SQL_str + "' + deDateFrom.Text + '"';
269. end;
270. SQL_str := SQL_str + ');
271. // Фамилия
272. if eLastName.Text <> '' then
273. SQL_str := SQL_str + 'and (z.familija = "' + eLastName.Text + '" )';
274. // Имя
275. if eFirstName.Text <> '' then
276. SQL_str := SQL_str + 'and (z.imja = "' + eFirstName.Text + '" )';
277. // Отчество
278. if eSecondName.Text <> '' then
279. SQL_str := SQL_str + 'and (z.otchestvo = "' + eSecondName.Text + '" )';
280. // Дата рождения
281. s := deBornDate.Text;
282. try
283. StrToDate( s );
284. SQL_str := SQL_str + 'and (z.data_rojld = "' + s + '" )';
285. except
286. end;
287. SQL_str := SQL_str + ';';
288. try
289. ADOQuery1.Close;
290. except
291. end;
292. ADOQuery1.SQL.Clear;
293. ADOQuery1.SQL.Append( SQL_str );
294. try
295. ADOQuery1.Open;
296. except on E : Exception do
297. begin
298. ShowMessage( E.ClassName + ' db error: ' + E.Message );
299. end;
300. end;
301. if ADOQuery1.RecordCount > 0 then
302. begin

```

```

303. SetLength( dataArray, ADOQuery1.RecordCount );
304. StringGrid1.RowCount := ADOQuery1.RecordCount + 1;
305. ADOQuery1.First;
306. for i := 1 to ADOQuery1.RecordCount do
307. begin
308. dataArray[i-1].PREDSTAV_ID := StrToInt( ADOQuery1.FieldValues['PREDSTAV_ID'] );
309. dataArray[i-1].ZAJAVITEL_ID := StrToInt( ADOQuery1.FieldValues['ZAJAVITEL_ID'] );
310. dataArray[i-1].nomer_predst := ADOQuery1.FieldValues['nomer_predst'];
311. dataArray[i-1].date_predst := ADOQuery1.FieldValues['date_predst'];
312. dataArray[i-1].familija := ADOQuery1.FieldValues['familija'];
313. dataArray[i-1].imja := ADOQuery1.FieldValues['imja'];
314. dataArray[i-1].otchestvo := ADOQuery1.FieldValues['otchestvo'];
315. dataArray[i-1].data_rojd := ADOQuery1.FieldValues['data_rojd'];
316. StringGrid1.Cells[ 0, i ] := dataArray[i-1].nomer_predst;
317. StringGrid1.Cells[ 1, i ] := dataArray[i-1].date_predst ;
318. StringGrid1.Cells[ 2, i ] := dataArray[i-1].familija ;
319. StringGrid1.Cells[ 3, i ] := dataArray[i-1].imja ;
320. StringGrid1.Cells[ 4, i ] := dataArray[i-1].otchestvo ;
321. StringGrid1.Cells[ 5, i ] := dataArray[i-1].data_rojd ;
322. if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
323. end;
324. end;
325. end;
326. procedure TPNP_ManagePredstavEditForm.bPrintClick(Sender: TObject);
327. var
328. v, S, S_dodat, S_shabl: String;
329. PENS_ORGAN_ID, ZAJAVITEL_ID, PREDPR_ID, TRUD_KNIGA_ID: String;
330. W: Variant;
331. i: integer;
332. SystemTime: TSystemTime;
333. SQL_str: String;
334. function EnumProc(WinHandle: HWND; Param: LongInt): Boolean; stdcall;
335. var
336. WndCaption: array [0..255] of Char;
337. begin
338. ZeroMemory(@WndCaption, 255);
339. GetWindowText(WinHandle, WndCaption, 255);
340. if Pos('Microsoft Word', WndCaption) <> 0 then
341. begin
342. EnumProc := False;
343. MyWnd := WinHandle;
344. end
345. else
346. EnumProc := True;
347. end; // EnumProc
348. begin
349. if ( StringGrid1.Row > 0 ) and ( StringGrid1.Row < StringGrid1.RowCount ) then
350. begin
351. GetLocalTime(SystemTime);
352. S := 'Dodatok3 ' + Format('%d-%02d-%d %dч%дм%дс',[SystemTime.wDay, SystemTime.wMonth,
353. SystemTime.wYear, SystemTime.wHour, SystemTime.wMinute, SystemTime.wSecond]) + '.doc';
354. S_dodat := ExtractFilePath(ParamStr(0)) + '\ ' + S; // имя открываемого документа
355. screen.Cursor := crAppStart;
356. try // Если Word уже запущен
357. W := GetActiveOleObject('Word.Application'); // Взять ссылку на запущенный OLE объект
358. except
359. try // Word не запущен, запустить
360. W := CreateOleObject('Word.Application'); // Создать ссылку на зарегистрированный OLE объект
361. except
362. MessageDlg(Format('Невозможно запустить Microsoft Word %s%s',
363. ['Возможно, что документ уже открыт в Microsoft Word. ',
364. 'Закройте документ и повторите попытку.']), mtError, [mbOK], 0);
365. screen.Cursor := crDefault;
366. Exit;
367. end;
368. end;
369. W.Visible := False;
370. S_shabl := ExtractFilePath(ParamStr(0)) + 'Shablon.doc';
371. CopyFile(PChar(S_shabl), PChar(S_dodat), True);
372. W.Documents.Open(S_dodat);
373. W.ChangeFileOpenDirectory( ExtractFilePath( ParamStr(0) ) );
374. try
375. SQL_str := 'select * from PREDSTAV where PREDSTAV_ID = ' +
376. IntToStr( dataArray[StringGrid1.Row-1].PREDSTAV_ID ) + ''';
377. ADOQuery1.Close;
378. ADOQuery1.SQL.Clear;
379. ADOQuery1.SQL.Append( SQL_str );
380. try

```

```

381.     ADOQuery1.Open;
382.     except on E : Exception do
383.     begin
384.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
385.     Exit;
386.     end;
387.     end;
388.     if ADOQuery1.RecordCount > 0 then
389.     begin
390.     W.WordBasic.Call('Module1.GOTO_P4');
391.     v := ADOQuery1.FieldValues[ 'DATE_PREDST' ];
392.     W.WordBasic.Insert( v );
393.     W.WordBasic.Call('Module1.GOTO_P5');
394.     v := ADOQuery1.FieldValues[ 'NOMER_PREDST' ];
395.     W.WordBasic.Insert( v );
396.     W.WordBasic.Call('Module1.GOTO_P42');
397.     v := ADOQuery1.FieldValues[ 'LGOT_STATJA_ZAKONA' ];
398.     W.WordBasic.Insert( v );
399.     W.WordBasic.Call('Module1.GOTO_P43');
400.     v := ADOQuery1.FieldValues[ 'LGOT_NOMER_SPISKA' ];
401.     W.WordBasic.Insert( v );
402.     W.WordBasic.Call('Module1.GOTO_P44');
403.     v := ADOQuery1.FieldValues[ 'LGOT_RAZDEL_SPISKA' ];
404.     W.WordBasic.Insert( v );
405.     W.WordBasic.Call('Module1.GOTO_P45');
406.     v := ADOQuery1.FieldValues[ 'LGOT_PODRAZDEL_SPISKA' ];
407.     W.WordBasic.Insert( v );
408.     W.WordBasic.Call('Module1.GOTO_P46');
409.     v := ADOQuery1.FieldValues[ 'LGOT_PUNKT_STAJA' ];
410.     W.WordBasic.Insert( v );
411.     W.WordBasic.Call('Module1.GOTO_P48');
412.     v := ADOQuery1.FieldValues[ 'LGOT_PODS_PO_ZAPIS' ];
413.     W.WordBasic.Insert( v );
414.     W.WordBasic.Call('Module1.GOTO_P51');
415.     v := ADOQuery1.FieldValues[ 'DATA_PODPISANIA' ];
416.     W.WordBasic.Insert( v );
417.     PENS_ORGAN_ID := ADOQuery1.FieldValues[ 'PENS_ORGAN_ID' ];
418.     ZAJAVITEL_ID := ADOQuery1.FieldValues[ 'ZAJAVITEL_ID' ];
419.     SQL_str := 'select * from PENS_ORGAN where PENS_ORGAN_ID = "' + PENS_ORGAN_ID + '"';
420.     ADOQuery1.Close;
421.     ADOQuery1.SQL.Clear;
422.     ADOQuery1.SQL.Append( SQL_str );
423.     try
424.     ADOQuery1.Open;
425.     except on E : Exception do
426.     begin
427.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
428.     Exit;
429.     end;
430.     end;
431.     if ADOQuery1.RecordCount > 0 then
432.     begin
433.     W.WordBasic.Call('Module1.GOTO_P1');
434.     v := ADOQuery1.FieldValues[ 'NAZVANIE' ];
435.     W.WordBasic.Insert( v );
436.     W.WordBasic.Call('Module1.GOTO_P2');
437.     v := ADOQuery1.FieldValues[ 'NAS_PUNKT' ];
438.     W.WordBasic.Insert( v );
439.     W.WordBasic.Call('Module1.GOTO_P3');
440.     v := ADOQuery1.FieldValues[ 'OBLAST' ];
441.     W.WordBasic.Insert( v );
442.     end;
443.     SQL_str := 'select * from ZAJAVITEL where ZAJAVITEL_ID = "' + ZAJAVITEL_ID + '"';
444.     ADOQuery1.Close;
445.     ADOQuery1.SQL.Clear;
446.     ADOQuery1.SQL.Append( SQL_str );
447.     try
448.     ADOQuery1.Open;
449.     except on E : Exception do
450.     begin
451.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
452.     Exit;
453.     end;
454.     end;
455.     if ADOQuery1.RecordCount > 0 then
456.     begin
457.     W.WordBasic.Call('Module1.GOTO_P14');
458.     v := ADOQuery1.FieldValues[ 'FAMILIJA' ];

```

```

459.     W.WordBasic.Insert( v );
460.     W.WordBasic.Call('Module1.GOTO_P15');
461.     v := ADOQuery1.FieldValues[ 'IMJA' ];
462.     W.WordBasic.Insert( v );
463.     W.WordBasic.Call('Module1.GOTO_P16');
464.     v := ADOQuery1.FieldValues[ 'OTCHESTVO' ];
465.     W.WordBasic.Insert( v );
466.     W.WordBasic.Call('Module1.GOTO_P17');
467.     v := ADOQuery1.FieldValues[ 'DATA_ROJD' ];
468.     W.WordBasic.Insert( v );
469.     W.WordBasic.Call('Module1.GOTO_P18');
470.     v := ADOQuery1.FieldValues[ 'ADRES' ];
471.     W.WordBasic.Insert( v );
472.     v := ADOQuery1.FieldValues[ 'SOST_ZAJAVITEL_ID' ];
473.     if '1' = v then W.WordBasic.Call('Module1.GOTO_P53_1')
474.     else if '2' = v then W.WordBasic.Call('Module1.GOTO_P53_2')
475.     else if '3' = v then W.WordBasic.Call('Module1.GOTO_P53_3');
476.     v := ADOQuery1.FieldValues[ 'DATE_SOSTOYANIE' ];
477.     W.WordBasic.Insert( v );
478.     PREDPR_ID := ADOQuery1.FieldValues[ 'PREDPR_ID' ];
479.     SQL_str := 'select * from PREDPR where PREDPR_ID = "' + PREDPR_ID + '"';
480.     ADOQuery1.Close;
481.     ADOQuery1.SQL.Clear;
482.     ADOQuery1.SQL.Append( SQL_str );
483.     try
484.     ADOQuery1.Open;
485.     except on E : Exception do
486.     begin
487.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
488.     Exit;
489.     end;
490.     end;
491.     if ADOQuery1.RecordCount > 0 then
492.     begin
493.     W.WordBasic.Call('Module1.GOTO_P6');
494.     v := ADOQuery1.FieldValues[ 'NAZVANIE' ];
495.     W.WordBasic.Insert( v );
496.     W.WordBasic.Call('Module1.GOTO_P7');
497.     v := ADOQuery1.FieldValues[ 'INDEKS' ];
498.     W.WordBasic.Insert( v );
499.     W.WordBasic.Call('Module1.GOTO_P8');
500.     v := ADOQuery1.FieldValues[ 'NAS_PUNKT' ];
501.     W.WordBasic.Insert( v );
502.     W.WordBasic.Call('Module1.GOTO_P9');
503.     v := ADOQuery1.FieldValues[ 'ULICA' ];
504.     W.WordBasic.Insert( v );
505.     W.WordBasic.Call('Module1.GOTO_P10');
506.     v := ADOQuery1.FieldValues[ 'DOM' ];
507.     W.WordBasic.Insert( v );
508.     W.WordBasic.Call('Module1.GOTO_P11');
509.     v := ADOQuery1.FieldValues[ 'TEL_RUK' ];
510.     W.WordBasic.Insert( v );
511.     W.WordBasic.Call('Module1.GOTO_P12');
512.     v := ADOQuery1.FieldValues[ 'TEL_BUH' ];
513.     W.WordBasic.Insert( v );
514.     W.WordBasic.Call('Module1.GOTO_P13');
515.     v := ADOQuery1.FieldValues[ 'TEL_O_K' ];
516.     W.WordBasic.Insert( v );
517.     W.WordBasic.Call('Module1.GOTO_P49');
518.     v := ADOQuery1.FieldValues[ 'RUKOVODITEL' ];
519.     W.WordBasic.Insert( v );
520.     W.WordBasic.Call('Module1.GOTO_P50');
521.     v := ADOQuery1.FieldValues[ 'PROFSOUZ' ];
522.     W.WordBasic.Insert( v );
523.     end;
524.     end;
525.     SQL_str := 'select * from NETRUD where ZAJAVITEL_ID = "' + ZAJAVITEL_ID + '"';
526.     ADOQuery1.Close;
527.     ADOQuery1.SQL.Clear;
528.     ADOQuery1.SQL.Append( SQL_str );
529.     try
530.     ADOQuery1.Open;
531.     except on E : Exception do
532.     begin
533.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
534.     Exit;
535.     end;
536.     end;

```



```

537.   if ADOQuery1.RecordCount > 0 then
538.   begin
539.   W.WordBasic.Call('Module1.GOTO_P19');
540.   ADOQuery1.First;
541.   for i := 1 to ADOQuery1.RecordCount do
542.   begin
543.   v := IntToStr( i );
544.   W.WordBasic.Insert( v );
545.   W.WordBasic.NextCell;
546.   v := ADOQuery1.FieldValues[ 'F_I_O' ];
547.   W.WordBasic.Insert( v );
548.   W.WordBasic.NextCell;
549.   v := ADOQuery1.FieldValues[ 'RODSTVEN_SVAZ' ];
550.   W.WordBasic.Insert( v );
551.   W.WordBasic.NextCell;
552.   v := ADOQuery1.FieldValues[ 'DATA_ROJD' ];
553.   W.WordBasic.Insert( v );
554.   if i < ADOQuery1.RecordCount then
555.   begin
556.   W.WordBasic.NextCell;
557.   ADOQuery1.Next;
558.   end;
559.   end;
560.   end;
561.   SQL_str := 'select * from TRUD_KNIGA where ZAJAVITEL_ID = ' + ZAJAVITEL_ID + ''';
562.   ADOQuery1.Close;
563.   ADOQuery1.SQL.Clear;
564.   ADOQuery1.SQL.Append( SQL_str );
565.   try
566.   ADOQuery1.Open;
567.   except on E : Exception do
568.   begin
569.   ShowMessage( E.ClassName + ' db error: ' + E.Message );
570.   Exit;
571.   end;
572.   end;
573.   if ADOQuery1.RecordCount > 0 then
574.   begin
575.   W.WordBasic.Call('Module1.GOTO_P23');
576.   v := ADOQuery1.FieldValues[ 'DATA_ZAPOLN' ];
577.   W.WordBasic.Insert( v );
578.   TRUD_KNIGA_ID := ADOQuery1.FieldValues[ 'TRUD_KNIGA_ID' ];
579.   SQL_str := 'select * from ZAPIS_TRUD where TRUD_KNIGA_ID = ' + TRUD_KNIGA_ID + ''';
580.   ADOQuery1.Close;
581.   ADOQuery1.SQL.Clear;
582.   ADOQuery1.SQL.Append( SQL_str );
583.   try
584.   ADOQuery1.Open;
585.   except on E : Exception do
586.   begin
587.   ShowMessage( E.ClassName + ' db error: ' + E.Message );
588.   Exit;
589.   end;
590.   end;
591.   if ADOQuery1.RecordCount > 0 then
592.   begin
593.   ADOQuery1.First;
594.   W.WordBasic.Call('Module1.GOTO_P24_1');
595.   for i := 1 to ADOQuery1.RecordCount do
596.   begin
597.   v := ADOQuery1.FieldValues[ 'N_PRIEMA' ];
598.   W.WordBasic.Insert( v );
599.   W.WordBasic.NextCell;
600.   v := ADOQuery1.FieldValues[ 'GOD_PRIEMA' ];
601.   W.WordBasic.Insert( v );
602.   W.WordBasic.NextCell;
603.   v := ADOQuery1.FieldValues[ 'MES_PRIEMA' ];
604.   W.WordBasic.Insert( v );
605.   W.WordBasic.NextCell;
606.   v := ADOQuery1.FieldValues[ 'CISLO_PRIEMA' ];
607.   W.WordBasic.Insert( v );
608.   W.WordBasic.NextCell;
609.   v := ADOQuery1.FieldValues[ 'SVED_PO_STROKE_PRIEM' ];
610.   W.WordBasic.Insert( v );
611.   W.WordBasic.NextCell;
612.   v := ADOQuery1.FieldValues[ 'OSNOVANIE_STROKI_PRIEM' ];
613.   W.WordBasic.Insert( v );
614.   W.WordBasic.NextCell;

```

