

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ

Пояснювальна записка

до дипломної роботи

бакалавр

(освітньо-кваліфікаційний рівень)

**на тему «Розробка комп'ютерної системи організації кадрового обліку
підприємства»**

Виконав: студент 4 курсу, групи ІТз-141
напряму підготовки 6.040302 „Інформатика”
спеціальності 7.04030201 „Інформатика”

_____ Нехаєнко С.М.
(підпис)

Керівник,
доцент, к. ф.-м.н. _____ Ковальов Ю.Г.
(підпис)

Рецензент,
доцент, к.т.н. _____ Фесенко Т.М.
(підпис)

СЄВЕРОДОНЕЦЬК
2018 року

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ
КАФЕДРА ПРОГРАМУВАННЯ ТА МАТЕМАТИКИ
Освітньо-кваліфікаційний рівень бакалавр
Напрямок підготовки 6.040302 „Інформатика”

ЗАТВЕРДЖУЮ
Завідувач кафедри
ПМ,
д.т.н., доцент
_____Лифар
В.О.
«___» _____ 2018

р.

З А В Д А Н Н Я

НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Нехаенко Сергію Миколайовичу

1. Тема роботи Розробка комп'ютерної системи організації кадрового обліку підприємства

керівник роботи: доцент Ковальов Юрій Григорійович

затверджені наказом вищого навчального закладу від “_”_2018 року №__

2. Строк подання студентом роботи 03 червня 2018 р.

3. Вихідні дані до роботи:

3.1 Літературні джерела:

1. Половнев Н.М., Якимов А.М. Системы автоматизированной обработки учетной информации. - М.: Финансы и статистика, 1994. - 187 с.
2. Твердохлеб Н.Г., Татарчук Н.И., Сендзюк М.А. Организация машинной обработки экономической информации в промышленности. - К.: Вища школа, 1984. - 416с.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Вступ

4.2 Аналіз предметної галузі (огляд літератури), з висвітленням наступних питань:

4.3 Основна частина, в якій висвітлити:

Інформаційна модель об'єкту.

Програмну реалізацію моделі.

4.4 Висновки

4.5 Перелік використаних джерел

5. Перелік графічного матеріалу немає

6. Дата видачі завдання 7 лютого 2018 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломної роботи	Строк виконання етапів роботи	Примітка
1	Одержання завдання на виконання роботи	01.02.18	
2	Укладання і погодження з керівником плану і етапів виконання роботи	20.02.18	
3	Узагальнення даних літературних джерел, укладання розділу «Аналіз предметної галузі»	1.03.18	
4	Аналіз шляхів виконання завдання. Вибір і погодження з керівником оптимального шляху	20.03.18	
3	Проектування інфологічної моделі задачі що реалізується.	15.04.18	
5	Укладання та тестування програмного продукту	1.05.18	
6	Укладання, оформлення та погодження пояснювальної записки з керівником	25.05.18	
7	Здача готової пояснювальної записки на кафедру	03.06.18	
8	Укладання доповіді і презентації	10.06.18	

Студент
С.М.

(підпис)

Нехаєнко

Керівник роботи
Ю.Г.

(підпис)

Ковальов

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ
дипломної роботи студента гр. ІТ-141з Нехаєнко С.М.

Науковий керівник

Доцент, к.ф.-м.н.

Ковальов Ю.Г.

Оцінка наукового керівника: _____

Рецензент Фесенко Т.М., доцент каф. ПМ СНУ ім.В.Даля

ПБ, місто роботи, посада

Оцінка рецензента: _____

Кінцева оцінка за результатами захисту:

Голова ЕК

підпис

Лифар В.О.

РЕФЕРАТ

Текст – 81., рис – 10, табл. – 1, додатків – 1, літературних джерел – 10

У роботі обґрунтовано вибір теми дослідження, доведено її актуальність. Також наведена характеристика предметної області. У роботі наведені результати дослідження предметної області: комп'ютерна система організації кадрового обліку підприємства. Вказана інформаційна сукупність, використовувана у процесі розв'язання даної задачі.

Розроблено програму яка дозволяє робити облік співробітників підприємства.

Ключові слова: КОМП'ЮТЕРНА СИСТЕМА, АВТОМАТИЗОВАНИЙ ОБЛІК, BORLAND DELPHI 7 .

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД.....	9
1.1.Характеристика предметної області	9
1.2 Призначення і область застосування	12
1.3 Огляд програм автоматизації кадрового обліку	16
1.4 Перспективи розвитку світового ринку інформаційних систем обліку і контролю руху кадрів на підприємстві.....	25
РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ	27
2.1 Характеристика задачі.....	27
2.2. Основні вимоги до системи кадрового обліку підприємства	29
РОЗДІЛ 3 ОПИС ПРОГРАМИ.....	36
3.1 Вимоги до програми.....	36
3.2 Проектування інтерфейсу програми.....	37
3.3 Опис застосування програми.....	39
ВИСНОВКИ.....	45
СПИСОК ЛІТЕРАТУРИ.....	46
ДОДАТОК А.....	47

Актуальність досліджень. Функціонування будь-якої системи управління пов'язано з циркуляцією в ній інформації. Будь-який процес управління - перш за все, інформаційний процес, який передбачає виконання функцій збору, передачі, обробки, аналізу інформації та прийняття відповідних рішень. Забезпечення якісного інформаційного обслуговування користувачів шляхом видачі вірогідної, своєчасної та достатньої для прийняття управлінських рішень інформації у формі, яка є зручною для використання, є основним призначенням інформаційного забезпечення.

Задача обліку наявності та руху кадрів займає провідне місце в системі управління.

Успішне вирішення цієї задачі багато в чому залежить від раціональної організації інформаційного забезпечення, яке дозволяє вирішувати такі проблеми як: централізоване керування даними, інформаційна сумісність, гнучкість та ефективна актуалізація інформаційної бази.

При автоматизації за допомогою інформаційної системи зменшуються обсяги ручної праці, зменшується час на обробку та отримання необхідної інформації, зростає продуктивність роботи працівників відділу кадрів.

Об'єкт досліджень: Інформаційні процеси пов'язані з обліком наявності та руху кадрів.

Предмет досліджень: Побудова демонстраційної програми за допомогою будь-якого середовища візуального програмування, який був би доступний та легкий у використанні користувачу-початківцю.

Мета дослідження: Виявити найбільш зручне та доступне середовище візуального програмування, за допомогою якого створити демонстраційну програму.

Завдання дослідження:

а) скласти інформаційну модель;

б) розкрити алгоритми;

в) скласти програму;

Методологічна та теоретична основа дослідження: структура та методи побудови інформаційного забезпечення обліку співробітників підприємства

Методи дослідження: програмування за допомогою візуального середовища.

Практичне значення отриманих результатів. Розроблено програму яка дозволяє облік співробітників підприємства.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД

1.1 Характеристика предметної області

Успішність будь-якого бізнесу залежить, не в останню чергу, від кваліфікації персоналу підприємства, а від його вміння і бажання продуктивно працювати. Отже, питання продуманого управління обліку та контролю за рухом кадрів на підприємстві повинні займати одне з ключових місць у загальній стратегії розвитку підприємства. Важлива роль в оптимізації управління співробітниками будь-якого підприємства належить інформаційним системам обліку і контролю руху кадрів на підприємстві. У своїй діяльності компанії змушені наймати працівників, як на тимчасовій, так і постійній основі, вести документацію на кожного з них, підтримуючи її у вигляді, відповідному як внутрішнім, так і зовнішнім (кодексу законів про працю) вимог. Будь-яка інформаційна система надає всім зацікавленим особам (керівництву, менеджерам, самим працівникам, нарешті,

Раніше, використання інформаційних систем управління вважалося привілеєм великих підприємств. Тому масштаб і ціни цих систем були відповідними, а малим і середнім компаніям більшість продуктів були не по кишені. І хоча нині практично всі постачальники пропонують системи малого середнього класу, ринок все ще далекий від насичення. Тому перед керівництвом підприємств малого і середнього бізнесу встає вибір: купувати готову автоматизовану систему, або використовувати офісні кошти. Слід зазначити, що при придбанні готового програмного продукту в будь-якому випадку буде потрібна серйозна настройка програми під конкретні вимоги, а також обов'язкове подальше програмне супровід, і навчання персоналу, що виразиться в відчутних фінансових і тимчасових витратах,

З появою перших засобів автоматизації були розроблені і перші програми обліку та управління руху кадрів на підприємстві, число яких як в Україні, так і в інших країнах світу обчислюється зараз сотнями. Якщо говорити про Україну, то кожен поважаючий себе підприємство або організація, що мали власний відділ АСУ, ще в 1980-і рр. користувалися програмами обліку персоналу власної розробки. Ці програми спиралися на різні апаратні платформи (починаючи від мейнфреймів і закінчуючи ПК) і інструментальні засоби (починаючи від PL-1 і закінчуючи Clipper і FoxPro). З появою нових, більш досконалих, апаратних і інструментальних засобів, дані програми модифікувалися і розширювали свою функціональність (особливо швидко це відбувалося на Заході).

ERP-система(Англ. Enterprise Resource Planning System) - Це система планування ресурсів підприємства - корпоративна інформаційна система (КІС), призначена для автоматизації обліку і управління. Як правило, ERP-системи будуються за модульним принципом і в тій чи іншій мірі охоплюють всі ключові процеси діяльності компанії.

Історично концепція ERP стала розвитком простіших концепцій MRP (Material Requirement Planning - Планування матеріальних потреб) і MRP II (Manufacturing Resource Planning - Планування виробничих ресурсів). Використовуваний в ERP-системах програмний інструментарій дозволяє проводити виробниче планування, моделювати потік замовлень і оцінювати можливість їх реалізації в службах і підрозділах підприємства, пов'язуючи його зі збутом.

В основі ERP-систем лежить принцип створення єдиного сховища даних, що містить всю корпоративну бізнес-інформацію і забезпечує одночасний доступ до неї будь-якого необхідної кількості співробітників підприємства, наділених відповідними повноваженнями. Зміна даних проводиться через функції (функціональні можливості) системи. ERP-система складається з наступних елементів [2]:

- модель управління інформаційними потоками (ІП) на підприємстві;
- апаратно-технічна база і кошти комунікацій;

- СУБД, системне і забезпечує ПО;
- набір програмних продуктів, що автоматизують управління ІІ;
- регламент використання і розвитку програмних продуктів;
- ІТ-департамент і забезпечують служби;
- власне користувачі програмних продуктів

Відмінність ERP-системами від інших інформаційних систем, наприклад, такий як системою електронного документообігу (СЕД) в тому, що, як правило, в ERP документи не "ведуться", а "проводяться" - вже після того, як вони здійснять свій життєвий цикл, тобто будуть створені, обговорені, перевірені, узгоджені, затверджені та т. д. А СЕД здійснює підтримку такого життєвого циклу документів на підприємстві.

Класичні ERP-системи, на відміну від так званого "коробкового" програмного забезпечення, відносяться до категорії "важких" програмних продуктів, що вимагають досить тривалої настройки, для того щоб почати ними користуватися. Вибір ERP-системи, придбання та впровадження, як правило, вимагають ретельного планування в рамках тривалого проекту з участю партнерської компанії - постачальника або консультанта. Оскільки ERP-системи будуються за модульним принципом, замовник часто (принаймні, на ранній стадії таких проектів) купує не повний спектр модулів, а обмежений їх комплект. В ході впровадження проектна команда, як правило, протягом декількох місяців здійснює настройку поставляються модулів.