```

615. W.WordBasic.NextCell;
616. W.WordBasic.NextCell;
617. W.WordBasic.NextCell;
618. W.WordBasic.NextCell;
619. W.WordBasic.NextCell;
620. v := ADOQuery1.FieldValues[ 'N_UVOLN' ];
621. W.WordBasic.Insert( v );
622. W.WordBasic.NextCell;
623. v := ADOQuery1.FieldValues[ 'GOD_UVOLN' ];
624. W.WordBasic.Insert( v );
625. W.WordBasic.NextCell;
626. v := ADOQuery1.FieldValues[ 'MES_UVOLN' ];
627. W.WordBasic.Insert( v );
628. W.WordBasic.NextCell;
629. v := ADOQuery1.FieldValues[ 'CISLO_UVOLN' ];
630. W.WordBasic.Insert( v );
631. W.WordBasic.NextCell;
632. v := ADOQuery1.FieldValues[ 'SVED_PO_STROKE_UVOL' ];
633. W.WordBasic.Insert( v );
634. W.WordBasic.NextCell;
635. v := ADOQuery1.FieldValues[ 'OSNOVANIE_STROKI_UVOL' ];
636. W.WordBasic.Insert( v );
637. W.WordBasic.NextCell;
638. W.WordBasic.NextCell;
639. W.WordBasic.NextCell;
640. W.WordBasic.NextCell;
641. W.WordBasic.NextCell;
642. if i < ADOQuery1.RecordCount then
643. begin
644. W.WordBasic.NextCell;
645. ADOQuery1.Next;
646. end;
647. end;
648. end;
649. end;
650. SQL_str := 'select * from OTHER_DOC where ZAJAVITEL_ID = ' + ZAJAVITEL_ID + ''';
651. ADOQuery1.Close;
652. ADOQuery1.SQL.Clear;
653. ADOQuery1.SQL.Append( SQL_str );
654. try
655. ADOQuery1.Open;
656. except on E : Exception do
657. begin
658. ShowMessage( E.ClassName + ' db error: ' + E.Message );
659. Exit;
660. end;
661. end;
662. if ADOQuery1.RecordCount > 0 then
663. begin
664. W.WordBasic.Call('Module1.GOTO_P24_2');
665. ADOQuery1.First;
666. for i := 1 to ADOQuery1.RecordCount do
667. begin
668. v := ADOQuery1.FieldValues[ 'N_DOC' ];
669. W.WordBasic.Insert( v );
670. W.WordBasic.NextCell;
671. v := ADOQuery1.FieldValues[ 'DOCUMENT' ];
672. W.WordBasic.Insert( v );
673. W.WordBasic.NextCell;
674. if i < ADOQuery1.RecordCount then
675. begin
676. W.WordBasic.NextCell;
677. ADOQuery1.Next;
678. end;
679. end;
680. end;
681. end;
682. W.WordBasic.Call('Module1.SAVE_DOCUMENT');
683. except
684. Screen.Cursor := crDefault;
685. MessageDlg('Ошибка при формировании документа', mtError, [mbOk], 0);
686. Exit;
687. end;
688. W.Visible := True;
689. EnumWindows(@EnumProc, 0);
690. if myWnd <> 0 then
691. begin
692. SetForegroundWindow(MyWnd);

```

```

693.   Windows.SetFocus(MyWnd)
694.   end;
695.   Screen.Cursor := crDefault;
696.   end;
697.   end;
698.   end.
699.
700.   unit uPNP_StajPoDrugDocEditForm;
701.   interface
702.   uses
703.   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
704.   StdCtrls, Buttons, Spin, ADODB, uPNP_ConnectionForm;
705.   type
706.   TPNP_StajPoDrugDocEditForm = class( TPNP_EditFormBase )
707.   Label1: TLabel;
708.   spOtherDocNumber: TSpinEdit;
709.   Label2: TLabel;
710.   eOtherDocDocument: TEdit;
711.   Label3: TLabel;
712.   seOtherDocYear: TSpinEdit;
713.   Label4: TLabel;
714.   spOtherDocMonth: TSpinEdit;
715.   Label5: TLabel;
716.   seOtherDocDay: TSpinEdit;
717.   spOtherDocCorrectDay: TSpinEdit;
718.   Label6: TLabel;
719.   BitBtn1: TBitBtn;
720.   BitBtn2: TBitBtn;
721.   ZAJAVITEL_ID: TEdit;
722.   procedure FillFieldDataArray; override;
723.   procedure FormCreate(Sender: TObject);
724.   procedure BitBtn1Click(Sender: TObject);
725.   private
726.   { Private declarations }
727.   public
728.   { Public declarations }
729.   end;
730.   var
731.   PNP_StajPoDrugDocEditForm: TPNP_StajPoDrugDocEditForm;
732.   implementation
733.   {$R *.DFM}
734.   procedure TPNP_StajPoDrugDocEditForm.FillFieldDataArray;
735.   var
736.   FieldData: TPNP_FieldData;
737.   begin
738.   FieldDataArrayLength := 7;
739.   SetLength( FieldDataArray, FieldDataArrayLength );
740.   FieldData := TPNP_FieldData.Create;
741.   FieldData.SetData( 'ZAJAVITEL_ID', 'ZAJAVITEL_ID', component_Edit );
742.   FieldDataArray[0] := FieldData;
743.   FieldData := TPNP_FieldData.Create;
744.   FieldData.SetData( 'N_DOC', 'spOtherDocNumber', component_Spin );
745.   FieldDataArray[1] := FieldData;
746.   FieldData := TPNP_FieldData.Create;
747.   FieldData.SetData( 'DOCUMENT', 'eOtherDocDocument', component_Edit );
748.   FieldDataArray[2] := FieldData;
749.   FieldData := TPNP_FieldData.Create;
750.   FieldData.SetData( 'LET', 'seOtherDocYear', component_Spin );
751.   FieldDataArray[3] := FieldData;
752.   FieldData := TPNP_FieldData.Create;
753.   FieldData.SetData( 'MES', 'spOtherDocMonth', component_Spin );
754.   FieldDataArray[4] := FieldData;
755.   FieldData := TPNP_FieldData.Create;
756.   FieldData.SetData( 'DNEJ', 'seOtherDocDay', component_Spin );
757.   FieldDataArray[5] := FieldData;
758.   FieldData := TPNP_FieldData.Create;
759.   FieldData.SetData( 'KOR_DAY', 'spOtherDocCorrectDay', component_Spin );
760.   FieldDataArray[6] := FieldData;
761.   end;
762.   procedure TPNP_StajPoDrugDocEditForm.FormCreate(Sender: TObject);
763.   begin
764.   ADOQuery1 := TADOQuery.Create( self );
765.   ADOQuery1.Connection := PNP_ConnectionForm.ADOConnection1;
766.   FillFieldDataArray;
767.   SetTableName( 'OTHER_DOC' );
768.   SetID_Field( 'OTHER_DOC_ID' );
769.   end;
770.   procedure TPNP_StajPoDrugDocEditForm.BitBtn1Click(Sender: TObject);

```

```

771. begin
772. if eOtherDocDocument.Text = " then eOtherDocDocument.Text := ' ';
773. ApplyData;
774. end;
775. end.
776. unit uPNP_StajPoDrugDocViewForm;
777. interface
778. uses
779. Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
780. Db, ADODB, Grids, DBGrids, Buttons, ExtCtrls, uPNP_ConnectionForm;
781. type
782. TPNP_StajPoDrugDocViewForm = class(TForm)
783. Panel1: TPanel;
784. SpeedButton1: TSpeedButton;
785. SpeedButton2: TSpeedButton;
786. SpeedButton3: TSpeedButton;
787. SpeedButton4: TSpeedButton;
788. DBGrid1: TDBGrid;
789. DataSource1: TDataSource;
790. ADODataSet1: TADODataSet;
791. ADOQuery1: TADOQuery;
792. procedure SpeedButton1Click(Sender: TObject);
793. procedure SpeedButton3Click(Sender: TObject);
794. procedure SpeedButton2Click(Sender: TObject);
795. procedure SpeedButton4Click(Sender: TObject);
796. procedure FormatTable;
797. procedure FormShow(Sender: TObject);
798. procedure CalcCommonStaj;
799. procedure CalcCommonStajPoDrugDoc;
800. procedure FormClose(Sender: TObject; var Action: TCloseAction);
801. private
802. { Private declarations }
803. SelectedRow: Integer;
804. ID: Integer;
805. public
806. { Public declarations }
807. ZAJAVITEL_ID: String;
808. TRUD_KNIGA_ID: String;
809. end;
810. var
811. PNP_StajPoDrugDocViewForm: TPNP_StajPoDrugDocViewForm;
812. implementation
813. uses uPNP_StajPoDrugDocEditForm;
814. {$R *.DFM}
815. type
816. TFake = class (TDBGrid);
817. procedure TPNP_StajPoDrugDocViewForm.SpeedButton1Click(Sender: TObject);
818. begin
819. Close;
820. end;
821. procedure TPNP_StajPoDrugDocViewForm.SpeedButton3Click(Sender: TObject);
822. begin
823. if TFake (DBGrid1).RowCount > 0 then
824. begin
825. SelectedRow := TFake (DBGrid1).Row;
826. ID := DBGrid1.Fields[0].AsInteger;
827. PNP_StajPoDrugDocEditForm.SetID( ID );
828. PNP_StajPoDrugDocEditForm.SetNewRecord( false );
829. PNP_StajPoDrugDocEditForm.Caption := 'Изменение информации по другому документу';
830. if mrOK = PNP_StajPoDrugDocEditForm.ShowModal then
831. begin
832. ADODataSet1.Active := false;
833. ADODataSet1.Active := true;
834. FormatTable;
835. TFake (DBGrid1).Row := SelectedRow;
836. end;
837. end;
838. end;
839. procedure TPNP_StajPoDrugDocViewForm.SpeedButton2Click(Sender: TObject);
840. var
841. SQL_str: String;
842. s: String;
843. begin
844. SQL_str := 'select case when max(N_PRIEMA) > max(N_UVOLN) then max(N_PRIEMA) ' +
845. 'else max(N_UVOLN) end as max_n from ZAPIS_TRUD where '+
846. 'ZAPIS_TRUD.TRUD_KNIGA_ID=''' + TRUD_KNIGA_ID + ''';';
847. ADOQuery1.Close;
848. ADOQuery1.SQL.Clear;

```

```

849.     ADOQuery1.SQL.Append( SQL_str );
850.     try
851.     ADOQuery1.Open;
852.     except on E : Exception do
853.     begin
854.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
855.     ADOQuery1.Close;
856.     Exit;
857.     end;
858.     end;
859.     PNP_StajPoDrugDocEditForm.Caption := 'Добавление другого документа заявителя';
860.     PNP_StajPoDrugDocEditForm.SetNewRecord( true );
861.     PNP_StajPoDrugDocEditForm.ZAJAVITEL_ID.Text := ZAJAVITEL_ID;
862.     s := ADOQuery1.FieldValues[ 'max_n' ];
863.     s := IntToStr( StrToInt(s) + 1 );
864.     PNP_StajPoDrugDocEditForm.spOtherDocNumber.Text := s;
865.     if mrOK = PNP_StajPoDrugDocEditForm.ShowModal then
866.     begin
867.     SelectedRow := TFake( DBGrid1 ).Row;
868.     ADODataSet1.Active := false;
869.     ADODataSet1.Active := true;
870.     FormatTable;
871.     TFake( DBGrid1 ).Row := SelectedRow;
872.     end;
873.     end;
874.     procedure TPNP_StajPoDrugDocViewForm.SpeedButton4Click(Sender: TObject);
875.     var
876.     Msg: String;
877.     begin
878.     if not DBGrid1.DataSource.DataSet.Eof then
879.     begin
880.     Msg := 'Выбранная запись будет удалена.' + #10#13 + 'Удалить запись?';
881.     if ( mrOK = MessageDlg( Msg, mtWarning, mbOKCancel, 0 ) ) then
882.     begin
883.     SelectedRow := TFake( DBGrid1 ).Row;
884.     DBGrid1.DataSource.DataSet.Delete;
885.     ADODataSet1.Active := false;
886.     ADODataSet1.Active := true;
887.     FormatTable;
888.     if TFake( DBGrid1 ).RowCount > 0 then
889.     begin
890.     if SelectedRow >= TFake( DBGrid1 ).RowCount then
891.     SelectedRow := TFake( DBGrid1 ).RowCount - 1;
892.     TFake( DBGrid1 ).Row := SelectedRow;
893.     end;
894.     end;
895.     end;
896.     end;
897.     procedure TPNP_StajPoDrugDocViewForm.FormShow(Sender: TObject);
898.     begin
899.     FormatTable;
900.     end;
901.     procedure TPNP_StajPoDrugDocViewForm.FormatTable;
902.     var
903.     i, w: Integer;
904.     List: TStringList;
905.     begin
906.     if DBGrid1.Columns.Count > 0 then
907.     begin
908.     List := TStringList.Create;
909.     List.Add( 'Порядковый номер записи' );
910.     List.Add( 'Документ' );
911.     List.Add( 'Лет стажа по другому документу' );
912.     List.Add( 'Месяцев стажа по другому документу' );
913.     List.Add( 'Дней стажа по другому документу' );
914.     List.Add( 'Корректировочный день' );
915.     DBGrid1.Columns.Items[0].Visible := false;
916.     w := DBGrid1.Width div (DBGrid1.Columns.Count-1);
917.     for i := 1 to DBGrid1.Columns.Count-1 do
918.     begin
919.     DBGrid1.Columns.Items[i].Width := w;
920.     if List.Count >= i then
921.     DBGrid1.Columns.Items[i].Title.Caption := List[i-1];
922.     end;
923.     List.Destroy;
924.     end;
925.     end;
926.     procedure TPNP_StajPoDrugDocViewForm.CalcCommonStaj; //28

```

```

927.     var
928.     SQL_str: String;
929.     LET, MES, DAY: Integer;
930.     v: variant;
931.     i: Integer;
932.     A1, A2: Integer;
933.     begin
934.     SQL_str := 'select t.sum_let, t.sum_mes, t.sum_day, o.sum_other_let, o.sum_other_mes, ' +
935.     'o.sum_other_day from predstav o join (trud_kniga t join zajavitel z ' +
936.     'on t.zajavitel_id = z.zajavitel_id) on o.zajavitel_id = z.zajavitel_id ' +
937.     'where z.zajavitel_id = ' + ZAJAVITEL_ID + ''';
938.     ADOQuery1.Close;
939.     ADOQuery1.SQL.Clear;
940.     ADOQuery1.SQL.Append( SQL_str );
941.     try
942.     ADOQuery1.Open;
943.     except on E : Exception do
944.     begin
945.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
946.     Exit;
947.     end;
948.     end;
949.     if ADOQuery1.RecordCount > 0 then
950.     begin
951.     ADOQuery1.First;
952.     LET := 0; MES := 0; DAY := 0;
953.     for i := 1 to ADOQuery1.RecordCount do
954.     begin
955.     v := ADOQuery1.FieldValues[ 'SUM_LET' ];
956.     if v <> NULL then LET := LET + v;
957.     v := ADOQuery1.FieldValues[ 'SUM_OTHER_LET' ];
958.     if v <> NULL then LET := LET + v;
959.     v := ADOQuery1.FieldValues[ 'SUM_MES' ];
960.     if v <> NULL then MES := MES + v;
961.     v := ADOQuery1.FieldValues[ 'SUM_OTHER_MES' ];
962.     if v <> NULL then MES := MES + v;
963.     v := ADOQuery1.FieldValues[ 'SUM_DAY' ];
964.     if v <> NULL then DAY := DAY + v;
965.     v := ADOQuery1.FieldValues[ 'SUM_OTHER_DAY' ];
966.     if v <> NULL then DAY := DAY + v;
967.     if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
968.     end;
969.     A1 := Trunc( DAY/30 );
970.     A2 := Trunc( (MES + A1)/12 );
971.     DAY := DAY - A1 * 30;
972.     MES := ( MES + A1 ) - A2 * 12;
973.     LET := LET + A2;
974.     SQL_str := 'select * from PREDSTAV where ZAJAVITEL_ID = ' + ZAJAVITEL_ID + ''';
975.     ADOQuery1.Close;
976.     ADOQuery1.SQL.Clear;
977.     ADOQuery1.SQL.Append( SQL_str );
978.     try
979.     ADOQuery1.Open;
980.     except on E : Exception do
981.     begin
982.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
983.     Exit;
984.     end;
985.     end;
986.     if ADOQuery1.RecordCount > 0 then
987.     begin
988.     SQL_str := 'update PREDSTAV set ' +
989.     'SUM_LET = ' + IntToStr( LET ) + ', ' +
990.     'SUM_MES = ' + IntToStr( MES ) + ', ' +
991.     'SUM_DAY = ' + IntToStr( DAY ) + ' ' +
992.     ' where ZAJAVITEL_ID = ' + ZAJAVITEL_ID + ''';
993.     ADOQuery1.Close;
994.     ADOQuery1.SQL.Clear;
995.     ADOQuery1.SQL.Append( SQL_str );
996.     try
997.     ADOQuery1.ExecSQL;
998.     except on E : Exception do
999.     begin
1000.     ShowMessage( E.ClassName + ' db error: ' + E.Message );
1001.     Exit;
1002.     end;
1003.     end;
1004.     end

```

```

1005.   end;
1006.   end;
1007.   procedure TPNP_StajPoDrugDocViewForm.CalcCommonStajPoDrugDoc; //27
1008.   var
1009.     SQL_str: String;
1010.     LET, MES, DAY: Integer;
1011.     v: variant;
1012.     i: Integer;
1013.     A1, A2: Integer;
1014.   begin
1015.     SQL_str := 'select * from OTHER_DOC where ZAJAVITEL_ID = ' + ZAJAVITEL_ID + ''';
1016.     ADOQuery1.Close;
1017.     ADOQuery1.SQL.Clear;
1018.     ADOQuery1.SQL.Append( SQL_str );
1019.     try
1020.       ADOQuery1.Open;
1021.     except on E : Exception do
1022.       begin
1023.         ShowMessage( E.ClassName + ' db error: ' + E.Message );
1024.         Exit;
1025.       end;
1026.     end;
1027.     if ADOQuery1.RecordCount > 0 then
1028.     begin
1029.       ADOQuery1.First;
1030.       LET := 0; MES := 0; DAY := 0;
1031.       for i := 1 to ADOQuery1.RecordCount do
1032.         begin
1033.           v := ADOQuery1.FieldValues[ 'LET' ];
1034.           if v <> NULL then LET := LET + v;
1035.           v := ADOQuery1.FieldValues[ 'MES' ];
1036.           if v <> NULL then MES := MES + v;
1037.           v := ADOQuery1.FieldValues[ 'DNEJ' ];
1038.           if v <> NULL then DAY := DAY + v;
1039.           v := ADOQuery1.FieldValues[ 'KOR_DAY' ];
1040.           if v <> NULL then DAY := DAY + v;
1041.           if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1042.         end;
1043.         A1 := Trunc( DAY/30 );
1044.         A2 := Trunc( (MES + A1)/12 );
1045.         DAY := DAY - A1 * 30;
1046.         MES := (MES + A1) - A2 * 12;
1047.         LET := LET + A2;
1048.         SQL_str := 'select * from PREDSTAV where ZAJAVITEL_ID = ' + ZAJAVITEL_ID + ''';
1049.         ADOQuery1.Close;
1050.         ADOQuery1.SQL.Clear;
1051.         ADOQuery1.SQL.Append( SQL_str );
1052.         try
1053.           ADOQuery1.Open;
1054.         except on E : Exception do
1055.           begin
1056.             ShowMessage( E.ClassName + ' db error: ' + E.Message );
1057.             Exit;
1058.           end;
1059.         end;
1060.         if ADOQuery1.RecordCount > 0 then
1061.         begin
1062.           SQL_str := 'update PREDSTAV set ' +
1063.             'SUM_OTHER_LET = ' + IntToStr( LET ) + ', ' +
1064.             'SUM_OTHER_MES = ' + IntToStr( MES ) + ', ' +
1065.             'SUM_OTHER_DAY = ' + IntToStr( DAY ) + ' ' +
1066.             ' where ZAJAVITEL_ID = ' + ZAJAVITEL_ID + ''';
1067.           ADOQuery1.Close;
1068.           ADOQuery1.SQL.Clear;
1069.           ADOQuery1.SQL.Append( SQL_str );
1070.           try
1071.             ADOQuery1.ExecSQL;
1072.           except on E : Exception do
1073.             begin
1074.               ShowMessage( E.ClassName + ' db error: ' + E.Message );
1075.               Exit;
1076.             end;
1077.           end;
1078.         end;
1079.       end;
1080.     end;
1081.   procedure TPNP_StajPoDrugDocViewForm.FormClose(Sender: TObject;
1082.     var Action: TCloseAction);

```

```

1083. begin
1084. CalcCommonStaj;
1085. CalcCommonStajPoDrugDoc;
1086. try
1087. if ADODataset1.RecordCount > 0 then
1088. ADODataset1.Active := false;
1089. except
1090. end;
1091. end;
1092. end.
1093.
1094. unit uPNP_TrudKnigaViewForm;
1095. interface
1096. uses
1097. Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
1098. Buttons, ExtCtrls, StdCtrls, Spin, Db, ADODB, Grids, DBGrids, uPNP_ConnectionForm,
1099. Mask, ToolEdit;
1100. type
1101. TPNPWorkRecord = record
1102. ZAPIS_TRUD_ID: String;
1103. n_priema : String;
1104. god_priema : String;
1105. mes_priema : String;
1106. cislo_priema : String;
1107. n_uvoldn : String;
1108. god_uvoldn : String;
1109. mes_uvoldn : String;
1110. cislo_uvoldn : String;
1111. end;
1112. TPNPWorkRecordList = array of TPNPWorkRecord;
1113. TPNP_TrudKnigaViewForm = class(TForm)
1114. Panel1: TPanel;
1115. SpeedButton1: TSpeedButton;
1116. SpeedButton2: TSpeedButton;
1117. SpeedButton3: TSpeedButton;
1118. SpeedButton4: TSpeedButton;
1119. GroupBox1: TGroupBox;
1120. Label1: TLabel;
1121. Year: TSpinEdit;
1122. Month: TSpinEdit;
1123. Label2: TLabel;
1124. Day: TSpinEdit;
1125. Label3: TLabel;
1126. Label4: TLabel;
1127. EnterComment: TEdit;
1128. EnterOrder: TEdit;
1129. Label5: TLabel;
1130. EscapeComment: TEdit;
1131. Label6: TLabel;
1132. EscapeOrder: TEdit;
1133. Label7: TLabel;
1134. StringGrid1: TStringGrid;
1135. StringGrid2: TStringGrid;
1136. ADOQuery1: TADOQuery;
1137. bCalculation: TButton;
1138. FillDate: TDateEdit;
1139. Label8: TLabel;
1140. Label9: TLabel;
1141. DaysForCorrect: TSpinEdit;
1142. procedure SpeedButton4Click(Sender: TObject);
1143. procedure SpeedButton3Click(Sender: TObject);
1144. procedure SpeedButton2Click(Sender: TObject);
1145. procedure SpeedButton1Click(Sender: TObject);
1146. procedure FormShow(Sender: TObject);
1147. procedure UpDateTables;
1148. procedure UpDateSmallTable( ID: String );
1149. procedure StringGrid1SelectCell(Sender: TObject; ACol, ARow: Integer;
1150. var CanSelect: Boolean);
1151. procedure bCalculationClick(Sender: TObject);
1152. procedure CalcCommonStajByTrud;
1153. procedure CalcCommonStajLgot;
1154. procedure CalcCommonStaj;
1155. procedure FormClose(Sender: TObject; var Action: TCloseAction);
1156. private
1157. { Private declarations }
1158. BookWork: TPNPWorkRecordList;
1159. public
1160. { Public declarations }

```



```

1161. ZAJAVITEL_ID: String;
1162. TRUD_KNIGA_ID: String;
1163. PREDSTAV_ID: String;
1164. end;
1165. var
1166. PNP_TrudKnigaViewForm: TPNP_TrudKnigaViewForm;
1167. implementation
1168. uses uPNP_TrudKnigaEditForm;
1169. {$R *.DFM}
1170. procedure TPNP_TrudKnigaViewForm.SpeedButton4Click(Sender: TObject);
1171. var
1172. Msg, SQL_str: String;
1173. SelectedRow: Integer;
1174. begin
1175. if ( StringGrid1.Row > 0 ) and ( StringGrid1.Row < StringGrid1.RowCount ) then
1176. begin
1177. Msg := 'Выбранная запись будет удалена.' + #10#13 + 'Удалить запись?';
1178. if ( mrOK = MessageDlg( Msg, mtWarning, mbOKCancel, 0 ) ) then
1179. begin
1180. SQL_str := 'delete from ZAPIS_TRUD where ZAPIS_TRUD_ID = "' +
1181. BookWork[ StringGrid1.Row-1 ].ZAPIS_TRUD_ID + '"';
1182. ADOQuery1.Close;
1183. ADOQuery1.SQL.Clear;
1184. ADOQuery1.SQL.Append( SQL_str );
1185. try
1186. ADOQuery1.ExecSQL;
1187. except on E : Exception do
1188. begin
1189. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1190. Exit;
1191. end;
1192. end;
1193. SelectedRow := StringGrid1.Row;
1194. UpdateTables;
1195. if StringGrid1.RowCount > 1 then
1196. begin
1197. if SelectedRow > StringGrid1.RowCount-1 then
1198. SelectedRow := StringGrid1.RowCount-1;
1199. StringGrid1.Row := SelectedRow;
1200. UpdateSmallTable( BookWork[SelectedRow-1].ZAPIS_TRUD_ID );
1201. end;
1202. end;
1203. end;
1204. {
1205. Msg := 'Выбранная запись будет удалена.' + #10#13 + 'Удалить запись?';
1206. if ( mrOK = MessageDlg( Msg, mtWarning, mbOKCancel, 0 ) ) then
1207. begin
1208. ShowMessage( 'Нажата кнопка ОК' );
1209. end;
1210. }
1211. end;
1212. procedure TPNP_TrudKnigaViewForm.SpeedButton3Click(Sender: TObject);
1213. begin
1214. if High( BookWork ) > -1 then
1215. begin
1216. PNP_TrudKnigaEditForm.Caption := 'Изменение записи трудовой книги';
1217. PNP_TrudKnigaEditForm.SetTableName( 'ZAPIS_TRUD' );
1218. PNP_TrudKnigaEditForm.SetID_Field( 'ZAPIS_TRUD_ID' );
1219. PNP_TrudKnigaEditForm.SetID( StrToInt( BookWork[ StringGrid1.Row-1 ].ZAPIS_TRUD_ID ) );
1220. PNP_TrudKnigaEditForm.SetNewRecord( false );
1221. PNP_TrudKnigaEditForm.TRUD_KNIGA_ID.Text := TRUD_KNIGA_ID;
1222. PNP_TrudKnigaEditForm.ZAPIS_TRUD_ID.Text := BookWork[ StringGrid1.Row-1 ].ZAPIS_TRUD_ID;
1223. PNP_TrudKnigaEditForm.ShowModal;
1224. end;
1225. end;
1226. procedure TPNP_TrudKnigaViewForm.SpeedButton2Click(Sender: TObject);
1227. var
1228. SQL_str: String;
1229. v: Variant;
1230. MAX: Integer;
1231. begin
1232. if StringGrid1.Row > 0 then
1233. begin
1234. MAX := 0;
1235. if TRUD_KNIGA_ID <> '' then
1236. begin
1237. SQL_str := 'select case when max(coalesce(N_PRIEMA,0)) > max(coalesce(N_UVOLN,0)) then ' +
1238. 'max(N_PRIEMA) else max(N_UVOLN) end as max_n from ZAPIS_TRUD where ' +

```

```

1239. 'ZAPIS_TRUD.trud_kniga_id=' + TRUD_KNIGA_ID + ''";
1240. ADOQuery1.Close;
1241. ADOQuery1.SQL.Clear;
1242. ADOQuery1.SQL.Append( SQL_str );
1243. try
1244. ADOQuery1.Open;
1245. except on E : Exception do
1246. begin
1247. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1248. ADOQuery1.Close;
1249. Exit;
1250. end;
1251. end;
1252. if ADOQuery1.RecordCount > 0 then
1253. begin
1254. v := ADOQuery1.FieldValues[ 'max_n' ];
1255. if NULL <> v then MAX := v;
1256. end;
1257. end;
1258. PNP_TrudKnigaEditForm.Caption := 'Добавление записи в трудовую книжку';
1259. PNP_TrudKnigaEditForm.SetTableName( 'ZAPIS_TRUD' );
1260. PNP_TrudKnigaEditForm.SetID_Field( 'ZAPIS_TRUD_ID' );
1261. PNP_TrudKnigaEditForm.SetNewRecord( True );
1262. PNP_TrudKnigaEditForm.TRUD_KNIGA_ID.Text := TRUD_KNIGA_ID;
1263. PNP_TrudKnigaEditForm.ZAJAVITEL_ID := ZAJAVITEL_ID;
1264. // PNP_TrudKnigaEditForm.ZAPIS_TRUD_ID.Text := BookWork[ StringGrid1.Row-1 ].ZAPIS_TRUD_ID;
1265. PNP_TrudKnigaEditForm.TrudKnigaEnterNumber.Value := MAX+1;
1266. PNP_TrudKnigaEditForm.TrudKnigaEscapeNumber.Value := MAX+2;
1267. if ( mrOK = PNP_TrudKnigaEditForm.ShowModal ) then
1268. begin
1269. UpDateTables;
1270. end;
1271. end;
1272. end;
1273. procedure TPNP_TrudKnigaViewForm.SpeedButton1Click(Sender: TObject);
1274. begin
1275. Close;
1276. end;
1277. procedure TPNP_TrudKnigaViewForm.UpDateSmallTable( ID: String );
1278. var
1279. i: Integer;
1280. SQL_str: String;
1281. begin
1282. StringGrid2.RowCount := 2;
1283. StringGrid2.Rows[0].Clear;
1284. StringGrid2.Rows[1].Clear;
1285. StringGrid2.Rows[0].Append( 'Номер льготного списка' );
1286. StringGrid2.Rows[0].Append( 'Лет льготного стажа' );
1287. StringGrid2.Rows[0].Append( 'Месяцев льготного стажа' );
1288. StringGrid2.Rows[0].Append( 'Дней льготного стажа' );
1289. if ID <> '' then
1290. begin
1291. ADOQuery1.Close;
1292. SQL_str := 'select l.nomer_lgotnogo_spiska, l.let, l.mes, l.dnej from lgot_staj l where l.zapis_trud_id = ' + ID + ''";
1293. ADOQuery1.SQL.Clear;
1294. ADOQuery1.SQL.Append( SQL_str );
1295. try
1296. ADOQuery1.Open;
1297. except on E : Exception do
1298. begin
1299. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1300. ADOQuery1.Close;
1301. Exit;
1302. end;
1303. end;
1304. if ADOQuery1.RecordCount > 0 then
1305. begin
1306. StringGrid2.RowCount := ADOQuery1.RecordCount + 1;
1307. ADOQuery1.First;
1308. for i := 1 to ADOQuery1.RecordCount do
1309. begin
1310. StringGrid2.Cells[ 0, i ] := ADOQuery1.FieldValues['nomer_lgotnogo_spiska'];
1311. StringGrid2.Cells[ 1, i ] := ADOQuery1.FieldValues['let'];
1312. StringGrid2.Cells[ 2, i ] := ADOQuery1.FieldValues['mes'];
1313. StringGrid2.Cells[ 3, i ] := ADOQuery1.FieldValues['dnej'];
1314. if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1315. end;
1316. end;

```