Застосування ERP системи дозволяє використовувати одну інтегровану програму замість декількох розрізнених. Єдина система може управляти обробкою, логістикою, дистрибуцією, запасами, доставкою, виставлянням рахунків-фактур і бухгалтерським обліком.

Існує хибна думка, що іноді ERP складно або неможливо адаптувати під документообіг компанії і її специфічні бізнес-процеси. Насправді, будь-якого впровадження ERP-системи передує етап опису бізнес-процесів компанії. По суті ERP-система являє собою віртуальну проекцію компанії.

Сучасні інформаційні системи обліку і контролю руху кадрів на підприємстві призначені для оптимізації роботи, в першу чергу, керівництва і персоналу кадрових служб підприємств (крім бухгалтерії і деяких інших підрозділів) і грають велику роль в підвищенні продуктивності їхньої праці. Зокрема, менеджери по персоналу за допомогою таких систем позбавляються від виконання рутинних операцій при роботі з кадрами, підготовки та обліку наказів (існують оцінки, що тільки на роботу з документацією по персоналу кадровики витрачають до 60% свого робочого часу).

Автоматизоване зберігання і обробка повної кадрової інформації також дозволяє ефективно здійснювати підбір і переміщення співробітників.

Крім того, автоматизований розрахунок заробітної плати з урахуванням інформації про позиції штатного розкладу, відпустках, лікарняних, відрядженнях, пільги і стягнення дає можливість точно і оперативно нараховувати зарплату, формувати звіти, відносити витрати на собівартість. І це лише деякі з функцій сучасних інформаційних систем обліку і контролю руху кадрів на підприємстві.

1.2 Призначення і область застосування

Інформаційна система обліку і контролю руху кадрів на підприємстві являє собою місце користувача-фахівця тієї чи іншої професії, обладнане засобами, необхідними для автоматизації виконання ним певних функцій. В основному, інформаційна система, орієнтовані на користувача, який не має спеціальної підготовки по використанню обчислювальної техніки. Тому, що розробляється програмний продукт має зрозумілий і доступний інтерфейс програми. Основним призначенням інформаційною системою обліку і контролю руху кадрів на підприємстві можна вважати децентралізована обробку інформації на робочих місцях.

Для реалізації ідеї розподіленого управління треба було створення на кожному рівні управління і для кожної предметної області робочих місць базу професійних персональних комп'ютерів. Наприклад, у сфері економіки на таких інформаційних

системах обліку та контролю руху кадрів на підприємстві можна здійснювати планування, моделювання, оптимізацію процесів, прийняття рішень в різних інформаційних системах і для різних сполучень завдань. Для кожного об'єкта управління необхідно передбачати інформаційну систему, що відповідатимуть їхнім значенням.

Однак принципи створення будь-яких інформаційних систем повинні бути загальними [3]:

- системність.
- гнучкість.
- стійкість.
- ефективність.

Системність. Інформаційні системи слід розглядати як системи, структура яких визначається функціональним призначенням.

Гнучкість. Інформаційна система пристосована до можливих перебудов, завдяки модульності побудови всіх підсистем і стандартизації їх елементів.

Стійкість. Принцип полягає в тому, що інформаційна система повинна виконувати основні функції незалежно від впливу на неї внутрішніх і зовнішніх факторів, що обурюють. Це означає, що неполадки в окремих її частинах повинні бути легко устраними, а працездатність системи швидко відновлючі.

Ефективність. Інформаційну систему слід розглядати як інтегральний показник рівня реалізації наведених вище принципів, віднесеного до витрат на створення і експлуатацію системи.

Функціонування інформаційної системи може дати бажаний ефект за умови правильного розподілу функцій і навантаження між людиною і машинними засобами обробки інформації, ядром якої є комп'ютер.

Створення такого "гібридного" інтелекту в даний час є проблемою. Однак реалізація цього підходу при розробці та функціонуванні може дати відчутні результати - інформаційна система обліку і контролю руху кадрів на підприємстві стане засобом підвищення не тільки продуктивності праці і ефективності управління, а й соціальної комфортності фахівців. При цьому людина в системі повинен залишатися провідною ланкою.

На виробничих підприємствах інформаційні системи є важливою структурною складовою управління персоналом як персональне засіб планування, управління, обробки даних і прийняття рішень. Інформаційна система обліку і контролю руху кадрів на підприємстві - це спеціалізована програма, яка має набір технічних засобів і програмного забезпечення, орієнтованого на конкретного фахівця - адміністратора з управління персоналом.

У той же час до будь-якої інформаційної системи можна висунути і ряд загальних вимог, які повинні забезпечуватися при його створенні, а саме [3]:

- безпосереднє наявність засобів обробки інформації;
- можливість роботи в діалоговому (інтерактивному) режимі;
- виконання основних вимог ергономіки: раціональний розподіл функцій між оператором, елементами комплексу інформаційних систем і навколишнім середовищем, створення комфортних умов роботи, зручність конструкцій інформаційних систем, облік психологічних факторів людини-оператора, привабливість форм і кольору елементів ІС і ін .;
- досить висока продуктивність і надійність ПК, що працює в інформаційній системі;
- адекватне характеру вирішуваних завдань програмне забезпечення;
- максимальний ступінь автоматизації рутинних процесів;
- оптимальні умови для самообслуговування фахівців як операторів інформаційних систем;
- інші чинники, що забезпечують максимальну комфортність і задоволеність фахівця використанням інформаційних систем, як робочий інструмент.

Структура інформаційної системи обліку та контролю руху кадрів на підприємстві

включає сукупність підсистем - технічної, інформаційної, програмної та організаційної.

До інформаційної підсистеми відносяться масиви інформації, що зберігається в локальних базах даних, як правило, на дискових накопичувачах. Сюди ж відноситься і системи управління базами даних.

Програмне забезпечення включає операційні системи, сервісні програми, стандартні програми користувачів і пакети прикладних програм, виконані за модульним принципом і орієнтовані на вирішення певного класу задач, обумовленого призначенням АІС. У міру необхідності в програмне забезпечення включаються також модулі для роботи з графічною інформацією.

Організаційне забезпечення інформаційної системи обліку та контролю руху кадрів на підприємстві має на меті організацію їх функціонування, розвитку, підготовки кадрів.

Якщо пристрій інформаційної системи обліку та контролю руху кадрів на підприємстві досить складно, а користувач не має спеціальних навичок, можливе застосування спеціальних навчальних засобів, які дозволяють поступово ввести користувача в середу його основного автоматизованого робочого місця. При реалізації функцій інформаційної системи (тобто власне його функціонування) необхідні методики визначення мети поточної діяльності, інформаційної потреби, всіляких сценаріїв для опису процесів її реалізації.

Методика проектування інформаційної системи не може не бути пов'язаною з методикою його функціонування, так як функціонування розвиненого інформаційного продукту передбачає можливість його розвитку самими користувачами. Мовні засоби інформаційної системи є реалізацією методичних засобів з точки зору кінцевого користувача, а програмні реалізують мовні засоби користувача і дають можливість кінцевому користувачеві виконувати всі необхідні дії.

Робота в інформаційній системі обліку і контролю руху кадрів на підприємстві пов'язана з накопиченням великої кількості інформації про особисті дані співробітників.

Традиційно інформація зберігатися на паперових носіях.

При цьому важко здійснити швидкий відбір потрібних даних при прийомі на роботу, відхід у відпустку, звільнення співробітника.

Дана розробка здатна істотно полегшити працю працівника відділу кадрів у виконанні його повсякденних завдань.

1.3 Огляд програм автоматизації кадрового обліку

Програми кадрового обліку потрібні, перш за все, фахівцям з кадрового діловодства. Такі системи служать для вирішення двох завдань: автоматизації оформлення документів і розрахунку заробітної плати.

Особиста справа - сукупність документів, що містять відомості про працівника і його трудової діяльності. Особиста справа оформляється після видання наказу про зарахування на роботу і містить:

- особовий листок з обліку кадрів;
- автобіографію;
- копії документів про освіту;
- копію документа про затвердження на посаді;
- характеристики або рекомендаційні листи;
- заяву про прийом на роботу або трудовий договір;
- копію наказу про прийом на роботу;
- список наукових праць і винаходів.

Крім того, багато систем також дозволяють зберігати інформацію про сімейний стан співробітника і військовому обліку.

Інформація про співробітника буде неповною, якщо не зберігати дані про історію його роботи, про перекладах на нові посади, заохочень та стягнень. Зазвичай можливість зберігання даних такого роду називають можливістю відстеження руху співробітників, зберігання організаційної структури та штатного розкладу.

Облік робочого часу співробітників дозволяє реєструвати дії співробітника, автоматично виставляє порушення, вважає фактично відпрацьований час, становить особисті табелі, відображає оперативну інформацію. Також розробники часто додають можливість реєстрації в системі відряджень, лікарняних і відпусток.

Після того, як в систему занесені всі перераховані вище дані, залишилося ввести тарифні ставки і інформацію про податки, щоб було достатньо відомостей для автоматизації розрахунку заробітної плати.

Ну і, звичайно, набагато корисніше буде система, яка вміє виводити все зберігається в ній інформацію у вигляді стандартних форм, сформованих за правилами сучасного діловодства.

Оцінка персоналу.

Якщо системи кадрового обліку найбільше потрібні співробітнику відділу кадрів і бухгалтеру, то системи оцінки персоналу дуже корисні також і менеджерам. Для найбільш розумного і ефективного використання співробітників необхідно правильно оцінити його. Різні системи, методи і методика оцінки персоналу дозволяють виявити і розкрити потенціал кожного співробітника, і направити цей потенціал на реалізацію стратегічних цілей компанії. Оцінка завжди має на увазі наявність критеріїв оцінки (шкали, компетенції, КРІ) і ідеального «портрета» співробітника, описаного в термінах критеріїв. З точки зору побудови системи можна виділити чотири об'єкти оцінки (то, що оцінюємо) і чотири групи методик оцінки, від яких залежить організація процесу оцінки: Об'єкти оцінки:

- психологічні характеристики особистості;
- знання, вміння і навички співробітника;
- поведінка (цінності, відносини, думки);
- ефективність праці.

Групи методик оцінки:

- атестація;
- тестування і вимір, тренажери;
- інтерв'ювання, опитування;
- ділові ігри.

Об'єкти оцінки часто називають компетенціями. З технічної точки зору оцінка всіх компетенцій здійснюється однаково, тому головне при виборі програми - реалізація тих чи інших методик. "Атестація" - визначення рівня відповідності. Атестація є однією з найбільш поширених форм оцінки роботи персоналу, а точніше визначення рівня відповідності працівника займаній посаді і (або) місця на яке він претендує.

Системи тестування зазвичай включають систему зберігання результатів тестування, кошти проведення тестування, набір готових тестів і конструктор нових. Інтерв'ювання та опитування зазвичай дають більш повне уявлення про працівника, але їх результати важче формально оцінити і зберегти в структурованому вигляді для подальшого пошуку. Повернемося до основної мети оцінки - підбору кадрів і розстановці їх на відповідні посади. Вирішення цього завдання найпростіше починати з пошуку співробітників, що володіють певними компетенціями і подальшого аналізу їх особистих якостей і реального досвіду роботи. Так як подібні системи знаходять своє місце на комп'ютерах керівництва, то необхідними функціями є: зберігання історії навчання співробітника, планування потреб в нових співробітниках, контроль над адаптацією недавно прийнятих працівників в колективі. Кадрове агентство Системи для кадрових агентств найчастіше абсолютно окремі програмні продукти. Але, все ж, у них є ряд функцій, які зближують їх з іншими кадровими системами: зберігання особистої інформації про працівників і, найчастіше, їх компетенцій. У той же час, є ряд специфічних функцій: зберігання інформації про роботодавців, вакансії, резюме у вільній і структурованій формі, розширені можливості пошуку.

Для підвищення ефективності роботи важливі функції роботи з інтернетом: пошук і розміщення вакансій, імпорт резюме з різних систем. Деякі системи для створення більш повної картини дозволяють зберігати інформацію про співробітників після працевлаштування, записувати відгуки від роботодавців і використовувати ці відомості при повторному зверненні цих людей.

1С: Зарплата і управління персоналом 8.

Персонал - це найбільш цінний ресурс будь-якої компанії. І як будь-який ресурс він вимагає обліку і ефективного управління. Від того, як збудовано управління персоналом, багато в чому залежить ефективність роботи підприємства в цілому. Кваліфіковані, ініціативні й лояльні співробітники здатні істотно підвищити якість роботи і конкурентоспроможність будь-якої компанії.

Управління даними про сотні й тисячі працівників, проведення заходів з підбору і навчання персоналу, оцінка кваліфікації виробничого і управлінського складу є трудомісткими процесами. Використання спеціалізованих програмних продуктів дає компанії можливість скоротити часові витрати на обробку великої кількості інформації і аналіз даних, дозволяє керівництву ефективно планувати і здійснювати кадрову політику.

Для управління людськими ресурсами підприємства фірмою "1С" розробтан сучасний інструмент автоматизації задач управління персоналом, ведення кадрового учета и расчетов в соответствии с требованиями законодательства для коммерческих предприятий различного масштаба – 1С:Зарплата и управление персоналом 8.



Рисунок 1.1. – Функциональные возможности 1С:Зарплата и управление персоналом

В 1С:Зарплата и управление персоналом 8 поддерживаются все основные процессы управления персоналом, а также процессы кадрового учета, расчета зарплаты, исчисления налогов, формирования отчетов и справок в государственные органы и социальные фонды, планирование расходов на оплату труда. Учтены требования законодательства, реальная практика работы предприятий и перспективные мировые тенденции развития подходов к управлению персоналом.

Зручні і гнучкі механізми настройки звітів дозволяють отримувати повну і достовірну інформацію в самих різних аналітичних розрізах для різних категорій користувачів: керівництва, служби управління персоналом, кадрової служби та інших.

З огляду на відмінності в потребах компаній різного рівня, фірма "1С" пропонує три програмні продукти з різним функціональним наповненням і складністю:

1С: Зарплата і управління персоналом 8. Базова версія - продукт для невеликої організації, що дозволяє на одному робочому місці в повному обсязі автоматизувати ведення кадрового обліку, розрахунку заробітної плати та обчислення необхідних податків і внесків відповідно до вимог законодавства.

1С: Зарплата і управління персоналом 8 ПРОФ дозволяє не тільки вести кадровий облік і розрахунок заробітної плати в компаніях, що мають складну юридичну структуру, а й автоматизувати базові функції управління персоналом (підбір, навчання, мотивація).

1С: Зарплата і управління персоналом 8 КОРП- комплексне рішення для автоматизації всіх завдань управління персоналом на середніх і великих підприємствах, для яких ефективно управління людськими ресурсами є необхідною умовою успішної роботи на ринку. Продукт дозволяє вирішувати завдання реалізації кадрової політики підприємства та всебічної оцінки персоналу на сучасному рівні, правильно і швидко обробляти інформацію про стан персоналу підприємства і давати на її основі якісні та змістовні висновки про здібності працівників, планувати навчання, розвиток та кар'єру, приймати обґрунтовані управлінські рішення.

Програма обліку робочого часу і контролю роботи співробітників CS Polibase Bio призначена для обліку робочого часу, обсягів і термінів виконуваної роботи і контролю роботи співробітників.

Програма обліку робочого часу і контролю роботи співробітників пов'язана з розкладом роботи співробітників і журналом табельного обліку, що дозволяє відстежувати запізнення і позаурочні години роботи, а також розраховувати погодинну і відрядну складові заробітної плати з об'єктивних фактичних даних.

Функциональные возможности	Базовая	ПРОФ	КОРП
Учет кадров и персонализированный учет	●	●	●
Расчет и учет заработной платы	●	●	●
Выплаты и депонирование	●	●	●
Исчисление налогов и страховых взносов	●	●	●
Подготовка регламентированной отчетности	●	●	●
Учет деятельности нескольких организаций	○	●	●
Возможность конфигурирования прикладного решения	○	●	●
Поддержка клиент-серверного варианта работы	○	●	●
Работа с распределенной информационной базой	○	●	●
Набор персонала с поиском кандидатов в интернете	○	○	●
Грейды и KPI	○	○	●
Социальные льготы и компенсации	○	○	●
Адаптация, обучение и развитие персонала	○	○	●
Кадровый резерв и управление талантами	○	○	●
Учет расходов по направлениям и проектам	○	○	●
Охрана труда, допуски, медосмотры, инструктажи	○	○	●
Удаленный доступ для руководителей и сотрудников	○	○	●

Рисунок 1.2 – Функціональні можливості 1С: Зарплата і управління персоналом 8. Базова версія

Реєстрація та облік робочого часу може здійснюватися за допомогою сканера відбитків пальців (біометричний спосіб) і без допомоги сканера відбитків пальців (традиційний спосіб), а також за допомогою пластикових карт із магнітною смугою, штрих-кодом або безконтактних пластикових карт. На відміну від інших способів обліку і контролю робочого часу біометричний спосіб виключає помилки і обман з боку співробітників при реєстрації в журналі парафій і відходів.

Система обліку і контролю роботи співробітників використовує спеціальним чином організовані стандарти і регламенти робіт, що дозволяє враховувати і контролювати керівництвом підприємства правильність і своєчасність виконання робіт.

Система може поставлятися окремим модулем або в складі системи комплексної автоматизації підприємства CS Polibase.

- Облік співробітників і робочого часу;
- Повноцінна база даних про працівників підприємства;
- Ведення штатного розкладу роботи співробітників;
- Прив'язка позицій штатного розкладу до підрозділів;
- Ведення погодинного графіка роботи;
- Облік парафій і відходів співробітників на робочі місця;
- облік робочого часу по співробітниках, підрозділам і компанії в цілому;
- Табель роботи співробітників;

- Ідентифікація співробітників і облік робочого часу;
- Ідентифікація співробітників на пункті реєстрації за відбитками пальців;
- Реєстрація часу приходу на роботу і звільнення з роботи;
- Розрахунок фактичного часу перебування співробітника на роботі.

Можливість прикріплення до контактів, завданням і діям будь-яких документів і зовнішніх файлів будь-якого формату (текст, електронні таблиці, електронні листи, файли PDF, зображення, звук та ін.) І їх перегляд та редагування у відповідному цьому формату редакторі (MS Word, MS Excel, MS Office Outlook, Acrobat Reader, редактор зображень, Media Player і ін.) викликом безпосередньо з картки контакту, завдання, дії.

Особливості графічного інтерфейсу системи CS Polibase Control

В системі використовується єдина форма графічного інтерфейсу, що істотно полегшує розуміння системи і навчання персоналу

Первинний графічний інтерфейс заснований на списках, а не на формах. Списки дозволяють оперувати на множинах даних і представляти результати комплексних запитів. Форми даних використовуються при редагуванні об'єктів обліку

Єдині правила роботи зі списками (додавання, зміна, видалення записів, пряма і зворотна сортування по будь-яким стовпцями, множинне виділення записів)

Оригінальний (який не має аналогів в інших системах), зручний і наочний механізм завдання будь-яких комплексних запитів з використанням уніфікованих навігаторів пошуку, що дозволяють задавати широкий вибір комбінацій критеріїв пошуку в будь-якому поєднанні

Уніфіковані блоки статистичної обробки інформації та підготовки звітів, розроблені за єдиними правилами і мають типові інтерфейси, дозволяють отримувати зведені і детальні звіти в будь-якому поєднанні критеріїв запиту за всіма основними показниками роботи підприємства

Використання кольорової палітри для виділення даних в залежності від контексту завдання

Наявність внутрішнього векторного і текстового редакторів документів і використання векторної графіки (наприклад, для представлення структури підприємства)

Додаткові можливості

Можливість організації кількох пунктів біометричної реєстрації співробітників в одній локальній мережі

Можливість організації єдиної системи обліку робочого часу для підприємства, що має територіально віддалені підрозділи, за допомогою системи віддалених обмінів

Можливість реєстрації будь-яких інших дій співробітників тільки по біометричного вирішенню

Експорт даних у зовнішніх форматах (MS Excel, текст, HTML).

E-Staff Рекрутер.

Програмний продукт E-Staff Рекрутер призначений для рекрутингових агентств, а також для кадрових служб підприємств, які здійснюють підбір співробітників. E-Staff Рекрутер - система повного циклу, яка автоматизує більшість рутинних операцій в рекрутингу.

E-Staff Рекрутер може використовуватися як комплексна програма для HR, що займаються підбором персоналу, так і програма для кадрових агентств. В обох випадках E-Staff Рекрутер дозволяє автоматизувати більшість операцій, пов'язаних з підбором персоналу, включаючи обробку резюме, ведення бази кандидатів, управління вакансіями, роботу з замовниками, отримання аналітичних даних і т.д.

1.4 Перспективи розвитку світового ринку інформаційних систем обліку і контролю руху кадрів на підприємстві

У сучасній висококонтурному бізнес-середовищі різко зросла потреба підприємств у висококваліфікованих співробітників. Посилився попит керівництва і кадрових служб підприємств на більш спеціалізовані дані про персонал, які допомагають

краще спланувати майбутні потреби підприємств в персоналі і реалізовувати свої плани шляхом продуманого рекрутингу, призначення і збереження в штаті талановитих і висококваліфікованих співробітників. Саме тому сегмент додатків обліку і контролю руху кадрів на підприємстві є зараз одним з найбільш швидко розвиваються на світовому ринку бізнес-додатків. За прогнозом IDC (звіт "2009: Додати A Workforce Management Odyssey, Workforce Management Applications Market Forecast and Analysis, 2005-2009" 2005 р. до більш ніж \$ 4 млрд. в 2009 р. (С щорічними темпами зростання в 29%). Для порівняння, щорічні темпи зростання всього ринку бізнес-додатків складуть лише 15%. Частка програм обліку і контролю руху кадрів на підприємстві на світовому ринку бізнес-додатків збільшиться з 1,8% в 2012р. до 3,4% в 2016 р. (Звіт "Workforce Management Applications in 2017: Додати Overview and Vendor Strategies"). На думку IDC, до складу додатків обліку і контролю руху кадрів на підприємстві входять транзакціональні і аналітичні програмні засоби, які повинні допомогти кадровим службам в наймі, призначення та збереженні висококваліфікованих співробітників [4].