```

1317. end;
1318. end;
1319. procedure TPNP_TrudKnigaViewForm.UpDateTables;
1320. var
1321. i: Integer;
1322. SQL_str: String;
1323. v: Variant;
1324. begin
1325. StringGrid1.RowCount := 2;
1326. StringGrid1.Rows[0].Clear;
1327. StringGrid1.Rows[1].Clear;
1328. StringGrid1.Rows[0].Append( 'Номер приема' );
1329. StringGrid1.Rows[0].Append( 'Год приема' );
1330. StringGrid1.Rows[0].Append( 'Месяц приема' );
1331. StringGrid1.Rows[0].Append( 'Число приема' );
1332. StringGrid1.Rows[0].Append( 'Номер увольнения' );
1333. StringGrid1.Rows[0].Append( 'Год увольнения' );
1334. StringGrid1.Rows[0].Append( 'Месяц увольнения' );
1335. StringGrid1.Rows[0].Append( 'Число увольнения' );
1336. if ZAJAVITEL_ID <> '' then
1337. begin
1338. SQL_str := 'select t.data_zapoln, z.ZAPIS_TRUD_ID, z.n_priema, z.god_priema, z.mes_priema, ' +
1339. 'z.cislo_priema, z.sved_po_stroke_priem, z.osnovanie_stroki_priem, ' +
1340. 'z.n_uvoldn, z.god_uvoldn, z.mes_uvoldn, z.cislo_uvoldn, z.sved_po_stroke_uvold, ' +
1341. 'z.osnovanie_stroki_uvold, dl.let, dl.mes, dl.dnej, dl.kor_day ' +
1342. 'from dlit_staj_rabot dl right outer join ' +
1343. '(trud_kniga t join zapis_trud z on t.trud_kniga_id=z.trud_kniga_id) ' +
1344. 'on dl.zapis_trud_id=z.zapis_trud_id where t.zajavitel_id = '' + ZAJAVITEL_ID + ''';
1345. ADOQuery1.Close;
1346. ADOQuery1.SQL.Clear;
1347. ADOQuery1.SQL.Append( SQL_str );
1348. try
1349. ADOQuery1.Open;
1350. except on E : Exception do
1351. begin
1352. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1353. Exit;
1354. end;
1355. end;
1356. if ADOQuery1.RecordCount > 0 then
1357. begin
1358. StringGrid1.RowCount := ADOQuery1.RecordCount + 1;
1359. if StringGrid1.RowCount < 2 then StringGrid1.RowCount := 2;
1360. v := ADOQuery1.FieldValues['data_zapoln'];
1361. if v <> NULL then FillDate.Text := v;
1362. v := ADOQuery1.FieldValues['let'];
1363. if v <> NULL then Year.Text := v;
1364. v := ADOQuery1.FieldValues['mes'];
1365. if v <> NULL then Month.Text := v;
1366. v := ADOQuery1.FieldValues['dnej'];
1367. if v <> NULL then Day.Text := v;
1368. v := ADOQuery1.FieldValues['kor_day'];
1369. if v <> NULL then DaysForCorrect.Text := v;
1370. v := ADOQuery1.FieldValues['sved_po_stroke_priem'];
1371. if v <> NULL then EnterComment.Text := v;
1372. v := ADOQuery1.FieldValues['osnovanie_stroki_priem'];
1373. if v <> NULL then EnterOrder.Text := v;
1374. v := ADOQuery1.FieldValues['sved_po_stroke_uvold'];
1375. if v <> NULL then EscapeComment.Text := v;
1376. v := ADOQuery1.FieldValues['osnovanie_stroki_uvold'];
1377. if v <> NULL then EscapeOrder.Text := v;
1378. ADOQuery1.First;
1379. SetLength( BookWork, ADOQuery1.RecordCount );
1380. for i := 1 to ADOQuery1.RecordCount do
1381. begin
1382. v := ADOQuery1.FieldValues['ZAPIS_TRUD_ID'];
1383. if v <> NULL then BookWork[i-1].ZAPIS_TRUD_ID := v;
1384. v := ADOQuery1.FieldValues['n_priema'];
1385. if v <> NULL then BookWork[i-1].n_priema := v;
1386. v := ADOQuery1.FieldValues['god_priema'];
1387. if v <> NULL then BookWork[i-1].god_priema := v;
1388. v := ADOQuery1.FieldValues['mes_priema'];
1389. if v <> NULL then BookWork[i-1].mes_priema := v;
1390. v := ADOQuery1.FieldValues['cislo_priema'];
1391. if v <> NULL then BookWork[i-1].cislo_priema := v;
1392. v := ADOQuery1.FieldValues['n_uvoldn'];
1393. if v <> NULL then BookWork[i-1].n_uvoldn := v;
1394. v := ADOQuery1.FieldValues['god_uvoldn'];

```

```

1395.   if v <> NULL then BookWork[i-1].god_uvoldn := v;
1396.   v := ADOQuery1.FieldValues['mes_uvoldn'];
1397.   if v <> NULL then BookWork[i-1].mes_uvoldn := v;
1398.   v := ADOQuery1.FieldValues['cislo_uvoldn'];
1399.   if v <> NULL then BookWork[i-1].cislo_uvoldn := v;
1400.   StringGrid1.Cells[ 0, i ] := BookWork[i-1].n_priema ;
1401.   StringGrid1.Cells[ 1, i ] := BookWork[i-1].god_priema ;
1402.   StringGrid1.Cells[ 2, i ] := BookWork[i-1].mes_priema ;
1403.   StringGrid1.Cells[ 3, i ] := BookWork[i-1].cislo_priema;
1404.   StringGrid1.Cells[ 4, i ] := BookWork[i-1].n_uvoldn ;
1405.   StringGrid1.Cells[ 5, i ] := BookWork[i-1].god_uvoldn ;
1406.   StringGrid1.Cells[ 6, i ] := BookWork[i-1].mes_uvoldn ;
1407.   StringGrid1.Cells[ 7, i ] := BookWork[i-1].cislo_uvoldn ;
1408.   if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1409.   end;
1410.   UpdateSmallTable( BookWork[0].ZAPIS_TRUD_ID );
1411.   end;
1412.   end
1413.   else
1414.   begin
1415.   UpdateSmallTable( " );
1416.   end;
1417.   end;
1418.   procedure TPNP_TrudKnigaViewForm.FormShow(Sender: TObject);
1419.   begin
1420.   UpdateTables;
1421.   end;
1422.   procedure TPNP_TrudKnigaViewForm.StringGrid1.SelectCell(Sender: TObject;
1423.   ACol, ARow: Integer; var CanSelect: Boolean);
1424.   begin
1425.   UpdateSmallTable( BookWork[ARow-1].ZAPIS_TRUD_ID );
1426.   end;
1427.   procedure TPNP_TrudKnigaViewForm.bCalculationClick(Sender: TObject);
1428.   var
1429.   SQL_str: String;
1430.   NOM, GOD_U, MES_U, CISLO_U: Integer;
1431.   GOD_P, MES_P, CISLO_P: Integer;
1432.   DPU: Integer;
1433.   Found, Flag: Boolean;
1434.   i: Integer;
1435.   v, ZAPIS_TRUD_ID: variant;
1436.   DUDPU, DUPD, DUPM, DUPM1: Integer;
1437.   DAY, MES, LET: Integer;
1438.   begin
1439.   NOM := 1;
1440.   GOD_U := 0;
1441.   MES_U := 0;
1442.   CISLO_U := 0;
1443.   SQL_str := 'select * from ZAPIS_TRUD where TRUD_KNIGA_ID = "' + TRUD_KNIGA_ID + '"';
1444.   ADOQuery1.Close;
1445.   ADOQuery1.SQL.Clear;
1446.   ADOQuery1.SQL.Append( SQL_str );
1447.   try
1448.   ADOQuery1.Open;
1449.   except on E : Exception do
1450.   begin
1451.   ShowMessage( E.ClassName + ' db error: ' + E.Message );
1452.   Exit;
1453.   end;
1454.   end;
1455.   if ADOQuery1.RecordCount > 0 then
1456.   begin
1457.   ZAPIS_TRUD_ID := ADOQuery1.FieldValues[ 'ZAPIS_TRUD_ID' ];
1458.   while True do
1459.   begin
1460.   Found := False;
1461.   ADOQuery1.First;
1462.   for i := 1 to ADOQuery1.RecordCount do
1463.   begin
1464.   v := ADOQuery1.FieldValues[ 'N_PRIEMA' ];
1465.   if ( v <> NULL ) and ( v = NOM ) then
1466.   begin
1467.   Found := True;
1468.   Break;
1469.   end;
1470.   if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1471.   end;
1472.   if Found then

```

```

1473. begin
1474. v := ADOQuery1.FieldValues[ 'ISP_RASCH' ];
1475. if ( NULL = v ) or (( v <> NULL ) and ( v = 0 )) then
1476. begin
1477. INC( NOM );
1478. continue;
1479. end;
1480. v := ADOQuery1.FieldValues[ 'GOD_PRIEMA' ];
1481. if v <> NULL then GOD_P := v;
1482. v := ADOQuery1.FieldValues[ 'MES_PRIEMA' ];
1483. if v <> NULL then MES_P := v;
1484. v := ADOQuery1.FieldValues[ 'CISLO_PRIEMA' ];
1485. if v <> NULL then CISLO_P := v;
1486. Inc( NOM );
1487. DPU := (GOD_P * 360 + MES_P * 30 + CISLO_P) - (GOD_U * 360 + MES_U * 30 + CISLO_U + 1);
1488. Flag := True;
1489. while Flag do
1490. begin
1491. Found := False;
1492. ADOQuery1.First;
1493. for i := 1 to ADOQuery1.RecordCount do
1494. begin
1495. v := ADOQuery1.FieldValues[ 'N_UVOLN' ];
1496. if ( v <> NULL ) and ( v = NOM ) then
1497. begin
1498. Found := True;
1499. Break;
1500. end;
1501. if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1502. end;
1503. if Found then
1504. begin
1505. v := ADOQuery1.FieldValues[ 'GOD_UVOLN' ];
1506. if NULL <> v then GOD_U := v else Break;
1507. v := ADOQuery1.FieldValues[ 'MES_UVOLN' ];
1508. if NULL <> v then MES_U := v else Break;
1509. v := ADOQuery1.FieldValues[ 'CISLO_UVOLN' ];
1510. if NULL <> v then CISLO_U := v else Break;
1511. Inc( NOM );
1512. end
1513. else
1514. begin
1515. ADOQuery1.First;
1516. for i := 1 to ADOQuery1.RecordCount do
1517. begin
1518. v := ADOQuery1.FieldValues[ 'N_PRIEMA' ];
1519. if ( v <> NULL ) and ( v = NOM ) then
1520. begin
1521. Found := True;
1522. Break;
1523. end;
1524. if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1525. end;
1526. if Found then
1527. begin
1528. Inc( NOM );
1529. Continue;
1530. end
1531. else
1532. begin
1533. MessageDlg( 'Нарушена целостность нумерации записей', mtError, [mbOK], 0 );
1534. Flag := False;
1535. Continue;
1536. end;
1537. end;
1538. end; // while Flag
1539. if not Flag then Break;
1540. DUDPU := CISLO_U + DPU - CISLO_P;
1541. DUPD := CISLO_U - CISLO_P;
1542. DUPM := MES_U - MES_P;
1543. DUPM1 := ( MES_U - 1 ) - MES_P;
1544. // Расчет длительности стажа по трудовой книжке:
1545. DAY := 0;
1546. MES := 0;
1547. // Расчет длительности дней:
1548. if (DPU < 0) and (DUDPU >= 0) then DAY := DUDPU;
1549. if (DPU < 0) and (DUDPU < 0) then DAY := DUDPU + 30;
1550. if (DPU > 0) and (DUPD >= 0) then DAY := DUPD;

```

```

1551.   if (DPU > 0) and (DUPD < 0) then DAY := DUPD + 30;
1552.   // Расчет длительности месяцев:
1553.   if (DPU < 0) and (DUDPU >= 0) and (DUPM >= 0) then MES := DUPM;
1554.   if (DPU < 0) and (DUDPU >= 0) and (DUPM < 0) then MES := DUPM + 12;
1555.   if (DPU < 0) and (DUDPU < 0) and (DUPM1 >= 0) then MES := DUPM1;
1556.   if (DPU < 0) and (DUDPU < 0) and (DUPM1 < 0) then MES := DUPM1+12;
1557.   if (DPU > 0) and (DUPD >= 0) and (DUPM >= 0) then MES := DUPM;
1558.   if (DPU > 0) and (DUPD >= 0) and (DUPM < 0) then MES := DUPM + 12;
1559.   if (DPU > 0) and (DUPD < 0) and (DUPM1 >= 0) then MES := DUPM1;
1560.   if (DPU > 0) and (DUPD < 0) and (DUPM1 < 0) then MES := DUPM1+12;
1561.   // Расчет длительности лет:
1562.   LET := Trunc( ((GOD_U * 360 + MES_U * 30 + CISLO_U) - (GOD_P * 360 + MES_P * 30 + CISLO_P))/360 );
1563.   SQL_str := 'select * from DLIT_STAJ_RABOT where ZAPIS_TRUD_ID = ' + ZAPIS_TRUD_ID + ''';
1564.   ADOQuery1.Close;
1565.   ADOQuery1.SQL.Clear;
1566.   ADOQuery1.SQL.Append( SQL_str );
1567.   try
1568.   ADOQuery1.Open;
1569.   except on E : Exception do
1570.   begin
1571.   ShowMessage( E.ClassName + ' db error: ' + E.Message );
1572.   Exit;
1573.   end;
1574.   end;
1575.   if ADOQuery1.RecordCount > 0 then
1576.   begin
1577.   v := ADOQuery1.FieldValues[ 'DLIT_STAJ_RABOT_ID' ];
1578.   SQL_str := 'update DLIT_STAJ_RABOT set ' +
1579.   'LET = ' + IntToStr( LET ) + ',' +
1580.   'MES = ' + IntToStr( MES ) + ',' +
1581.   'DNEJ = ' + IntToStr( DAY ) + '' +
1582.   ' where DLIT_STAJ_RABOT_ID = ' + v + ''';
1583.   ADOQuery1.Close;
1584.   ADOQuery1.SQL.Clear;
1585.   ADOQuery1.SQL.Append( SQL_str );
1586.   try
1587.   ADOQuery1.ExecSQL;
1588.   except on E : Exception do
1589.   begin
1590.   ShowMessage( E.ClassName + ' db error: ' + E.Message );
1591.   Exit;
1592.   end;
1593.   end;
1594.   end
1595.   else
1596.   begin
1597.   SQL_str := 'insert into DLIT_STAJ_RABOT( LET, MES, DNEJ, KOR_DAY, ZAPIS_TRUD_ID )values(' +
1598.   '' + IntToStr( LET ) + ',' +
1599.   '' + IntToStr( MES ) + ',' +
1600.   '' + IntToStr( DAY ) + ',' +
1601.   ''1, '' +
1602.   '' + ZAPIS_TRUD_ID + '''';
1603.   ADOQuery1.Close;
1604.   ADOQuery1.SQL.Clear;
1605.   ADOQuery1.SQL.Append( SQL_str );
1606.   try
1607.   ADOQuery1.ExecSQL;
1608.   except on E : Exception do
1609.   begin
1610.   ShowMessage( E.ClassName + ' db error: ' + E.Message );
1611.   Exit;
1612.   end;
1613.   end;
1614.   end;
1615.   end; // if Found
1616.   end; // while True
1617.   end; // if ADOQuery1.RecordCount > 0
1618.   end;
1619.   procedure TPNP_TrudKnigaViewForm.CalcCommonStajByTrud; //21
1620.   var
1621.   SQL_str: String;
1622.   LET, MES, DAY: Integer;
1623.   v: variant;
1624.   i: Integer;
1625.   A1, A2: Integer;
1626.   begin
1627.   SQL_str := 'select d.let, d.mes, d.dnej, d.kor_day from dlit_staj_rabot d ' +
1628.   'where d.zapis_trud_id in (select zapis_trud_id from zapis_trud where ' +

```