Розробники додатків з обліку і контролю руху кадрів на підприємстві в даний час прагнуть до якнайшвидшої їх адаптації у замовників. На думку аналітиків IDC, швидкість адаптації WFM-додатків тепер більш ніж раніше, залежить від того, які ресурси їх розробники інвестують в навчання користувачів і передачу їм відповідних знань і навичок до і після продажу ПО. Перш за все, виробникам WFM-додатків необхідно переконати споживачів у вигоді їх використання.

У загальному контексті, Workforce Management (WFM) - це методологія планування робочого часу співробітників компанії, яка є складовою концепції управління персоналом (HRM).

В контексті інформаційних технологій, WFM є рішення для автоматизації процесу управління робочим часом персоналу. Найбільшого поширення WFM-рішення отримали в Call Center / Contact Center, де основною метою їх використання є підвищення продуктивності за рахунок [2]:

- складання оптимального розкладу роботи операторів відповідно до рівня їх кваліфікації, напрямком спеціалізації, а так само побажаннями співробітників;
- ефективного планування чисельності персоналу на підставі аналізу роботи за попередні періоди часу;
- моніторингу роботи операторів в режимі реального часу;
- створення комфортного робочого середовища для операторів і т.д.

Існують також оцінки аналітиків, що в нових WFM-додатках будуть реалізовані можливості управління витратами на персонал, обліку робочого часу персоналу і управління його діяльністю. Пізніше будуть також адаптовані додатки з управління навичками персоналу, управління наймом групи співробітників і ін.

РОЗДІЛ 2 ПОСТАНОВКА ЗАДАЧІ

В бакалаврській роботі проектується інформаційна система для розв'язку задачі по обліку наявності та руху кадрів в організації.

Пропонується автоматизувати робоче місце працівника відділу кадрів.

Основною метою створення комп'ютерної системи організації кадрового обліку підприємства є удосконалення обліку наявності та руху кадрів на основі автоматизації виконуваних функцій, реалізація елементів безпаперової технології при його веденні, оперативне формування у базі даних інформації про наявність, рух та звільнення працівників організації, які використовуються у відділі праці та заробітної плати (ВПЗ), бухгалтерії та іншими службами при виконанні функцій управління.

Розробка даного КС ведеться з можливістю об'єднання його, при необхідності, в мережу разом з КС працівників ВПЗ, де нараховується заробітна плата на основі даних отриманих з відділу кадрів та підрозділів; з КС начальників структурних підрозділів, де ведеться облік використання робочого часу.

2.1 Характеристика задачі [6].

Основним призначенням автоматизації обліку наявності та руху кадрів є отримання інформації, яка дозволяє здійснювати контроль та аналіз прийому працівників на роботу, переміщення їх на іншу посаду чи в інший підрозділ, звільнення працівників та співставлення цих відомостей з плановими даними по кількості персоналу з метою здійснення контролю за дотриманням встановленої чисельності працівників.

Доцільність автоматизації роботи працівника відділу кадрів по обліку кадрів на ПЕОМ визначається, передусім, скороченням затрат часу на обробку інформації, підвищенням якості обліку, контролю за наявністю та рухом кадрів, підвищенням оперативності отримання результатних даних, збільшенням точності та достовірності інформації, можливістю обробки

інформації, об'єм якої збільшується, а також організація комплексної системи автоматизованої обробки інформації працівників відділу кадрів, ВПЗ, бухгалтерії організації. При цьому управлінський персонал звільняється від рутинних, чисто технічних операцій по обробці інформації, приділяючи більше уваги аналізу отриманих результатів і розробці відповідних управлінських рішень по удосконаленню управління персоналом організації.

Основний зміст робіт, які виконує працівник відділу кадрів заключається у формуванні та веденні первинних документів, масивів з первинною інформацією, видачі необхідних документів та довідок бухгалтерії та ВПЗ, а також у веденні довідкових масивів.

Об'єктом, при управлінні яким розв'язується дана задача, є відділ кадрів обласного споживчого союзу.

Розв'язок задачі припиняється при виникненні збійних ситуацій в апаратному забезпеченні або при виході з ладу електропостачання ЕОМ, а також при несанкціонованому доступі і відсутності необхідної для розв'язання задачі інформації.

При розв'язанні на комп'ютерній системі організації кадрового обліку підприємства відбувається наступний розподіл функцій між персоналом і технічними засобами.

У функції працівника відділу кадрів входить:

- складання та візуальна перевірка первинних документів;
- введення та контроль даних;
- друк вихідних документів;
- підпис всіх вихідних і вхідних документів..

У функції технічних засобів входить:

- зберігання даних на машинних носіях і забезпечення їх збереження;
- контроль інформації;
- обробка даних;
- формування довідок;
- друк вихідних документів.

2.2 Основні вимоги до системи кадрового обліку підприємства

Оскільки прийняття управлінських рішень базується на обліковоаналітичних даних, виникає необхідність в організації ефективного обліку й аналізу. Забезпечення оперативності, доступності, зниження трудомісткості обробки інформації як прояв вдалої й ефективної організації обліковоаналітичних процесів на підприємствах ґрунтується на використанні сучасних інформаційних технологій. Комп'ютерно-комунікаційна техніка виступає одним з основних інструментів для фахівців з обліку й аналізу у вирішенні поставлених завдань, пов'язаних з підготовкою та наданням інформації різним групам користувачів.

Потреби різних організаційних рівнів і функціональних сфер менеджменту задовольняють п'ять головних типів інформаційних систем: діло-процесійні, офісно-автоматизаційні, управлінсько-інформаційні, системи підтримки рішення, системи підтримки виконання рішень.

Діло-процесійна система (англ. Transaction-Processing System) – комп'ютерна інформаційна система, що виконує щоденні поточні операції, потрібні для розвитку бізнесу, і забезпечує пряму підтримку на операційному рівні організації. Ця система – головне джерело інформації, яку використовують інші типи комп'ютерних систем організації.

Завдання офісно-автоматизаційної системи (англ. Office Automation System) – полегшити зв'язок і підвищити продуктивність праці менеджерів і офісних працівників через документи і передавальні процеси. Це система селекторної інформації, що може також містити електронний календар, телеконференцію, графіки та інше. Автоматизована інформаційна система є поліструктурним об'єктом і включає взаємопов'язану сукупність комплексу апаратно-програмних засобів для збирання, передачі і обробки інформації, відповідної інформаційної бази і персоналу, який бере участь у обробці, збиранні, передачі і використанні інформації. У сучасному комп'ютеризованому, взаємопов'язаному глобалізованому бізнес середовищі бухгалтер повинен вирішувати багато складних питань, таких що не були притаманні носіям цієї професії в минулому, наприклад, як захоплювати і записувати нові ділові операції і події, розвивати процеси з доданою вартістю

бізнесу та інформацією, створювати нові можливості для ланцюжків доданої вартості і ланцюжків поставок, поширювати корисні знання для широкого кола споживачів інформації, надавати послуги забезпечення по всьому спектру економічної діяльності, - це тільки деякі з найбільш цікавих тем, що представляють інтерес [4].

Дослідники виокремлюють наступні практичні наслідки впровадження інформаційних технологій у бухгалтерський облік: по-перше – це зростання продуктивності. Технології бухгалтерського обліку сприяють зростанню продуктивності завдяки внутрішнім математичним перевіркам та спеціальним характеристикам автоматичного обліку, таким що направлені на зменшення помилок введення. Це допомагає бухгалтерам зосередитися на перевірці точності та достовірності фінансової інформації, а не на встановленні рівноваги чисел задля визначення математичних помилок. Програмне забезпечення бухгалтерського обліку також дозволяє бухгалтерам добирати різноманітні рахунки та журнали до своїх фінансових звітів. Зокрема, введені дані можуть бути скореговані дуже швидко і легко, що допомагає звільнити час та направити його на вирішення інших бухгалтерських питань.

По-друге, це прискорення процесу звітності. З давніх часів фінансові відділи витрачали тижні на закриття фінансових періодів, особливо під час квартальних, або наприкінці року звітів. Довгострокові періоди закриття затягували процес публікації фінансових звітів, що дуже ускладнювало керівникам підприємств завдання огляду та оцінювання бізнес-операцій. Сучасні технології скоротили період закриття з тижнів до діб, дозволив бухгалтерам більше часу приділяти точності фінансової документації ще до того, коли вона попаде на перевірку керівникам. Скорочення терміну складання звітів також допомагає керівництву швидше реагувати на економічні виклики, що, безумовно, є дуже потужним інструментом в сучасному конкурентному бізнес-середовищі. Великим компаніям, таким що мають декілька регіональних представництв, можливість швидко звітувати дуже спрощує підготовку заключного корпоративного звіту.

По-третє, це – централізований облік. Сьогодні існує багато програм з бухгалтерського обліку на ринку інформаційних технологій, які можуть

використовуватися за межами не тільки інтранету компанії, але й мережі на базі комп'ютерних серверів. Це дозволяє численним користувачам та різним територіям долучатися до бухгалтерської інформації з різних відділів компанії. Компанії мають можливість створювати централізований бухгалтерський облік, використовуючи територіальні, регіональні, або національні офісні настройки. Централізований бухгалтерський облік забезпечує безпеку та гарантує, що він ведеться надійними робітниками. Він також зберігає витрати продуктивних сил тим, що дозволяє бухгалтерам в центральному офісі долучатися напряму до фінансової інформації з відділів на відстані завдяки використанню інтранету компанії [2].

Автоматизована інформаційна система є поліструктурним об'єктом і включає взаємопов'язану сукупність комплексу апаратно-програмних засобів для збирання, передачі і обробки інформації, відповідної інформаційної бази і персоналу, який бере участь у обробці, збиранні, передачі і використанні інформації. Облік та аналіз є важливими функціональними компонентами інформаційної системи підприємства. У свою чергу, впровадження інформаційних технологій відповідає завданням бухгалтерського обліку. Впровадження ІТ доцільно здійснювати, дотримуючись чотирьох фаз: прийняття рішення про інвестування в ІТ та його перевірка; освоєння ІТ та їх адаптування до професійних потреб; керування інформатизацією та її контроль; розповсюдження ІТ на інші сфери діяльності на підприємстві.

Широке використання обчислювальної техніки спочатку зумовило централізовану обробку облікової інформації, оскільки всі обчислювальні пристрої були в одному відділі підприємства. Висока вартість комп'ютерів другого і третього покоління, складність експлуатації та обслуговування спричиняли необхідність створення великих централізованих обчислювальних центрів зі спеціальним штатом операторів та програмістів і, як наслідок, потребували максимального ступеня централізації обліку. На великих підприємствах, де збереглися цехові бухгалтерії, децентралізація обліку втратила свої звичайні форми, оскільки цехові бухгалтерії отримували готові зведення від обчислювального центру. Функції цехових бухгалтерій зводилися лише до контролю за правильністю оформлення первинних

документів та отриманих звітних зведень. Саме це й дало підставу проф. П.О. Додонову стверджувати про однозначну централізацію облікового апарату в умовах застосування обчислювальної техніки. Тільки такий ступінь централізації обліку забезпечував повне завантаження обчислювальних центрів, сприяв уніфікації бухгалтерського обліку, зниженню витрат на його ведення.

Однак максимальна централізація обліку мала і негативні наслідки: обробка облікової інформації відривалася від місця здійснення господарських операцій, збільшувалася тривалість документообігу, підвищувалися витрати на переміщення носіїв інформації до місця їх обробки тощо.

З розвитком можливостей обчислювальної техніки та її здешевленням, з появою персональних комп'ютерів, зі створенням АРМ бухгалтерів виникла можливість децентралізувати облік, оскільки тепер обробка первинної облікової інформації та формування кінцевої інформації може відбуватися безпосередньо на робочих місцях облікових працівників.

Розглядаючи поняття централізації і децентралізації стосовно як автоматизованого бухгалтерського обліку, так і без комп'ютерних способів обробки облікової інформації, слід мати на увазі, що вони є досить умовними. Відносність категорій централізації і децентралізації як у паперових, так і в комп'ютерних формах обліку наочно показав проф. К.Н. Нарібаєв, який довів відносність цих категорій при побудові облікового апарату. Науковець писав, що "...реально не може існувати повністю централізована система, так само як і повністю децентралізована система стає недієздатною. Можна говорити про ступінь централізації і децентралізації як про оптимальне поєднання цих процесів при певних умовах". Отже, є дві конкуруючі тенденції, які впливають на вибір структурної схеми системи обробки облікових даних на сучасному етапі розвитку технічних засобів обчислювальної техніки та засобів зв'язку. Перша полягає у підвищенні ступеня централізації обробки, друга – в посиленні децентралізації. Тенденції породили абсолютно нові риси в централізації та децентралізації обліку

Узагальнення українського та світового досвіду організації обліку в умовах його комп'ютеризації дали можливість визначити основну тенденцію в централізації і децентралізації виконання облікових функцій. Вона полягає в накопиченні і централізованій обробці облікових даних у єдиному центрі та децентралізованому введенні первинної інформації в базу даних і спільному використанні підсумкових облікових показників. КСБО поєднує переваги централізованої обробки даних з перевагами децентралізації.

В умовах швидкого розвитку інформаційних технологій та глобальної мережі Інтернет підприємства відмовляються використовувати власне обладнання та програмне забезпечення, віддаючи перевагу сервіс-орієнтованим «хмарним» технологіям, що потребує удосконалення методики обліку на підприємстві.

За умови використання хмарних обчислень користувачеві надається програмне забезпечення у формі Інтернет-сервісу, забезпечується доступ до власних даних, однак існують певні обмеження щодо управління операційною системою та програмним забезпеченням, з яким він працює. Метафорично «хмарою» називають мережу Інтернет. Інститутом інженерів з електроніки й електротехніки хмарні обчислення трактуються як парадигма. Інформація обробляється на клієнтській стороні та постійно зберігається на серверах, що в свою чергу розташовані в мережі Інтернет. Завдяки чому спрощується система доступу до баз даних.

На думку В.В. Муравського, «...поширення хмарних технологій сприяло розвитку комбінованого варіанту централізації обліку. При передачі хмарному середовищу виконання усіх обчислювальних процедур від первинного збору облікових даних до автоматизованого відправлення звітних документів зацікавленим фізичним та юридичним особам відбувається централізація обліку за комбінованим варіантом». Застосування хмарних технологій дозволяє говорити про необхідність виділення комбіновано-центрального варіанту організації обліку, що і визначає найбільш популярні форми організації обліку.

На сьогодні традиційно найбільш поширеним для нашої країни є бухгалтерський інсорсинг, тобто ведення обліку на підприємстві

бухгалтерською службою чи керівником (за умови дотримання вимог чинного законодавства). На думку Г.С. Кесарчук, тільки фірми, де недоцільно мати свою бухгалтерську службу у зв'язку з незначним обсягом доходу, доручають облік вести спеціалізованим компаніям. Проте практика останніх років свідчить про поширення передачі підприємствами на аутсорсинг частини бухгалтерських операцій.

Використання ж хмарних технологій на підприємстві може змінити позицію керівництва щодо доцільності отримання послуг аутсорсингової фірми. Адже, як правило, вдаються до бухгалтерського аутсорсингу представники малого та середнього бізнесу, які мають незначний обсяг операцій та намагаються зменшити свої витрати шляхом виключення зі штатного розпису посади бухгалтера та уникнення витрат на придбання технічного та програмного забезпечення. Якщо ж буде обрано варіант ведення обліку штатним бухгалтером (на неповний робочий день, із можливістю віддаленого робочого місця тощо) на основі використання хмарного сервісу, то можна уникнути не тільки зайвих витрат, але й забезпечити оперативність обробки даних, яка втрачається при аутсорсингу. Розуміючи це, аутсорсингові фірми також усе частіше вдаються до використання хмарних технологій. Так, обробляючи відскановані працівником підприємства-замовника первинні документи, бухгалтер аутсорсингової фірми отримує актуальну інформацію про стан справ підприємства-замовника, яка може бути використана для забезпечення додаткових послуг з додатковою оплатою.

За умови використання сучасних комунікаційних технологій дистанційна передача облікових даних здатна забезпечити зростання оперативності, своєчасності та економічної ефективності обліку. Автоматизація обліку дозволяє одноразово зібраною інформацією забезпечити здійснення усіх видів обліку, що разом з єдиною базою облікової інформації і є основою єдиного (інтегрованого) обліку та інтегрованої звітності.

Отже, інформаційну систему можна визначити як набір процедур: процес, збір і переробка інформації для підтримки планування, прийняття рішень, координації і

контролю, а комунікаційна функція системи полягає в швидкому та надійному передаванні даних.

РОЗДІЛ 3 ОПИС ПРОГРАМИ

3.1 Вимоги до програми [9-10]

Потрібно написати програму, яка повинна продемонструвати роботу з базою даних. Дана програма повинна забезпечувати:

Створення автоматизованого обліку співробітників фірми

Введення нових даних

Коректування текстового файлу

Можливість виводу на екран дисплея всього переліку наявних в текстовому файлі співробітників

Пошук по певних критеріях і виведення результату

Повинні бути автоматизовані процедури пошуку по даті народження.

За допомогою цієї програми можна буде вводити наступну інформацію про співробітників фірми:

Прізвище

Ім'я

По батькові

Дата народження

Місце народження.

Окрім вищеперелічених можливостей програма повинна мати також зручний інтерфейс для створення користувачеві комфортних умов роботи з програмою.

Вхідна оперативна інформація

дані, що зберігаються в текстовому файлі;

запити користувача.

Вихідна інформація

Інформація, що зберігається в текстовий файл;

Результати автоматизованого пошуку;

Вибирання програмних засобів, середовища реалізації і операційної системи

Для написання даного програмного продукту вибрана мова програмування Delphi пакет Borland Delphi 7 Studio

Інтегроване середовище розробки є багатооконною системою. Вона включає в себе все необхідне для швидкої розробки Windows-додатків, і може гнучко налаштовуватися. Результуючі програми добре оптимізовані за швидкістю виконань і витратам пам'яті. Дизайнер форм, інспектор об'єктів і інші засоби залишаються доступними під час роботи програми, тому вносити зміни можна в процесі відладки.

3.2 Проектування інтерфейсу програми

Спроекуємо інтерфейс програми. Borland Delphi 7 це робиться за допомогою форм. Форма є екранним об'єктом, зазвичай прямокутної форми, який можна застосовувати для надання інформації користувачеві і для обробки введення інформації від користувача. Саме на ній ми і розташовуватимемо решту всіх елементів інтерфейсу програми.

Розмістимо спершу на формі компонент, за допомогою якого здійснюватиме відображення і редагування записів, що містяться в наборі даних і таких, що представляються на регулярній сітці. За допомогою цього компоненту організуємо результати пошуків по базі даних. Додали 3 компоненти для управління базою даних (Додавання, Видалення, Редагування записів).

Кожен набір даних повинен асоціюватися з джерелом, щоб маніпулювати даними за допомогою компонент управління. З іншого боку, кожна компоненту управління повинна асоціюватися з джерелом, щоб вона могла приймати дані і маніпулювати ними.

Для введення і редагування записів використовуємо компоненти Edit.

Діаграма варіантів використання

Діаграма варіантів використання описує функціональне призначення системи або, іншими словами, те, що система робитиме в процесі свого функціонування. Діаграма варіантів використання є початковим

концептуальним уявленням або концептуальною моделлю системи в процесі її проектування і розробки.

Програма повинна виконувати, а саме:

читання інформації з текстового файлу;

проглядання інформації про співробітників фірми;

можливість додавання, редагування і видалення записів;

автоматизований пошук по критеріях;

Діаграма діяльності

Для моделювання процесу виконання операцій використовуються так звані діаграми діяльності. Кожне перебування на діаграмі діяльності відповідає виконанню деякої елементарної операції, а перехід в наступний стан спрацьовує тільки при завершенні цієї операції в попередньому стані.

Код програми складається з декількох цілком логічно закінчених частин, які умовно можна розділити на наступні групи:

1. код головної форми, який включає опис таких операцій, як

визначення фізичного шляху до текстового файлу;

додавання і видалення записів з текстового файлу;

автоматизований пошук по певному критерію;

виклик форм, пов'язаних з головною формою;

2. код форм, пов'язаних з головною формою, в яких міститься опис такої операції, як виклик довідки і допомоги.

Весь код додатку розташовується в 9 Unit'ах. Кожен Unit містить код обробки подій компонентів, розташованих на відповідних формах (Form), тобто Unit1 відповідає Form1, Unit2 – Form2 і так далі. Таким чином, структура програми складається з 9 логічно закінчених частин, які відповідають за виконання всіх вищеперелічених дій.

Лістинг програми у додатку А.

3.3 Опис застосування програми

Установка програми проводиться шляхом копіювання теки з виконуваним файлом програми і файлами, що містять записи бази даних.

Програмний продукт, написаний в середовищі Borland delphi 7 використанням візуальних компонентів не вимогливий до ресурсів системи.

При завантаженні програми з'являється вікно загрузки потім вікно введення пароля та представлений на рис 3.1-3.3, також автоматично завантажується база співробітників. Якщо ж бази не існує, вона створюється автоматично.