```

1629. 'trud_kniga_id = "" + TRUD_KNIGA_ID + "" );';
1630. ADOQuery1.Close;
1631. ADOQuery1.SQL.Clear;
1632. ADOQuery1.SQL.Append( SQL_str );
1633. try
1634. ADOQuery1.Open;
1635. except on E : Exception do
1636. begin
1637. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1638. Exit;
1639. end;
1640. end;
1641. if ADOQuery1.RecordCount > 0 then
1642. begin
1643. ADOQuery1.First;
1644. LET := 0; MES := 0; DAY := 0;
1645. for i := 1 to ADOQuery1.RecordCount do
1646. begin
1647. v := ADOQuery1.FieldValues[ 'LET' ];
1648. LET := LET + v;
1649. v := ADOQuery1.FieldValues[ 'MES' ];
1650. MES := MES + v;
1651. v := ADOQuery1.FieldValues[ 'DNEJ' ];
1652. DAY := DAY + v;
1653. v := ADOQuery1.FieldValues[ 'KOR_DAY' ];
1654. DAY := DAY + v;
1655. if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1656. end;
1657. A1 := Trunc( DAY/30 );
1658. A2 := Trunc( (MES + A1)/12 );
1659. DAY := DAY - A1 * 30;
1660. MES := (MES + A1) - A2 * 12;
1661. LET := LET + A2;
1662. SQL_str := 'select * from TRUD_KNIGA where TRUD_KNIGA_ID = "" + TRUD_KNIGA_ID + ""';
1663. ADOQuery1.Close;
1664. ADOQuery1.SQL.Clear;
1665. ADOQuery1.SQL.Append( SQL_str );
1666. try
1667. ADOQuery1.Open;
1668. except on E : Exception do
1669. begin
1670. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1671. Exit;
1672. end;
1673. end;
1674. if ADOQuery1.RecordCount > 0 then
1675. begin
1676. SQL_str := 'update TRUD_KNIGA set ' +
1677. 'SUM_LET = "" + IntToStr( LET ) + "";' +
1678. 'SUM_MES = "" + IntToStr( MES ) + "";' +
1679. 'SUM_DAY = "" + IntToStr( DAY ) + "";' +
1680. ' where TRUD_KNIGA_ID = "" + TRUD_KNIGA_ID + ""';
1681. ADOQuery1.Close;
1682. ADOQuery1.SQL.Clear;
1683. ADOQuery1.SQL.Append( SQL_str );
1684. try
1685. ADOQuery1.ExecSQL;
1686. except on E : Exception do
1687. begin
1688. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1689. Exit;
1690. end;
1691. end;
1692. end;
1693. end;
1694. end;
1695. procedure TPNP_TrudKnigaViewForm.CalcCommonStajLgot; //22
1696. var
1697. SQL_str: String;
1698. LET, MES, DAY: Integer;
1699. i, nomer_lgotnogo_spiska: Integer;
1700. v: variant;
1701. A1, A2: Integer;
1702. SUM_LGOT_STAJ_ID: String;
1703. begin
1704. for nomer_lgotnogo_spiska := 1 to 2 do
1705. begin
1706. SQL_str := 'select d.nomer_lgotnogo_spiska, d.let, d.mes, d.dnej ' +

```

```

1707. 'from lgot_staj d where d.nomer_lgotnogo_spiska = "" + IntToStr( nomer_lgotnogo_spiska ) +
1708. "" and d.zapis_trud_id in (select zapis_trud_id from zapis_trud ' +
1709. 'where trud_kniga_id = "" + TRUD_KNIGA_ID + "");';
1710. ADOQuery1.Close;
1711. ADOQuery1.SQL.Clear;
1712. ADOQuery1.SQL.Append( SQL_str );
1713. try
1714. ADOQuery1.Open;
1715. except on E : Exception do
1716. begin
1717. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1718. Exit;
1719. end;
1720. end;
1721. if ADOQuery1.RecordCount > 0 then
1722. begin
1723. ADOQuery1.First;
1724. LET := 0; MES := 0; DAY := 0;
1725. for i := 1 to ADOQuery1.RecordCount do
1726. begin
1727. v := ADOQuery1.FieldValues[ 'LET' ];
1728. if v <> NULL then LET := LET + v;
1729. v := ADOQuery1.FieldValues[ 'MES' ];
1730. if v <> NULL then MES := MES + v;
1731. v := ADOQuery1.FieldValues[ 'DNEJ' ];
1732. if v <> NULL then DAY := DAY + v;
1733. if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1734. end;
1735. A1 := Trunc( DAY/30 );
1736. A2 := Trunc( (MES + A1)/12 );
1737. DAY := DAY - A1 * 30;
1738. MES := (MES + A1) - A2 * 12;
1739. LET := LET + A2;
1740. SQL_str := 'select * from SUM_LGOT_STAJ where NOMER_LGOTNOGO_SPISKA = "" + IntToStr(
1741. nomer_lgotnogo_spiska ) +
1742. "" and PREDSTAV_ID = "" + PREDSTAV_ID + ""';
1743. ADOQuery1.Close;
1744. ADOQuery1.SQL.Clear;
1745. ADOQuery1.SQL.Append( SQL_str );
1746. try
1747. ADOQuery1.Open;
1748. except on E : Exception do
1749. begin
1750. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1751. Exit;
1752. end;
1753. end;
1754. if ADOQuery1.RecordCount > 0 then
1755. begin
1756. v := ADOQuery1.FieldValues[ 'SUM_LGOT_STAJ_ID' ];
1757. if NULL <> v then
1758. begin
1759. SUM_LGOT_STAJ_ID := v;
1760. SQL_str := 'update SUM_LGOT_STAJ set ' +
1761. 'SUM_LET_LGOT_STAJ = "" + IntToStr( LET ) + "" +
1762. 'SUM_MES_LGOT_STAJ = "" + IntToStr( MES ) + "" +
1763. 'SUM_DAY_LGOT_STAJ = "" + IntToStr( DAY ) + "" +
1764. ' where SUM_LGOT_STAJ_ID = "" + SUM_LGOT_STAJ_ID + ""';
1765. ADOQuery1.Close;
1766. ADOQuery1.SQL.Clear;
1767. ADOQuery1.SQL.Append( SQL_str );
1768. try
1769. ADOQuery1.ExecSQL;
1770. except on E : Exception do
1771. begin
1772. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1773. Exit;
1774. end;
1775. end;
1776. end
1777. else
1778. begin
1779. SQL_str := 'insert into SUM_LGOT_STAJ( NOMER_LGOTNOGO_SPISKA, ' +
1780. 'SUM_LET_LGOT_STAJ, SUM_MES_LGOT_STAJ, SUM_DAY_LGOT_STAJ, PREDSTAV_ID )values(' +
1781. '"" + IntToStr( nomer_lgotnogo_spiska ) + "" +
1782. '"" + IntToStr( LET ) + "" +
1783. '"" + IntToStr( MES ) + "" +

```



```

1784.    "" + IntToStr( DAY ) + "," +
1785.    "" + PREDSTAV_ID + "");';
1786.    ADOQuery1.Close;
1787.    ADOQuery1.SQL.Clear;
1788.    ADOQuery1.SQL.Append( SQL_str );
1789.    try
1790.    ADOQuery1.ExecSQL;
1791.    except on E : Exception do
1792.    begin
1793.    ShowMessage( E.ClassName + ' db error: ' + E.Message );
1794.    Exit;
1795.    end;
1796.    end;
1797.    end;
1798.    end;
1799.    end;
1800.    end;
1801.    procedure TPNP_TrudKnigaViewForm.CalcCommonStaj; //28
1802.    var
1803.    SQL_str: String;
1804.    LET, MES, DAY: Integer;
1805.    v: variant;
1806.    i: Integer;
1807.    A1, A2: Integer;
1808.    begin
1809.    SQL_str := 'select t.sum_let, t.sum_mes, t.sum_day, o.sum_other_let, o.sum_other_mes, ' +
1810.    'o.sum_other_day from predstav o join (trud_kniga t join zajavitel z ' +
1811.    'on t.zajavitel_id = z.zajavitel_id) on o.zajavitel_id = z.zajavitel_id ' +
1812.    'where z.zajavitel_id = "' + ZAJAVITEL_ID + "'";
1813.    ADOQuery1.Close;
1814.    ADOQuery1.SQL.Clear;
1815.    ADOQuery1.SQL.Append( SQL_str );
1816.    try
1817.    ADOQuery1.Open;
1818.    except on E : Exception do
1819.    begin
1820.    ShowMessage( E.ClassName + ' db error: ' + E.Message );
1821.    Exit;
1822.    end;
1823.    end;
1824.    if ADOQuery1.RecordCount > 0 then
1825.    begin
1826.    ADOQuery1.First;
1827.    LET := 0; MES := 0; DAY := 0;
1828.    for i := 1 to ADOQuery1.RecordCount do
1829.    begin
1830.    v := ADOQuery1.FieldValues[ 'SUM_LET' ];
1831.    if v <> NULL then LET := LET + v;
1832.    v := ADOQuery1.FieldValues[ 'SUM_OTHER_LET' ];
1833.    if v <> NULL then LET := LET + v;
1834.    v := ADOQuery1.FieldValues[ 'SUM_MES' ];
1835.    if v <> NULL then MES := MES + v;
1836.    v := ADOQuery1.FieldValues[ 'SUM_OTHER_MES' ];
1837.    if v <> NULL then MES := MES + v;
1838.    v := ADOQuery1.FieldValues[ 'SUM_DAY' ];
1839.    if v <> NULL then DAY := DAY + v;
1840.    v := ADOQuery1.FieldValues[ 'SUM_OTHER_DAY' ];
1841.    if v <> NULL then DAY := DAY + v;
1842.    if i <> ADOQuery1.RecordCount then ADOQuery1.Next;
1843.    end;
1844.    A1 := Trunc( DAY/30 );
1845.    A2 := Trunc( (MES + A1)/12 );
1846.    DAY := DAY - A1 * 30;
1847.    MES := ( MES + A1 ) - A2 * 12;
1848.    LET := LET + A2;
1849.    SQL_str := 'select * from PREDSTAV where ZAJAVITEL_ID = "' + ZAJAVITEL_ID + "'";
1850.    ADOQuery1.Close;
1851.    ADOQuery1.SQL.Clear;
1852.    ADOQuery1.SQL.Append( SQL_str );
1853.    try
1854.    ADOQuery1.Open;
1855.    except on E : Exception do
1856.    begin
1857.    ShowMessage( E.ClassName + ' db error: ' + E.Message );
1858.    Exit;
1859.    end;
1860.    end;
1861.    if ADOQuery1.RecordCount > 0 then

```

```

1862. begin
1863. SQL_str := 'update PREDSTAV set ' +
1864. 'SUM_LET = "" + IntToStr( LET ) + "";' +
1865. 'SUM_MES = "" + IntToStr( MES ) + "";' +
1866. 'SUM_DAY = "" + IntToStr( DAY ) + "" +
1867. ' where ZAJAVITEL_ID = "" + ZAJAVITEL_ID + ""';
1868. ADOQuery1.Close;
1869. ADOQuery1.SQL.Clear;
1870. ADOQuery1.SQL.Append( SQL_str );
1871. try
1872. ADOQuery1.ExecSQL;
1873. except on E : Exception do
1874. begin
1875. ShowMessage( E.ClassName + ' db error: ' + E.Message );
1876. Exit;
1877. end;
1878. end;
1879. end
1880. end;
1881. end;
1882. procedure TPNP_TrudKnigaViewForm.FormClose(Sender: TObject;
1883. var Action: TCloseAction);
1884. begin
1885. CalcCommonStajByTrud;
1886. CalcCommonStajLgot;
1887. CalcCommonStaj;
1888. end;
1889. end.
1890. interface
1891.
1892. uses
1893.   Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
1894.   StdCtrls, Buttons, Mask, ToolEdit, CurrEdit, Spin, ADODB, uPNP_ConnectionForm;
1895. type
1896.   TPNP_NetrudEditForm = class( TPNP_EditFormBase )
1897.     NetrudNumber: TSpinEdit;
1898.     NetrudFIO: TEdit;
1899.     NetrudRodich: TEdit;
1900.     BitBtn1: TBitBtn;
1901.     BitBtn2: TBitBtn;
1902.     Label1: TLabel;
1903.     Label2: TLabel;
1904.     Label3: TLabel;
1905.     Label4: TLabel;
1906.     NetrudBornDate: TDateEdit;
1907.     ZAJAVITEL_ID: TEdit;
1908.     procedure FillFieldDataArray; override;
1909.     procedure FormCreate(Sender: TObject);
1910.     procedure BitBtn1Click(Sender: TObject);
1911.   private
1912.     { Private declarations }
1913.   public
1914.     { Public declarations }
1915.   end;
1916.   var
1917.     PNP_NetrudEditForm: TPNP_NetrudEditForm;
1918. implementation
1919.   {$R *.DFM}
1920.   procedure TPNP_NetrudEditForm.FillFieldDataArray;
1921.   var
1922.     FieldData: TPNP_FieldData;
1923.   begin
1924.     FieldDataArrayLength := 4;
1925.     SetLength( FieldDataArray, FieldDataArrayLength );
1926.     FieldData := TPNP_FieldData.Create;
1927.     FieldData.SetData( 'ZAJAVITEL_ID', 'ZAJAVITEL_ID' , component_Edit );
1928.     FieldDataArray[0] := FieldData;
1929.     FieldData := TPNP_FieldData.Create;
1930.     FieldData.SetData( 'F_I_O', 'NetrudFIO' , component_Edit );
1931.     FieldDataArray[1] := FieldData;
1932.     FieldData := TPNP_FieldData.Create;
1933.     FieldData.SetData( 'RODSTVEN_SVAZ', 'NetrudRodich' , component_Edit );
1934.     FieldDataArray[2] := FieldData;
1935.     FieldData := TPNP_FieldData.Create;
1936.     FieldData.SetData( 'DATA_ROJD', 'NetrudBornDate' , component_Date );
1937.     FieldDataArray[3] := FieldData;
1938.   end;
1939.   procedure TPNP_NetrudEditForm.FormCreate(Sender: TObject);

```

```

1940. begin
1941.     ADOQuery1 := TADOQuery.Create( self );
1942.     ADOQuery1.Connection := PNP_ConnectionForm.ADOConnection1;
1943.     FillFieldDataArray;
1944.     SetTableName( 'NETRUD' );
1945.     SetID_Field( 'NETRUD_ID' );
1946. end;
1947. procedure TPNP_NetrudEditForm.BitBtn1Click(Sender: TObject);
1948. begin
1949.     ApplyData;
1950. end;
1951. end.
1952.
1953. unit uPNP_PensOrganEditForm;
1954. interface
1955. uses
1956.     Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
1957.     Db, ADODB, Grids, DBGrids, Buttons, ExtCtrls, StdCtrls, uPNP_ConnectionForm;
1958. type
1959.     TPNP_PensOrganEditForm = class( TPNP_EditFormBase )
1960.     Label1: TLabel;
1961.     PensName: TEdit;
1962.     Label2: TLabel;
1963.     PensLocation: TEdit;
1964.     Label3: TLabel;
1965.     PensRegion: TEdit;
1966.     BitBtn1: TBitBtn;
1967.     BitBtn2: TBitBtn;
1968.     procedure BitBtn1Click(Sender: TObject);
1969.     procedure FillFieldDataArray; override;
1970.     procedure FormCreate(Sender: TObject);
1971. private
1972.     { Private declarations }
1973. public
1974.     { Public declarations }
1975. end;
1976. var
1977.     PNP_PensOrganEditForm: TPNP_PensOrganEditForm;
1978. implementation
1979. {$R *.DFM}
1980. procedure TPNP_PensOrganEditForm.BitBtn1Click(Sender: TObject);
1981. begin
1982.     ApplyData;
1983. end;
1984. procedure TPNP_PensOrganEditForm.FillFieldDataArray;
1985. var
1986.     FieldData: TPNP_FieldData;
1987. begin
1988.     FieldDataArrayLength := 3;
1989.     SetLength( FieldDataArray, FieldDataArrayLength );
1990.     FieldData := TPNP_FieldData.Create;
1991.     FieldData.SetData( 'NAZVANIE', 'PensName' , component_Edit );
1992.     FieldDataArray[0] := FieldData;
1993.     FieldData := TPNP_FieldData.Create;
1994.     FieldData.SetData( 'NAS_PUNKT', 'PensLocation', component_Edit );
1995.     FieldDataArray[1] := FieldData;
1996.     FieldData := TPNP_FieldData.Create;
1997.     FieldData.SetData( 'OBLAST' , 'PensRegion' , component_Edit );
1998.     FieldDataArray[2] := FieldData;
1999. end;
2000. procedure TPNP_PensOrganEditForm.FormCreate(Sender: TObject);
2001. begin
2002.     ADOQuery1 := TADOQuery.Create( self );
2003.     ADOQuery1.Connection := PNP_ConnectionForm.ADOConnection1;
2004.     FillFieldDataArray;
2005.     SetTableName( 'PENS_ORGAN' );
2006.     SetID_Field( 'PENS_ORGAN_ID' );
2007. end;
2008. end.
2009.
2010. unit uPNP_PredprEditForm;
2011. interface
2012. uses
2013.     Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
2014.     StdCtrls, Buttons, Spin, Db, ADODB, Mask, DBCtrls, DBTables, uPNP_ConnectionForm;
2015. type
2016.     TPNP_PredprEditForm = class( TPNP_EditFormBase )
2017.     Label2: TLabel;

```