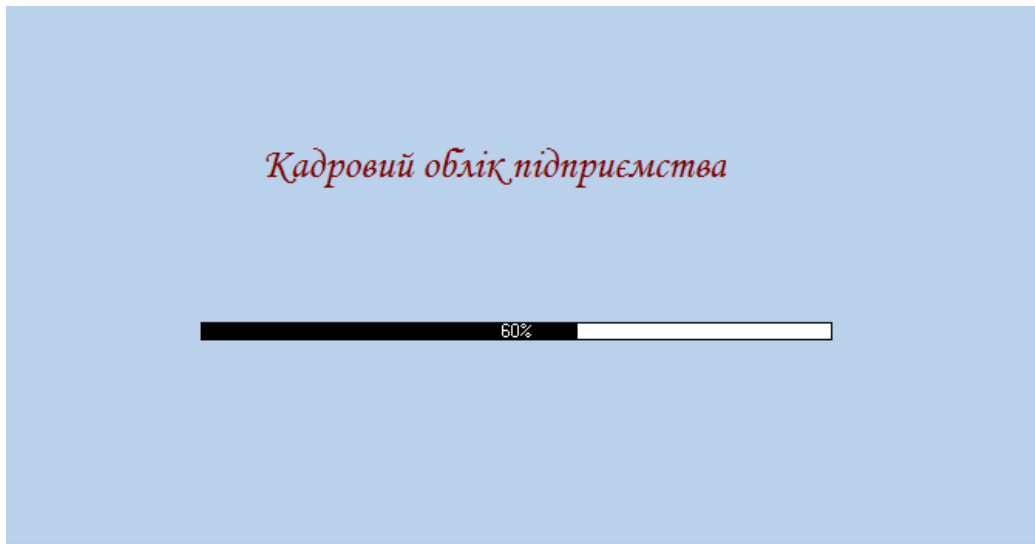


Рисунок 3.1

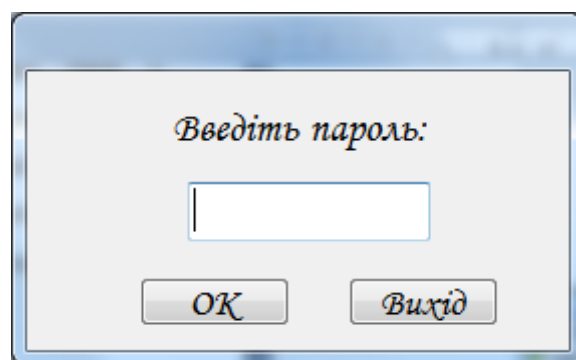


Рисунок 3.2 - вікно зміни пароля

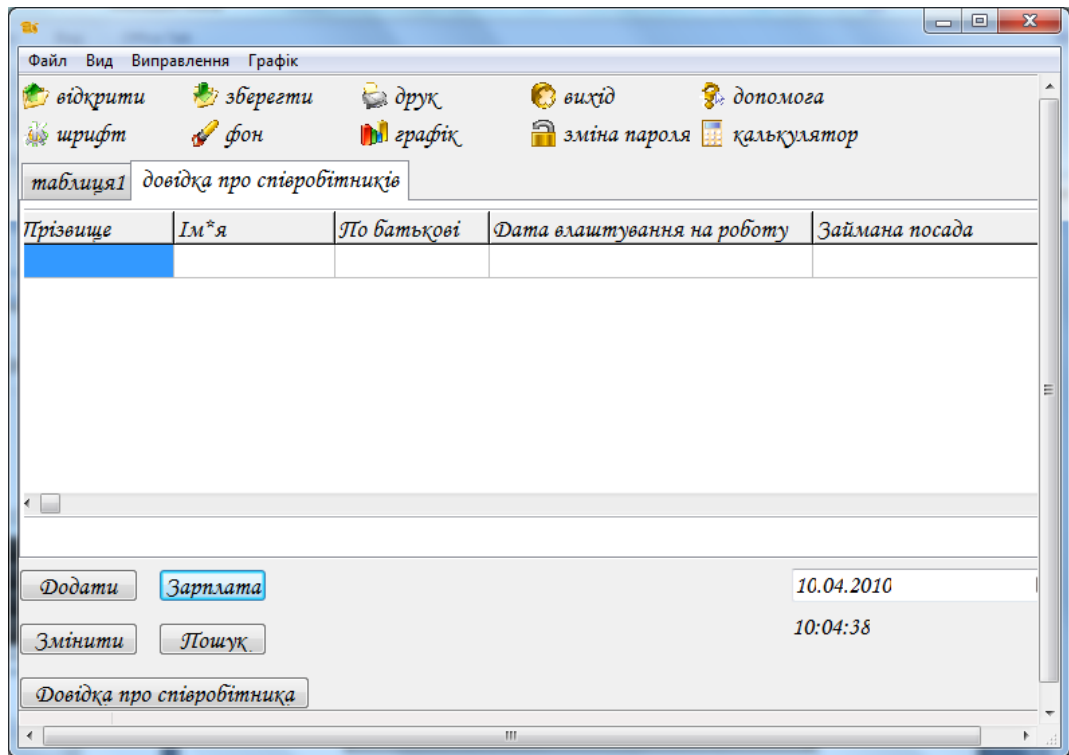


Рисунок 3.3 - Головна форма додатку

При натисненні на кнопку «Додати», з'явиться вікно додавання запису. Також присутня кнопка змінити запису див. рис. 3 4

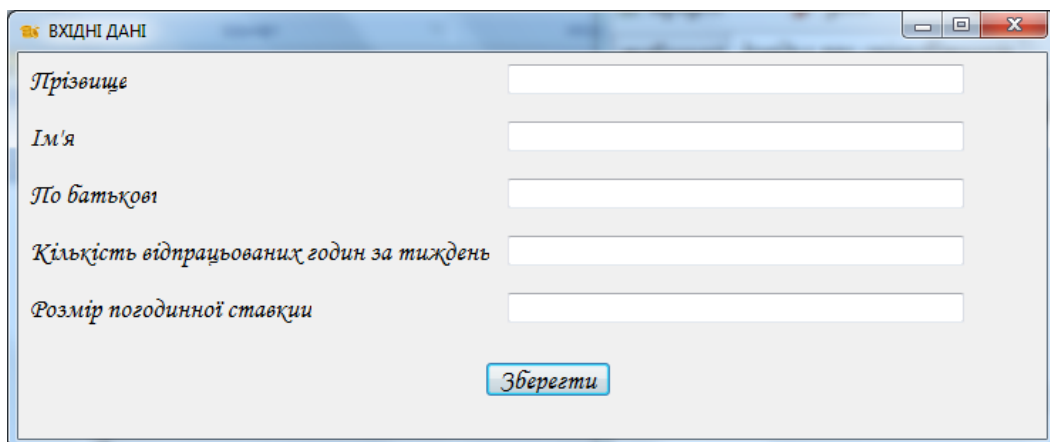


Рисунок 3.4 – Форма додавання запису

В програмі є можливість зміни пароля рис. 3 5

ЗМІНА ПАРОЛЯ

Введіть старий пароль:

Введіть новий пароль:

Повторіть новий пароль:

Рисунок 3.5 вікно зміни пароля

ДОВІДКА про співробітників

<i>Прізвище</i>	<i>Ім'я</i>	<i>По батькові</i>	
<input type="text" value="Петренко"/>	<input type="text" value="Остап"/>	<input type="text" value="Миколаевич"/>	
<i>Дата влаштування на роботу</i>		<i>Займана посада</i>	
<input type="text" value="20.10.2015"/>		<input type="text" value="економіст"/>	
<i>ІНП</i>	<i>Пенсійний</i>	<i>Дата народження</i>	
<input type="text" value="2134568"/>	<input type="text" value="789456"/>	<input type="text" value="20.10.1982"/>	
<i>Серія паспорта</i>	<i>Номер паспорта</i>	<i>Виданий паспорт</i>	
<input type="text" value="EE"/>	<input type="text" value="215896"/>	<input type="text" value="Ноль"/>	
<i>Місто</i>	<i>Вулиця</i>	<i>Будинок</i>	<i>Квартира</i>
<input type="text" value="Ноль"/>	<input type="text" value="Два"/>	<input type="text" value="10"/>	<input type="text" value="112"/>
<i>Телефон</i>	<i>Сімейний стан</i>	<i>Кількість дітей</i>	
<input type="text" value="123456"/>	<input type="text" value="одружений"/>	<input type="text" value="1"/>	

Для виходу з програми можна натиснути на хрестик, розташований в правому верхньому кутку або натиснути вихід

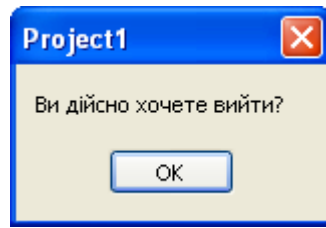


Рисунок 3.5 – Вихід

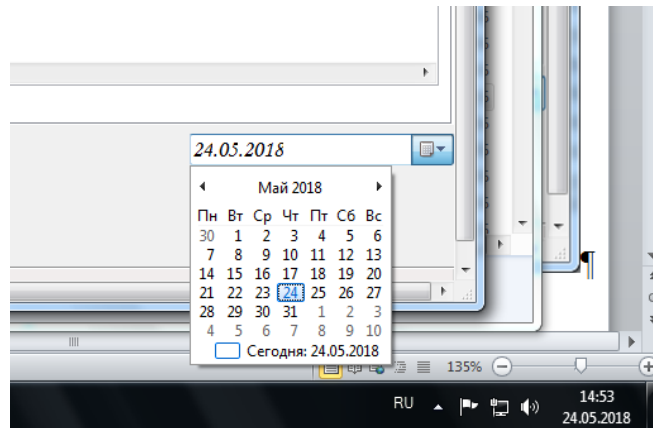


Рисунок 3.6 –Виставлення поточної дати

Фамилия	имя	отчество	подсчет з/п
Петров	Иван	Ильич	975
Васильев	Егор	Петрович	672

Рисунок 3.7 –Зарплата працівників

ВХІДНІ ДАНІ

Прізвище: Петров

Ім'я: Іван

По батькові: Ільич

Кількість відпрацьованих годин за тиждень: 39

Розмір погодинної ставки: 25

Зберегти

Рисунок 3.8 – Робота з даними

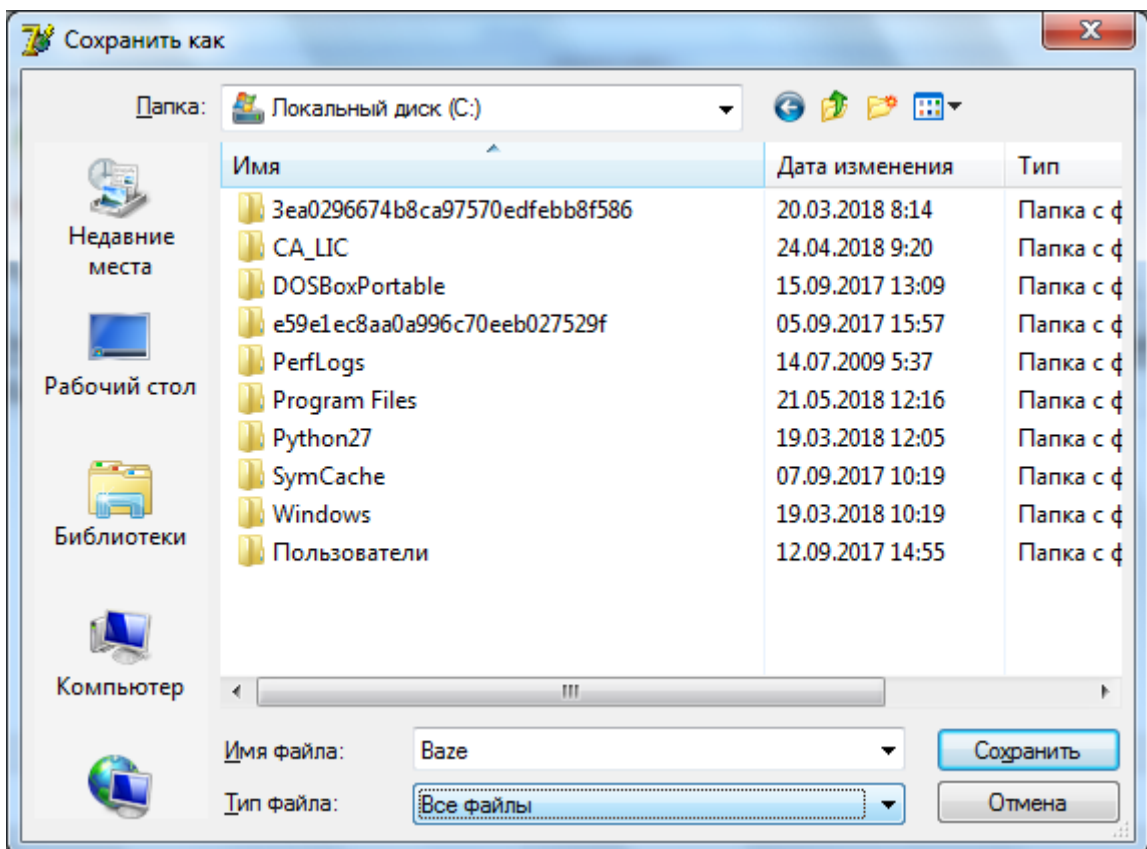


Рисунок 3.9 – збереження даних

ВИСНОВКИ

При виконанні проекту були пройдені всі етапи розробки спеціалізованого програмного забезпечення:

формалізація завдання;

збір необхідних початкових даних, використовуваних в програмі;

В процесі виконання роботи була проведена наступна робота:

1. Освоєння принципів розробки алгоритмів вирішення завдань на ПЕВМ;

2. Ознайомлення з різними сферами застосування ПЕВМ в сучасному суспільстві, придбання основних навиків роботи з персональним комп'ютером.

3. Розроблена програма яка дозволяє облік співробітників підприємства.

1. Половнев Н.М., Якимов А.М. Системы автоматизированной обработки учетной информации. - М.: Финансы и статистика, 1994. - 187 с.
2. Твердохлеб Н.Г., Татарчук Н.И., Сендзюк М.А. Организация машинной обработки экономической информации в промышленности. - К.: Вища школа, 1984. - 416с.
3. Сытник В.Ф., Пинчук Н.С., Волк Б.Г. Автоматизация расчетов по материально-техническому обеспечению производства. - К.: Техника, 1990. - 168 с.
4. Полковский Л.М., Зайдман С.А., Беркович М.Е. Автоматизация учета на базе персональных ЭВМ. - М.: Финансы и статистика, 1991. - 192 с.
5. Компьютеризация информационных процессов на промышленных предприятиях / В.Ф. Сытник, Х. Срока, Н.В. Еремина. - К.: Катовице: Техника: Экономическая академия им. К. Адамецкого, 1991. - 215 с.
6. Пономарева К.В., Кузьмин Л.Г. Информационное обеспечение АСУ. - М.: Высшая школа, 1991. - 222 с.
7. Пономарева К.В. Информационное обеспечение АСУ. - М.: Высшая школа, 1991. - 350с.
8. Попов А.А. Программирование в среде СУБД FoxPro 2.0. - М.: Радио и связь, 1993. - 352 с.
9. Потапенко Н.И., Орешко А.А. Конструирование программ и языки программирования. - Мн.: МГВРК, 1999.- 32 с.
10. Шамис В.А. Borland BORLAND DELPHI 7 Builder 6. Для профессионалов - СПб.: Питер, 2005 г. - 798 с.: ил.

Додаток А

Лістинг програми:

```
unit Unit1;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, Menus, StdCtrls, Grids, ComCtrls, ToolWin, ImgList, AppEvnts,  
ExtCtrls, XPMAN, ShellAPI;
```

```
type
```

```
sport = record
```

```
fam,name,otch,datau,dolg,INN,Pens,DR,spas,Npas,vpas,city,str,dom,kv,tel,sem,dety:string[40];  
end;
```

```
dengi=record
```

```
Fam:string[20];
```

```
Im:string[15];
```

```
ot:string[20];
```

```
kol_hour:string[10];
```

```
tarif:string[20];
```

```
end;
```

```
zap=record
```

```
sp:sport;
```

```
den:dengi;
```

```
end;
```

```
TForm1 = class(TForm)
```

```
MainMenu1: TMainMenu;
```

```
N1: TMenuItem;
```

```
PopupMenu1: TPopupMenu;
```

```
Button2: TButton;
```

```
Button3: TButton;
```

```
N10: TMenuItem;
```

N12: TMenuItem;
N13: TMenuItem;
N4: TMenuItem;
N14: TMenuItem;
N15: TMenuItem;
N16: TMenuItem;
N17: TMenuItem;
N18: TMenuItem;
N19: TMenuItem;
N20: TMenuItem;
N21: TMenuItem;
N22: TMenuItem;
Button1: TButton;
N23: TMenuItem;
ToolBarMenuItem: TMenuItem;
StatusBarMenuItem: TMenuItem;
N27: TMenuItem;
ToolBar1: TToolBar;
StatusBar1: TStatusBar;
Button5: TButton;
ImageList1: TImageList;
N11: TMenuItem;
OpenDialog1: TOpenDialog;
N24: TMenuItem;
ColorDialog1: TColorDialog;
DateTimePicker1: TDateTimePicker;
SaveDialog1: TSaveDialog;
PageControl1: TPageControl;
TabSheet1: TTabSheet;
StringGrid1: TStringGrid;
TabSheet2: TTabSheet;
StringGrid2: TStringGrid;
FontDialog1: TFontDialog;
N25: TMenuItem;
N26: TMenuItem;
N7: TMenuItem;
N8: TMenuItem;

N9: TMenuItem;
N28: TMenuItem;
N31: TMenuItem;
N30: TMenuItem;
ToolButton1: TToolButton;
ToolButton2: TToolButton;
ToolButton3: TToolButton;
ToolButton4: TToolButton;
ToolButton5: TToolButton;
ToolButton6: TToolButton;
ToolButton7: TToolButton;
ToolButton8: TToolButton;
ToolButton9: TToolButton;
ToolButton10: TToolButton;
N33: TMenuItem;
ApplicationEvents1: TApplicationEvents;
Timer1: TTimer;
Button6: TButton;
Label1: TLabel;
N32: TMenuItem;
XPManifest1: TXPManifest;
procedure FormShow(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure N5Click(Sender: TObject);
procedure N13Click(Sender: TObject);
procedure ToolBarMenuItemClick(Sender: TObject);
procedure StatusBarMenuItemClick(Sender: TObject);
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button5Click(Sender: TObject);
procedure ToolButton7Click(Sender: TObject);
procedure N24Click(Sender: TObject);
procedure N11Click(Sender: TObject);
procedure N10Click(Sender: TObject);
procedure ToolButton8Click(Sender: TObject);
procedure ToolButton3Click(Sender: TObject);
procedure ToolButton2Click(Sender: TObject);

```
procedure N25Click(Sender: TObject);
procedure N26Click(Sender: TObject);
procedure N31Click(Sender: TObject);
procedure N30Click(Sender: TObject);
procedure ApplicationEvents1ActionExecute(Action: TBasicAction;
  var Handled: Boolean);
procedure Timer1Timer(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure ToolButton11Click(Sender: TObject);
procedure ToolButton4Click(Sender: TObject);
procedure ToolButton1Click(Sender: TObject);
procedure N12Click(Sender: TObject);
procedure ToolButton6Click(Sender: TObject);
procedure N9Click(Sender: TObject);
procedure ToolButton9Click(Sender: TObject);
procedure N33Click(Sender: TObject);
procedure N2Click(Sender: TObject);
procedure ToolButton10Click(Sender: TObject);
procedure N6Click(Sender: TObject);
procedure ToolButton5Click(Sender: TObject);
```

```
private
  { Private declarations }
public
  { Public declarations }
end;
```

```
var
  s:sport;
  Form1: TForm1;
  n:integer;
  h:boolean;
```

```
implementation
```



```
uses Unit7, Unit2, Unit5, Unit4, Unit8, Unit9, Unit10, Unit11, Unit12,  
Unit13;
```

```
{ $R *.dfm }
```

```
procedure TForm1.FormShow(Sender: TObject);  
begin  
form7.ShowModal;  
with form1.stringgrid1 do  
begin  
cells[0,0]:= Прізвище;  
cells[1,0]:= Ім'я;  
cells[2,0]:= По батькові;  
cells[3,0]:= Кількість відпрацьованих годин за тиждень';  
cells[4,0]:= Розмір погодинної ставки;  
end;  
with form1.stringgrid2 do  
begin  
cells[0,0]:= Прізвище;  
cells[1,0]:= Ім'я;  
cells[2,0]:= По батькові;  
cells[3,0]:= Дата влаштування на роботу;  
cells[4,0]:= Займана посада;  
cells[5,0]:= ІПН;  
cells[6,0]:= Пенсійний;  
cells[7,0]:= Дата народження';  
cells[8,0]:= Серія паспорта;  
cells[9,0]:= Номер паспорта;  
cells[10,0]:= Ким виданий;  
cells[11,0]:= Місто;  
cells[12,0]:= Вулиця;  
cells[13,0]:= Будинок;  
cells[14,0]:= Квартира;  
cells[15,0]:= Телефон;  
cells[16,0]:= Сімейний стан;  
cells[17,0]:= Кількість дітей;
```

```

end;
end;

procedure TForm1.Button1Click(Sender: TObject);{ прописана кнопка додати в таб.1 }
begin
form2.Show;
Form2.Button3.Visible:=false;
Form2.Button4.Visible:=false;

end;

procedure TForm1.N5Click(Sender: TObject); { Про програму }
begin
Form5.Showmodal
end;

procedure TForm1.N13Click(Sender: TObject);{ вихід з програми }
begin
If Tag = 0 Then
Begin
ShowMessage( 'Ви дійсно хочете вийти?');
Tag:=1;
    Close;
end;
end;

procedure TForm1.ToolBarMenuItemClick(Sender: TObject);
begin
    // Включити або виключити відображення панель інструментів
    ToolBar1.Visible := not ToolBar1.Visible;
    ToolBarMenuItem.Checked := not ToolBarMenuItem.Checked;
end;

procedure TForm1.StatusBarMenuItemClick(Sender: TObject);
begin
    // Включити або виключити відображення рядок стану
    StatusBar1.Visible := not StatusBar1.Visible;
    StatusBarMenuItem.Checked := not StatusBarMenuItem.Checked;