```

2018.   PredprPlace: TEdit;
2019.   Label3: TLabel;
2020.   PredprName: TEdit;
2021.   Label1: TLabel;
2022.   PredprStreet: TEdit;
2023.   Label4: TLabel;
2024.   Label5: TLabel;
2025.   PredprIndex: TSpinEdit;
2026.   PredprBuildingNumber: TSpinEdit;
2027.   PredprBuhPhone: TSpinEdit;
2028.   Label6: TLabel;
2029.   PredprKadrPhone: TSpinEdit;
2030.   Label7: TLabel;
2031.   PredprBossPhone: TSpinEdit;
2032.   Label8: TLabel;
2033.   PredprIOBossName: TEdit;
2034.   Label9: TLabel;
2035.   PredprIOProfName: TEdit;
2036.   Label10: TLabel;
2037.   BitBtn1: TBitBtn;
2038.   BitBtn2: TBitBtn;
2039.   procedure BitBtn1Click(Sender: TObject);
2040.   procedure FillFieldDataArray; override;
2041.   procedure FormCreate(Sender: TObject); // Замена абстрактного метода
2042.   private
2043.     { Private declarations }
2044.   public
2045.     { Public declarations }
2046.   end;
2047.   var
2048.     PNP_PredprEditForm: TPNP_PredprEditForm;
2049.   implementation
2050.   {$R *.DFM}
2051.   procedure TPNP_PredprEditForm.BitBtn1Click(Sender: TObject);
2052.   begin
2053.     ApplyData;
2054.   end;
2055.   procedure TPNP_PredprEditForm.FillFieldDataArray;
2056.   var
2057.     FieldData: TPNP_FieldData;
2058.   begin
2059.     FieldDataArrayLength := 10;
2060.     SetLength( FieldDataArray, FieldDataArrayLength );
2061.     FieldData := TPNP_FieldData.Create;
2062.     FieldData.SetData( 'NAZVANIE' , 'PredprName'      , component_Edit );
2063.     FieldDataArray[0] := FieldData;
2064.     FieldData := TPNP_FieldData.Create;
2065.     FieldData.SetData( 'INDEKS'   , 'PredprIndex'    , component_Spin );
2066.     FieldDataArray[1] := FieldData;
2067.     FieldData := TPNP_FieldData.Create;
2068.     FieldData.SetData( 'NAS_PUNKT' , 'PredprPlace' , component_Edit );
2069.     FieldDataArray[2] := FieldData;
2070.     FieldData := TPNP_FieldData.Create;
2071.     FieldData.SetData( 'ULICA'     , 'PredprStreet' , component_Edit );
2072.     FieldDataArray[3] := FieldData;
2073.     FieldData := TPNP_FieldData.Create;
2074.     FieldData.SetData( 'DOM'       , 'PredprBuildingNumber', component_Spin );
2075.     FieldDataArray[4] := FieldData;
2076.     FieldData := TPNP_FieldData.Create;
2077.     FieldData.SetData( 'TEL_RUK'   , 'PredprBossPhone'  , component_Spin );
2078.     FieldDataArray[5] := FieldData;
2079.     FieldData := TPNP_FieldData.Create;
2080.     FieldData.SetData( 'TEL_BUH'   , 'PredprBuhPhone'   , component_Spin );
2081.     FieldDataArray[6] := FieldData;
2082.     FieldData := TPNP_FieldData.Create;
2083.     FieldData.SetData( 'TEL_O_K'   , 'PredprKadrPhone'  , component_Spin );
2084.     FieldDataArray[7] := FieldData;
2085.     FieldData := TPNP_FieldData.Create;
2086.     FieldData.SetData( 'RUKOVODITEL', 'PredprIOBossName' , component_Edit );
2087.     FieldDataArray[8] := FieldData;
2088.     FieldData := TPNP_FieldData.Create;
2089.     FieldData.SetData( 'PROFSOUZ'  , 'PredprIOProfName' , component_Edit );
2090.     FieldDataArray[9] := FieldData;
2091.   end;
2092.   procedure TPNP_PredprEditForm.FormCreate(Sender: TObject);
2093.   begin
2094.     ADOQuery1 := TADOQuery.Create( self );
2095.     ADOQuery1.Connection := PNP_ConnectionForm.ADOConnection1;

```

```
2096.     FillFieldDataArray;
2097.     SetTableName( 'PREDPR' );
2098.     SetID_Field( 'PREDPR_ID' );
2099. end;
2100. end.
```

## ДОДАТОК С

### Комп'ютерна презентація

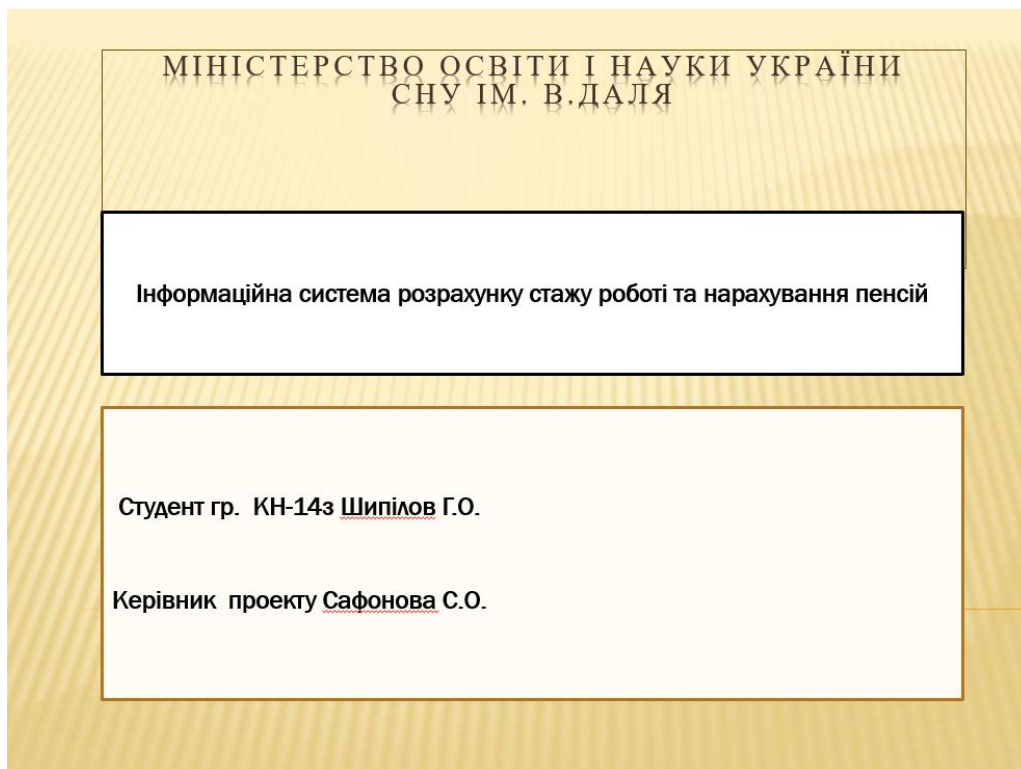


Рисунок С.1 – Слайд №1

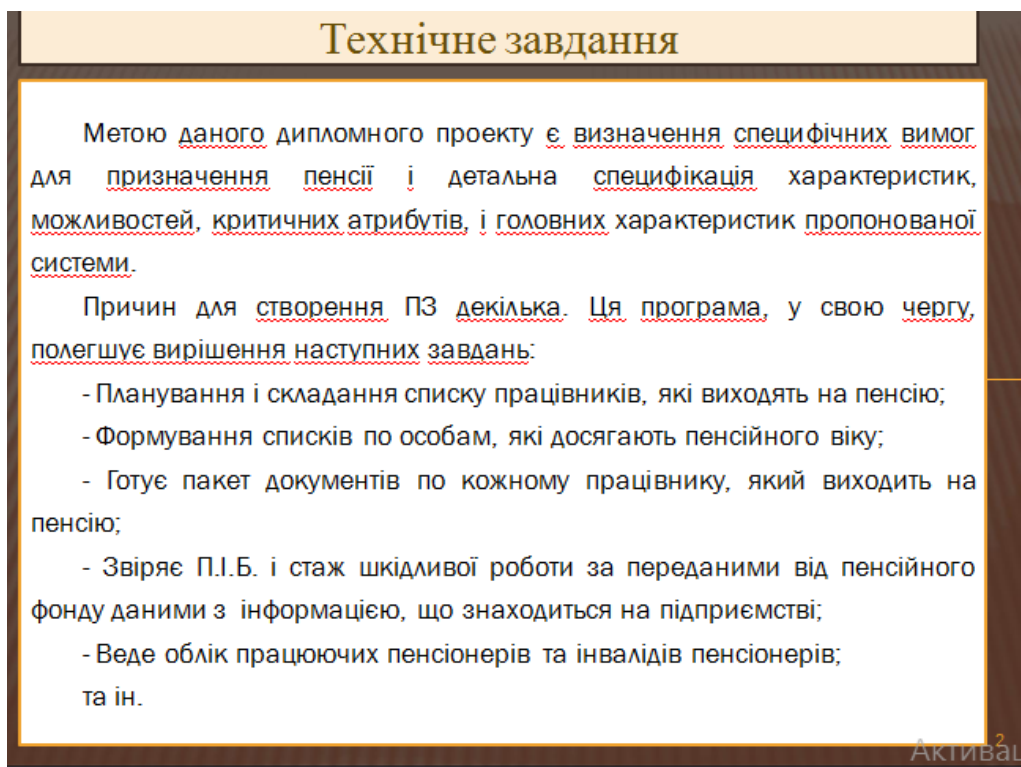


Рисунок С.2 – Слайд №2

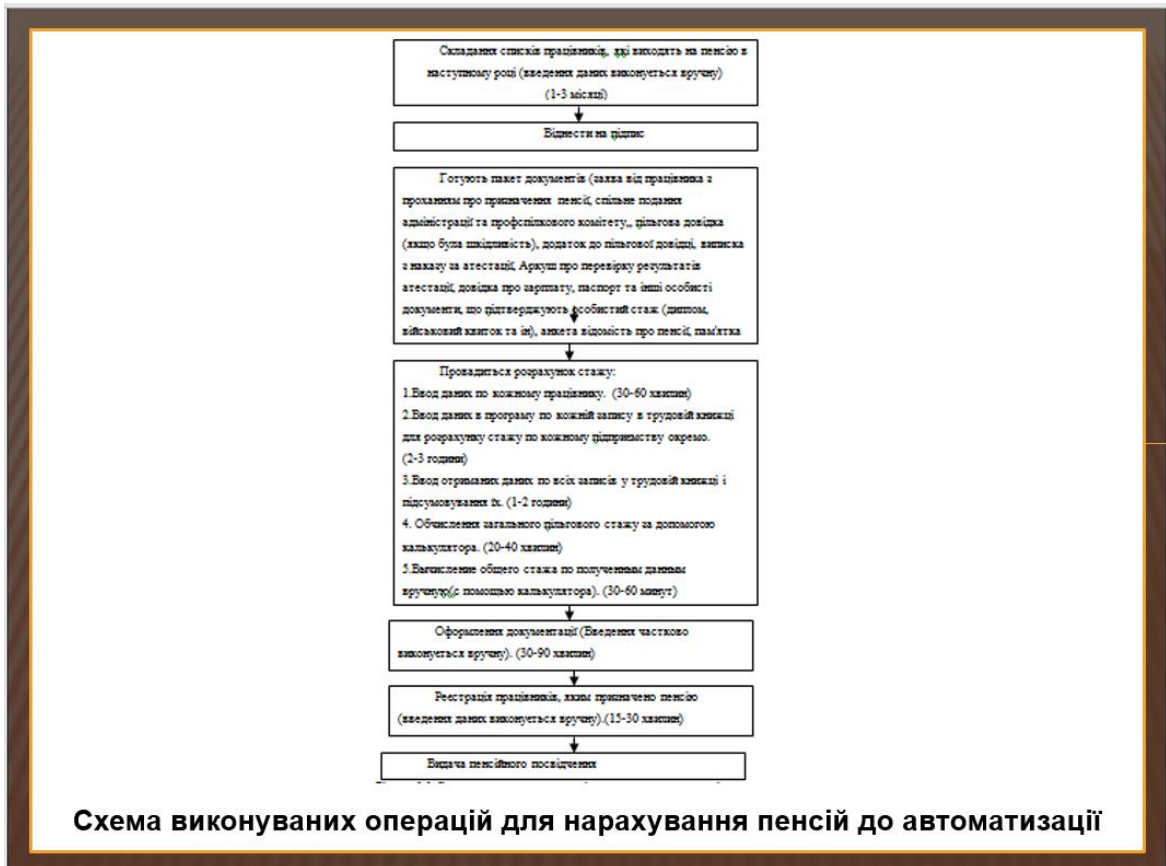


Рисунок С.3 – Слайд №3

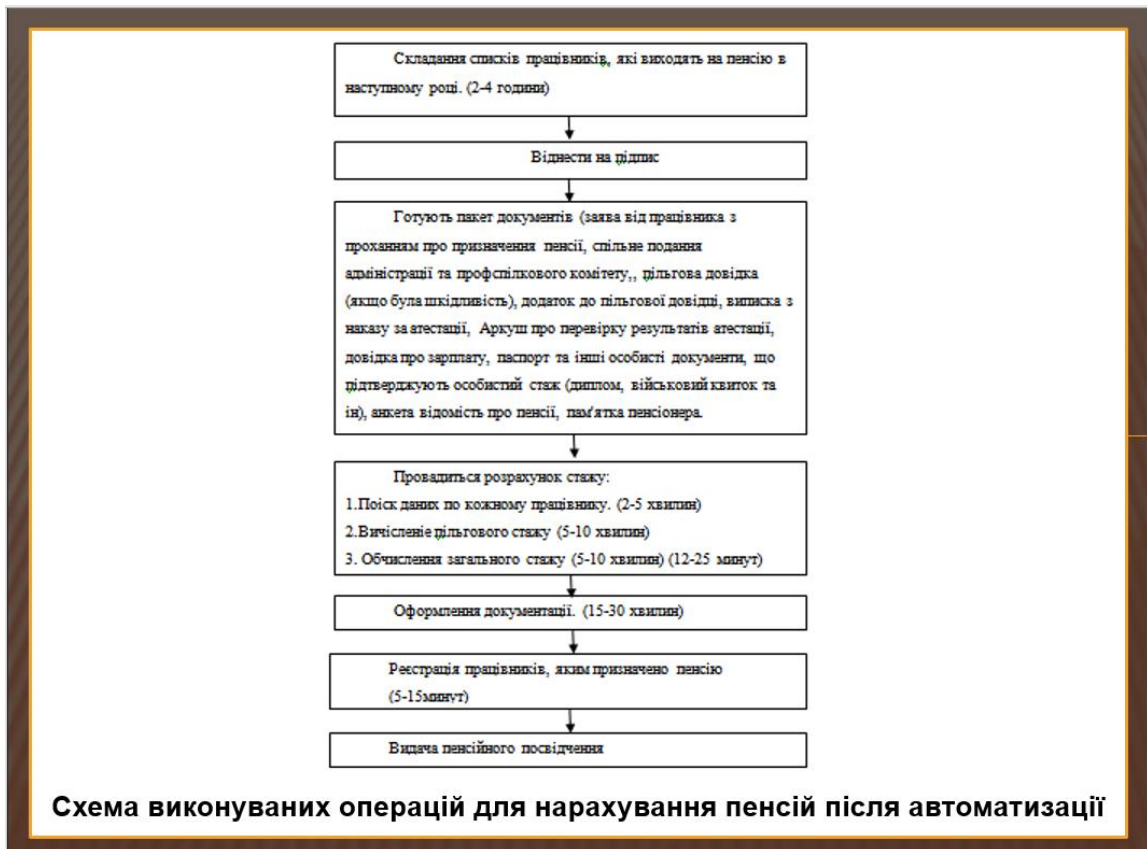


Рисунок С.4 – Слайд №4

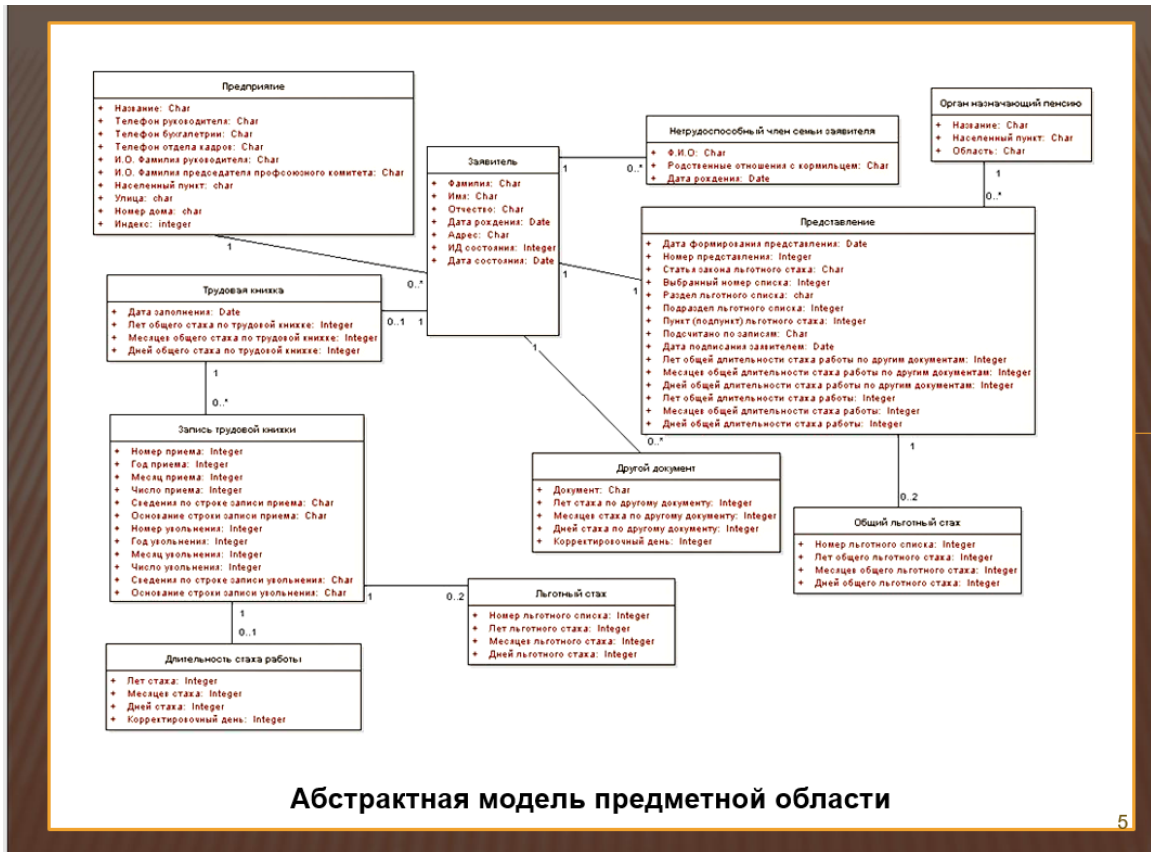


Рисунок С.5 – Слайд №5

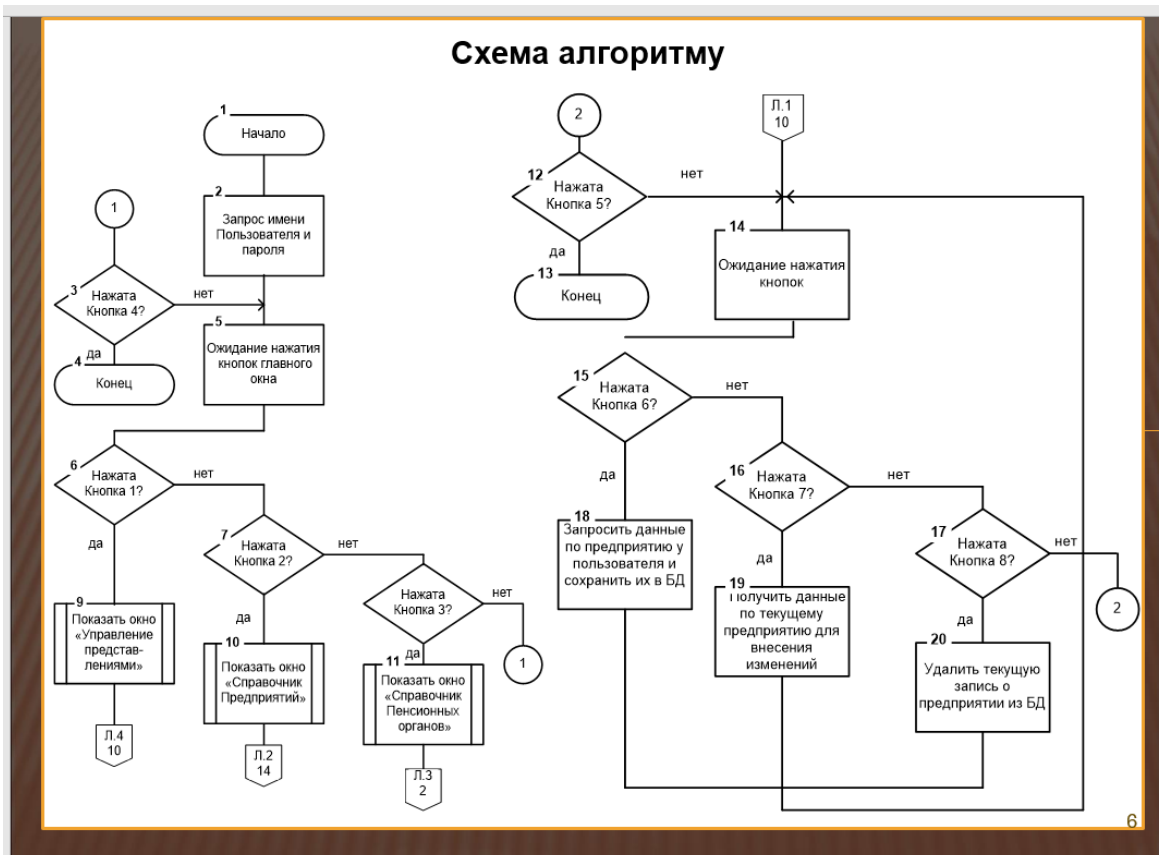


Рисунок С.6 – Слайд №6



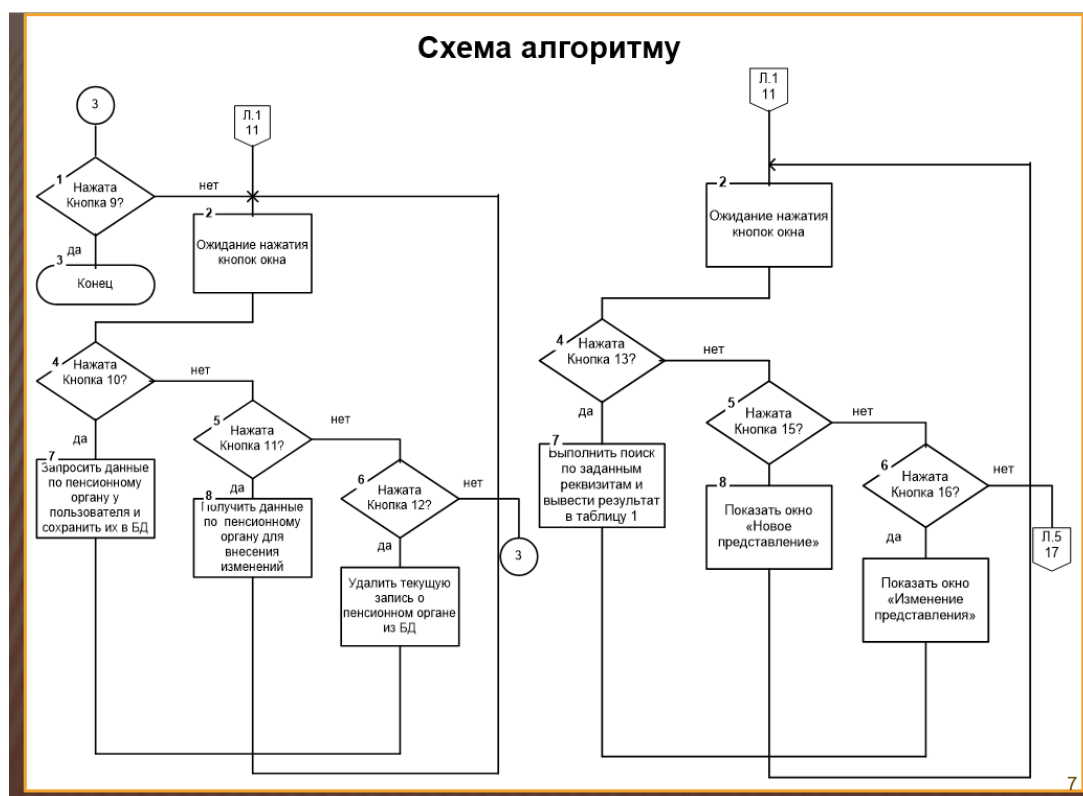


Рисунок С.7 – Слайд №7



Рисунок С.8 – Слайд №8

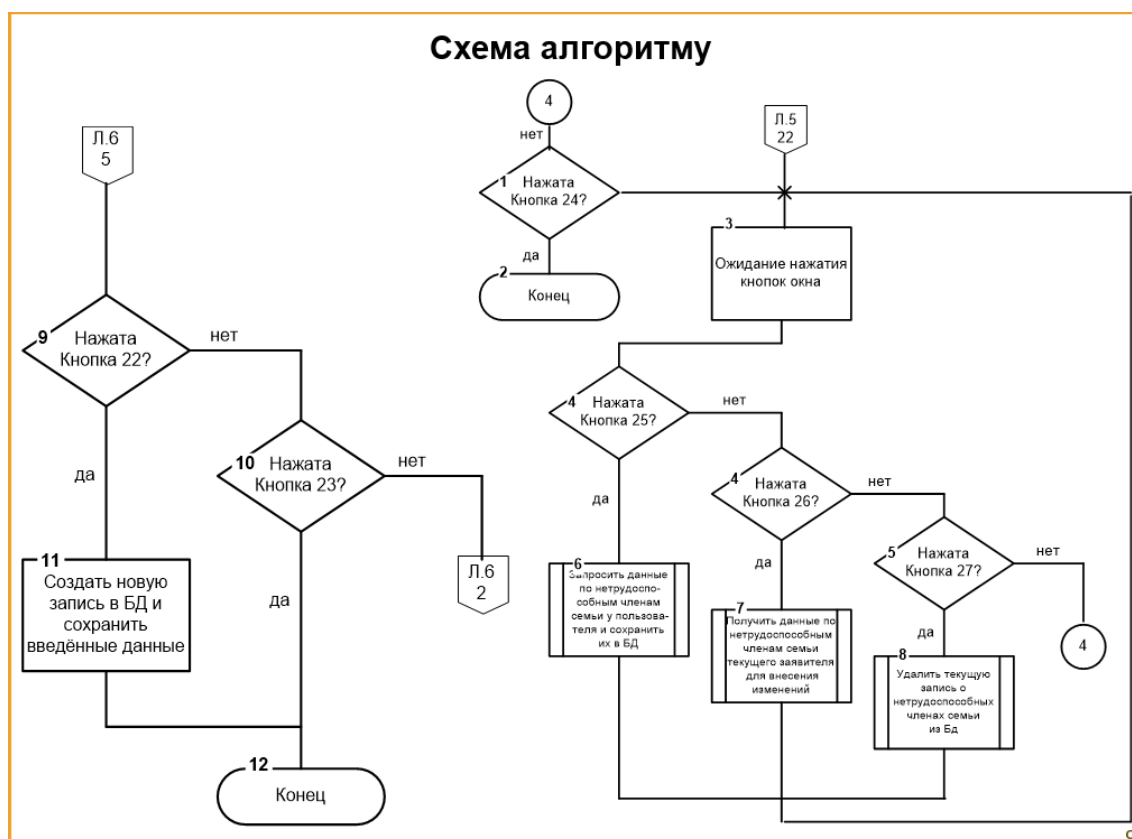


Рисунок С.9– Слайд №9

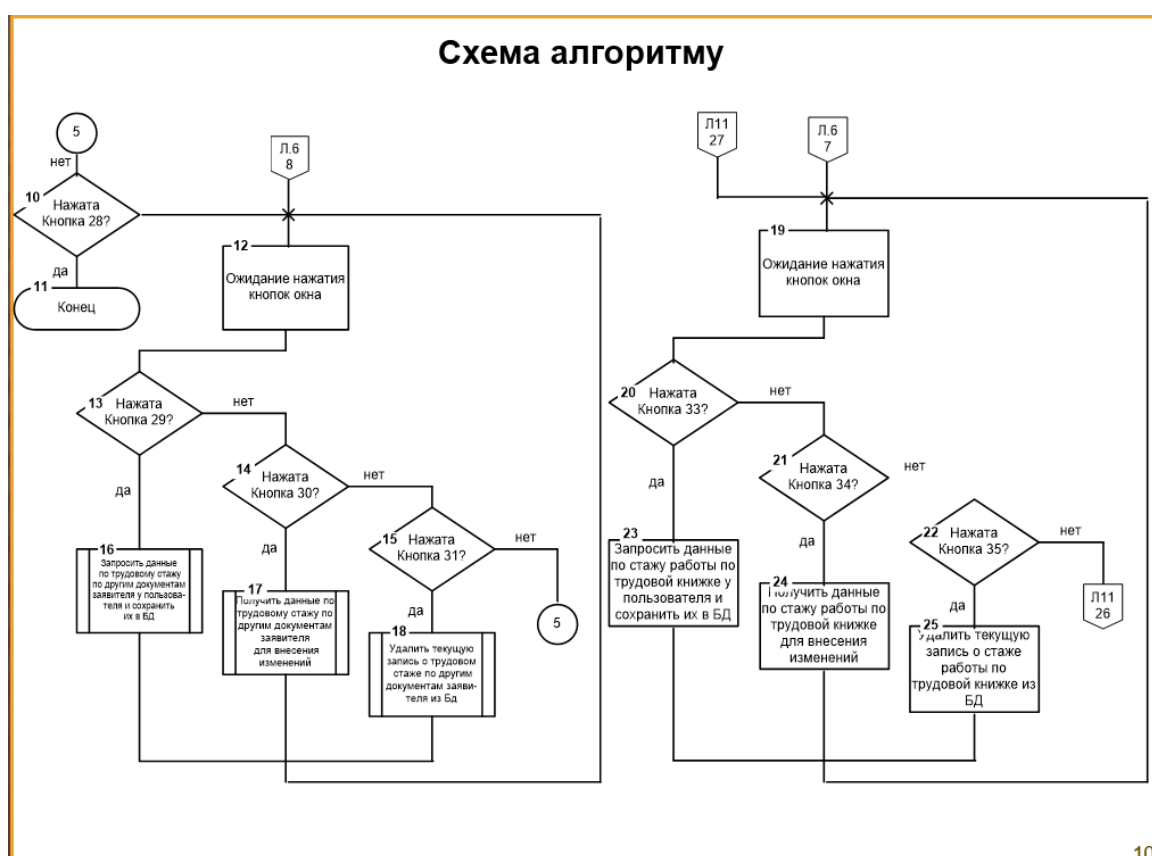


Рисунок С.10 – Слайд №10

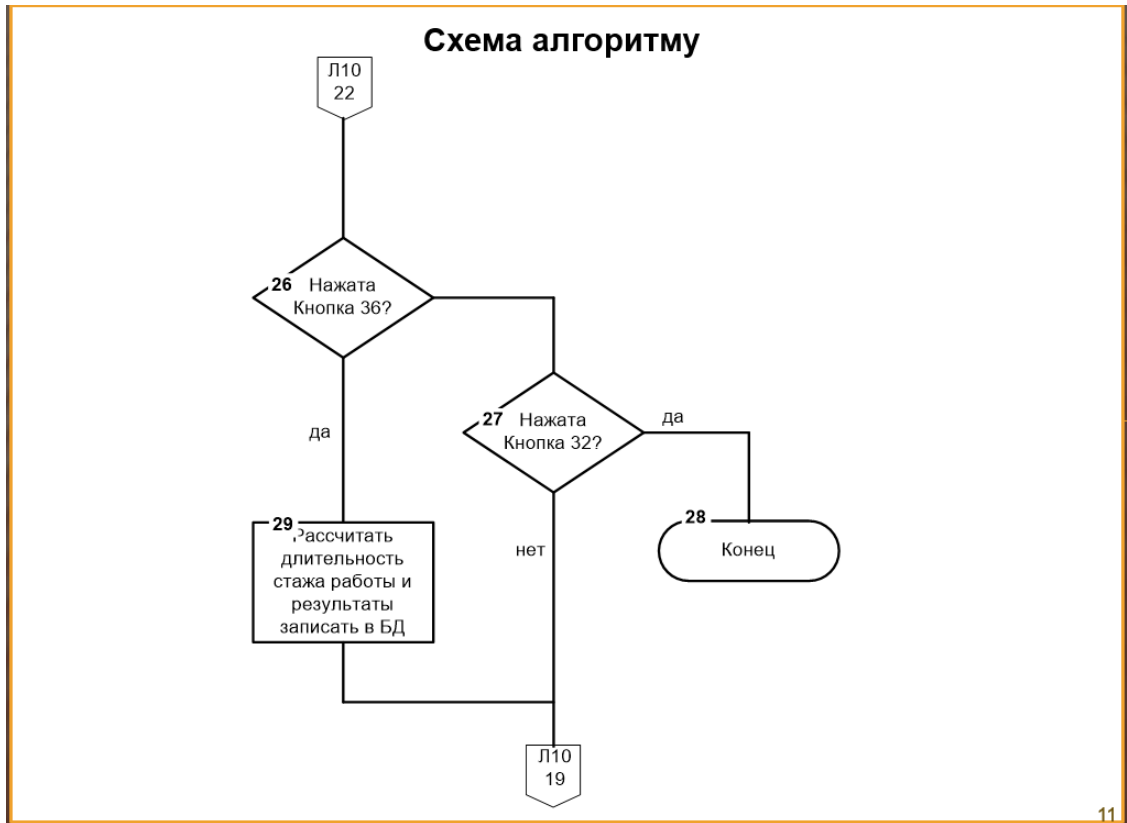


Рисунок С.11 – Слайд №11

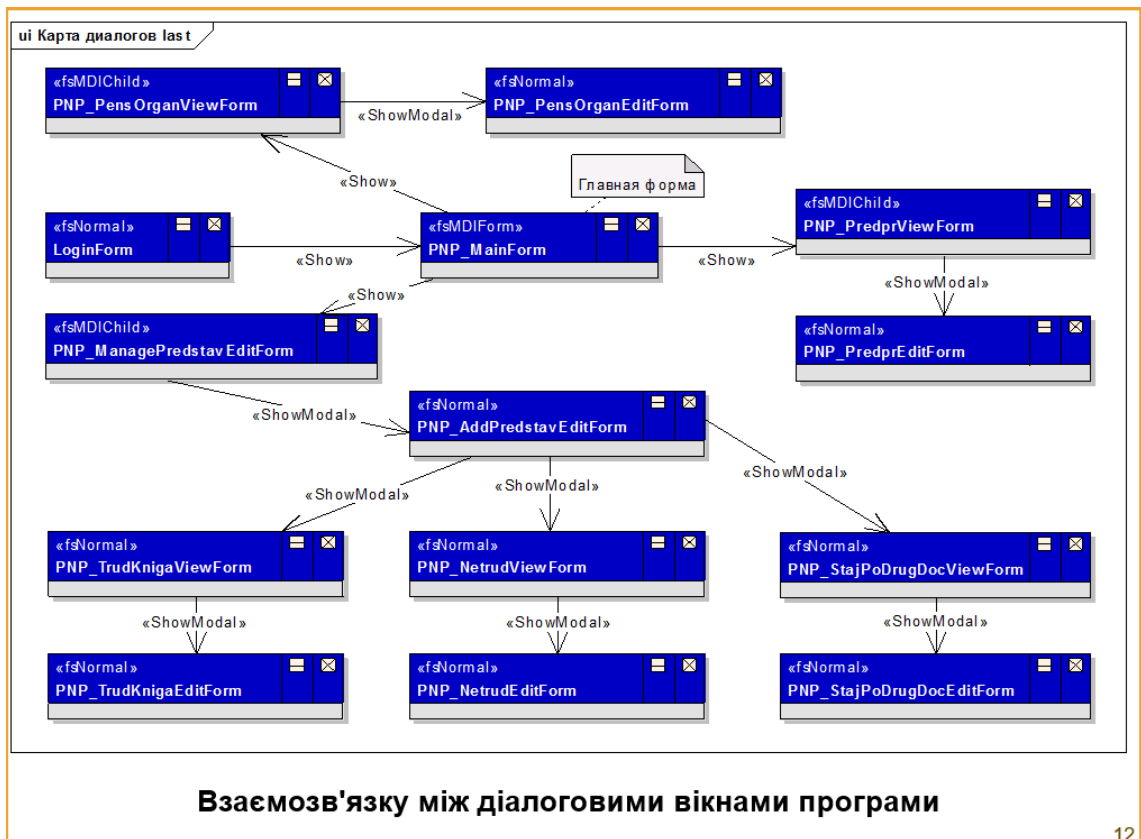
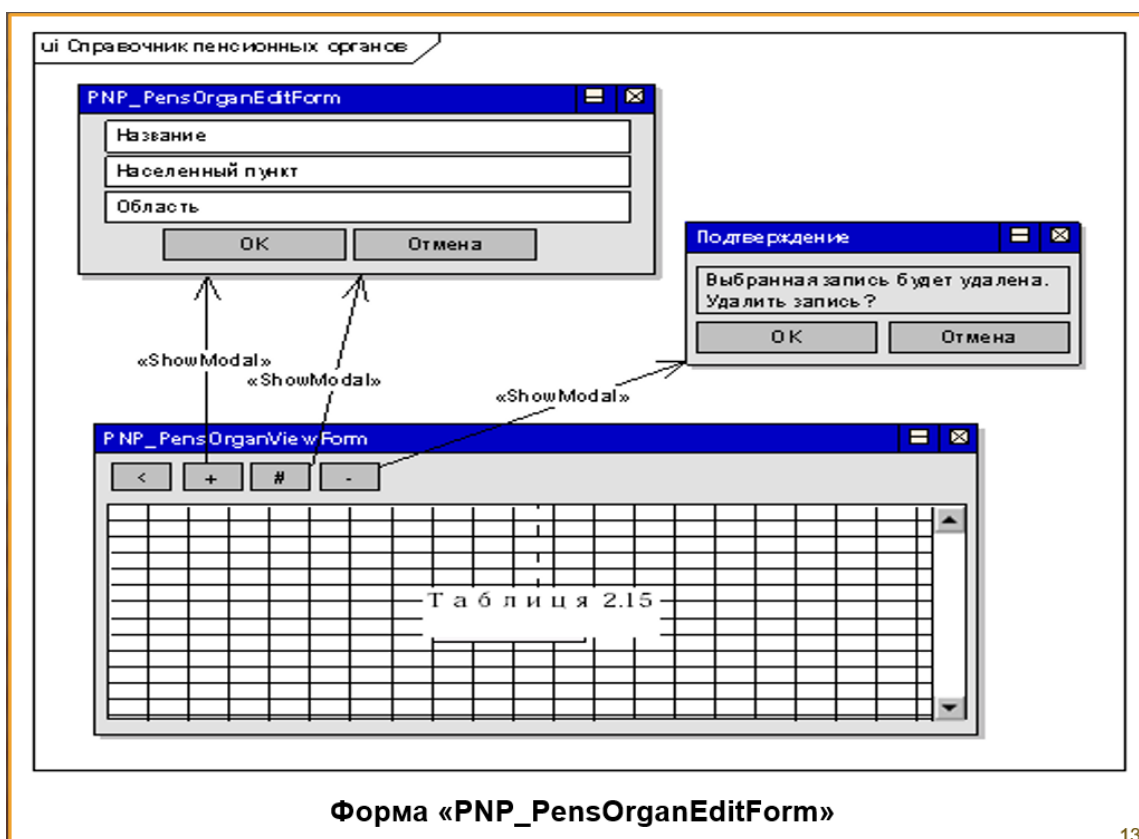


Рисунок С.12 – Слайд №12



13

Рисунок С.13 – Слайд №13

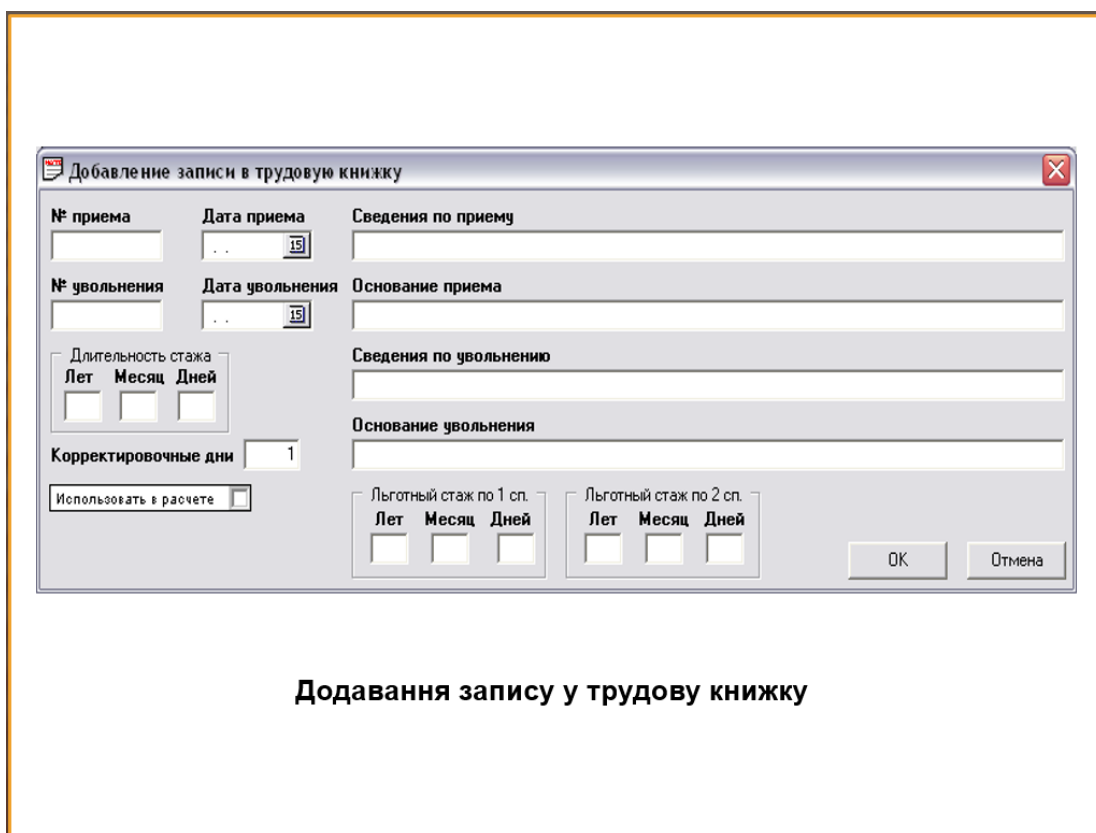
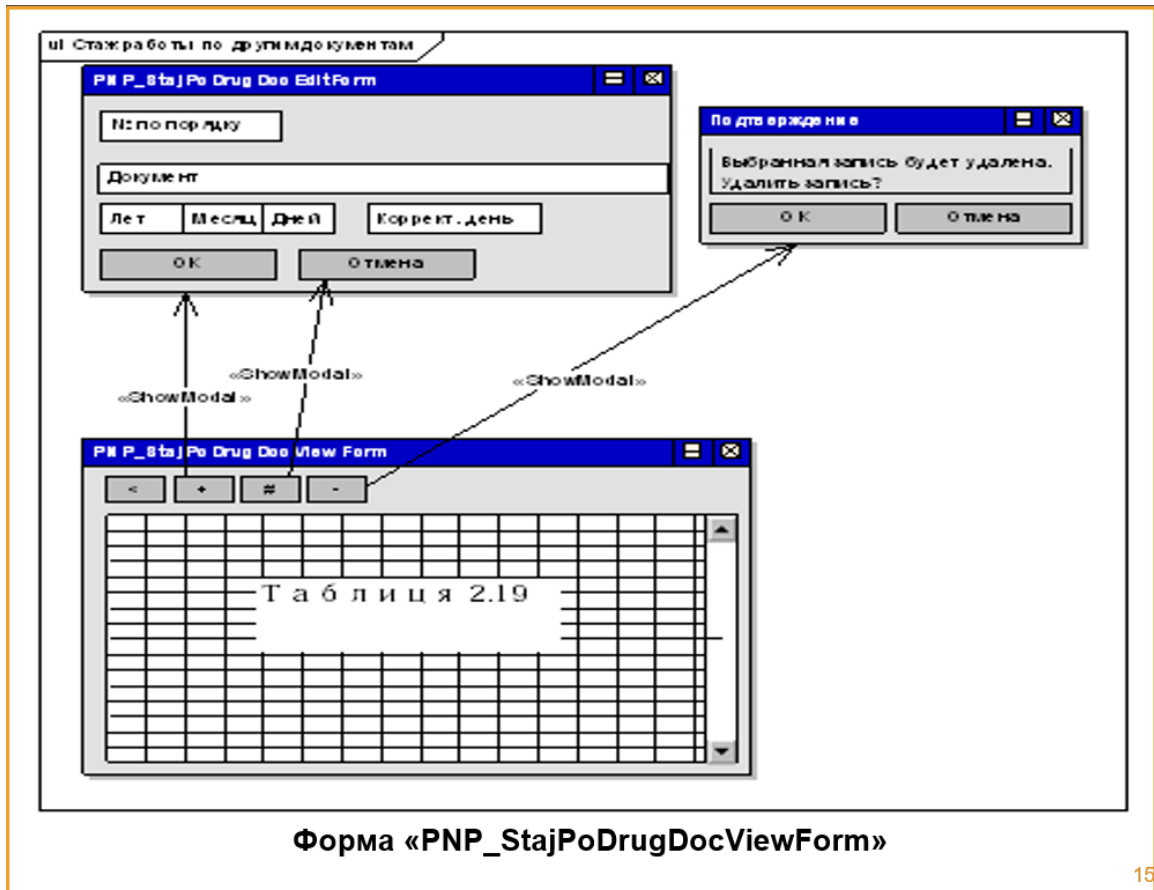


Рисунок С.14 – Слайд №14



15

Рисунок С.15 – Слайд №15

## Висновки

В даному дипломному проекті розроблено програмне забезпечення для розрахунку стажу роботи та нарахування пенсій. Дана програма значно полегшить роботу працівників відділу кадрів, а також скоротить час виконання операцій.

Зроблено аналіз операцій, які виконує працівник відділу кадрів до автоматизації та аналіз операцій, які буде виконувати робітник відділу кадрів після автоматизації. Сформовано технічне завдання.

Розроблено і описано програмне забезпечення, та наведені інструкції оператора.

Отримані результати розробки свідчать про її ефективність та доцільність впровадження та використання на підприємстві.

16

Рисунок С.16 – Слайд №16