```

end;

```
procedure TForm1.Button2Click(Sender: TObject); { підрахунок суми двох останніх  
стовпців, зарплата }
```

```
var
```

```
  i:integer;
```

```
begin
```

```
if form1.StringGrid1.Cells[0,1]<>" then begin
```

```
  for i:=1 to form9.stringgrid1.rowcount do
```

```
    form9.StringGrid1.rows[i].Clear;
```

```
      for i:=1 to form9.stringgrid1.rowcount-1 do begin
```

```
        if form1.stringgrid1.Cells[0,i]=" then break;
```

```
        form9.stringgrid1.Cells[0,i]:=form1.stringgrid1.Cells[0,i];
```

```
        form9.stringgrid1.Cells[1,i]:=form1.stringgrid1.Cells[1,i];
```

```
        form9.stringgrid1.Cells[2,i]:=form1.stringgrid1.Cells[2,i];
```

```
form9.stringgrid1.Cells[3,i]:=inttostr(strtoint(form1.stringgrid1.Cells[3,i])*strtoint(form1.stringg  
rid1.Cells[4,i]));
```

```
      end; end;
```

```
      form9.showmodal;
```

```
end;
```

```
procedure TForm1.Button3Click(Sender: TObject); { довідка про співробітника у  
вигляді анкети }
```

```
begin
```

```
  i:=form1.StringGrid2.Row;
```

```
  with form1.StringGrid2 do begin
```

```
    form4.Edit1.Text:=cells[0,i];
```

```
    form4.Edit2.Text:=cells[1,i];
```

```
    form4.Edit3.Text:=cells[2,i];
```

```
    form4.Edit4.Text:=cells[3,i];
```

```
    form4.Edit6.Text:=cells[4,i];
```

```
    form4.Edit5.Text:=cells[5,i];
```

```
    form4.Edit7.Text:=cells[6,i];
```

```
    form4.Edit11.Text:=cells[7,i];
```

```
form4.Edit9.Text:=cells[9,i];
form4.Edit10.Text:=cells[10,i];
form4.Edit12.Text:=cells[11,i];
form4.Edit13.Text:=cells[12,i];
form4.Edit14.Text:=cells[13,i];
form4.Edit15.Text:=cells[14,i];
form4.Edit17.Text:=cells[15,i];
form4.Edit18.Text:=cells[16,i];
form4.Edit16.Text:=cells[17,i];
end;
form4.show;
end;
```

```
procedure TForm1.Button5Click(Sender: TObject); {виклик форми пошук по
прізвищу}
```

```
begin
form8.ShowModal;

end;
```

```
procedure TForm1.ToolButton7Click(Sender: TObject);{прописана кнопка зміна
фону}
```

```
begin
if ColorDialog1.Execute then
    Form1.Color := ColorDialog1.Color;

end;
```

```
procedure TForm1.N24Click(Sender: TObject);{Зміна пароля }
begin
```

```
form10.edit1.text:=";
```

```
Form10.show;
```

```
end;
```

```
procedure TForm1.N11Click(Sender: TObject); { зберегти файл }
```

```
var
```

```
FileOut:file of zap;
```

```
z:zap;
```

```
i, j:integer;
```

```
route:string;
```

```
s:sport;
```

```
begin
```

```
if SaveDialog1.Execute then
```

```
begin
```

```
route := SaveDialog1.FileName;
```

```
assignfile(fileout,route);
```

```
rewrite(fileout);
```

```
if stringgrid1.rowcount>stringgrid2.rowcount then j:=stringgrid1.rowcount else  
j:=stringgrid2.rowcount;
```

```
with form1.stringgrid1 do
```

```
for i:=1 to j do
```

```
begin
```

```
if cells[0,i]<>" then begin
```

```
z.den.fam:=cells[0,i];
```

```
z.den.im:=cells[1,i];
```

```
z.den.ot:=cells[2,i];
```

```
z.den.kol_hour:=cells[3,i];
```

```
z.den.tarif:=cells[4,i]; end else
```

```
begin
```

```
z.den.fam:=";
```

```
z.den.im:=";
```

```
z.den.ot:=";
```

```
z.den.kol_hour:=";
```

```
z.den.tarif:=";
```

```
end;
```

```
if i<=stringgrid2.RowCount then
```

with form1.StringGrid2 do

begin

```
z.sp.fam:=cells[0,i];
z.sp.name:=cells[1,i];
z.sp.otch:=cells[2,i];
z.sp.datau:=cells[3,i];
z.sp.dolg:=cells[4,i];
z.sp.INN:=cells[5,i];
z.sp.Pens:=cells[6,i];
z.sp.DR:=cells[7,i];
z.sp.spas:=cells[8,i];
z.sp.Npas:=cells[9,i];
z.sp.vpas:=cells[10,i];
z.sp.city:=cells[11,i];
z.sp.str:=cells[12,i];
z.sp.dom:=cells[13,i];
z.sp.kv:=cells[14,i];
```

```
z.sp.tel:=cells[15,i];
z.sp.sem:=cells[16,i];
z.sp.dety:=cells[17,i];
```

end else

begin

```
z.sp.fam:="";
z.sp.name:="";
z.sp.otch:="";
z.sp.datau:="";
z.sp.dolg:="";
z.sp.INN:="";
z.sp.Pens:="";
z.sp.DR:="";
z.sp.spas:="";
z.sp.Npas:="";
z.sp.vpas:="";
z.sp.city:="";
z.sp.str:="";
z.sp.dom:="";
```

```
z.sp.kv:="";
z.sp.sem:="";
z.sp.dety:="";
z.sp.tel:="";
end;
write(fileout,z);
end;
closefile(fileout);
end;
```

```
end;
```

```
procedure TForm1.N10Click(Sender: TObject); { відкрити файл }
```

```
var
```

```
FileOut:file of zap;
```

```
z:zap;
```

```
i,j:integer;
```

```
route:string;
```

```
begin
```

```
if OpenFileDialog1.Execute then
```

```
begin
```

```
route := OpenFileDialog1.FileName;
```

```
with stringgrid1 do
```

```
begin
```

```
for i:=1 to rowcount do
```

```
rows[i].Clear;
```

```
assignfile(fileout,route);
```

```
reset(fileout);
```

```
i:=0;
```

```
while not eof(fileout) do
```

```
begin
```

```
i:=i+1;
```

```

read(fileout,z);
cells[0,i]:=z.den.Fam;
cells[1,i]:=z.den.Im;
cells[2,i]:=z.den.ot;
cells[3,i]:=z.den.kol_hour;
cells[4,i]:=z.den.tarif;
if z.sp.fam<>" then
with stringgrid2 do
begin
if i<>1 then form1.stringgrid2.rowcount:=form1.stringgrid2.rowcount+1;
cells[0,i]:=z.sp.fam;
cells[1,i]:=z.sp.name;
cells[2,i]:=z.sp.otch;
cells[3,i]:=z.sp.datau;
cells[4,i]:=z.sp.dolg;
cells[5,i]:=z.sp.INN;
cells[6,i]:=z.sp.Pens;
cells[7,i]:=z.sp.DR;
cells[8,i]:=z.sp.spas;
cells[9,i]:=z.sp.Npas;
cells[10,i]:=z.sp.vpas;
cells[11,i]:=z.sp.city;
cells[12,i]:=z.sp.str;
cells[13,i]:=z.sp.dom;
cells[14,i]:=z.sp.kv;
cells[15,i]:=z.sp.tel;
cells[16,i]:=z.sp.sem;
cells[17,i]:=z.sp.dety;
end;
end;
closefile(fileout);
end;
end;
end;
end;

```



```
procedure TForm1.ToolButton8Click(Sender: TObject);{ прописана кнопка графік }
begin
if form1.StringGrid1.Cells[0,1]<>" then begin
form12.Show;
end else showmessage(Недостатньо даних для побудови діаграми);
end;
```

```
procedure TForm1.ToolButton3Click(Sender: TObject);{прописана кнопка друк
голова форма таб.1 }
```

```
begin
StatusBar1.visible:=false;
Button1.visible:=false;
Button2.visible:=false;
Button3.visible:=false;
Button5.visible:=false;
Button6.visible:=false;
DateTimePicker1.visible:=false;
Label1.visible:=false;
ToolButton1.visible:=false;
ToolButton2.visible:=false;
ToolButton3.visible:=false;
ToolButton4.visible:=false;
ToolButton5.visible:=false;
ToolButton6.visible:=false;
ToolButton7.visible:=false;
ToolButton8.visible:=false;
ToolButton9.visible:=false;
ToolButton10.visible:=false;
ToolBar1.visible:=false;
StringGrid2.visible:=false;

PageControl1.Visible:=true;
StringGrid1.visible:=true;

Form1.Print;

StatusBar1.visible:=true;
```

```
Button1.visible:=true;
Button2.visible:=true;
Button3.visible:=true;
Button5.visible:=true;
Button6.visible:=true;
DateTimePicker1.visible:=true;
Label1.visible:=true;
ToolButton1.visible:=true;
ToolButton2.visible:=true;
ToolButton3.visible:=true;
ToolButton4.visible:=true;
ToolButton5.visible:=true;
ToolButton6.visible:=true;
ToolButton7.visible:=true;
ToolButton8.visible:=true;
ToolButton9.visible:=true;
ToolButton10.visible:=true;
ToolBar1.visible:=true;
StringGrid2.visible:=true;
end;
```

```
procedure TForm1.ToolButton2Click(Sender: TObject);
var
  FileOut:file of zap;
  z:zap;
  i, j:integer;
  route:string;
  s:sport;
begin
  if SaveDialog1.Execute then
    begin
      route := SaveDialog1.FileName;
      assignfile(fileout,route);
      rewrite(fileout);
```

```

        if stringgrid1.rowcount>stringgrid2.rowcount then j:=stringgrid1.rowcount else
j:=stringgrid2.rowcount;
    with form1.stringgrid1 do
    for i:=1 to j do
    begin
        if cells[0,i]<>" then begin
            z.den.fam:=cells[0,i];
            z.den.im:=cells[1,i];
            z.den.ot:=cells[2,i];
            z.den.kol_hour:=cells[3,i];
            z.den.tarif:=cells[4,i]; end else
        begin
            z.den.fam:="";
            z.den.im:="";
            z.den.ot:="";
            z.den.kol_hour:="";
            z.den.tarif:="";
        end;
        if i<=stringgrid2.RowCount then
        with form1.StringGrid2 do
        begin
            z.sp.fam:=cells[0,i];
            z.sp.name:=cells[1,i];
            z.sp.otch:=cells[2,i];
            z.sp.datau:=cells[3,i];
            z.sp.dolg:=cells[4,i];
            z.sp.INN:=cells[5,i];
            z.sp.Pens:=cells[6,i];
            z.sp.DR:=cells[7,i];
            z.sp.spas:=cells[8,i];
            z.sp.Npas:=cells[9,i];
            z.sp.vpas:=cells[10,i];
            z.sp.city:=cells[11,i];
            z.sp.str:=cells[12,i];
            z.sp.dom:=cells[13,i];
            z.sp.kv:=cells[14,i];

```

```
z.sp.tel:=cells[15,i];
z.sp.sem:=cells[16,i];
z.sp.dety:=cells[17,i];
end else
begin
  z.sp.fam:="";
  z.sp.name:="";
  z.sp.otch:="";
  z.sp.datau:="";
z.sp.dolg:="";
z.sp.INN:="";
z.sp.Pens:="";
z.sp.DR:="";
z.sp.spas:="";
z.sp.Npas:="";
z.sp.vpas:="";
z.sp.city:="";
z.sp.str:="";
z.sp.dom:="";
z.sp.kv:="";
z.sp.sem:="";
z.sp.dety:="";
z.sp.tel:="";
end;
write(fileout,z);
end;
closefile(fileout);
end;
end;
```

```
procedure TForm1.N25Click(Sender: TObject);{ зміна фону }
begin
if ColorDialog1.Execute then
```

```
Form1.Color := ColorDialog1.Color;  
end;
```

```
procedure TForm1.N26Click(Sender: TObject);{ зміна шрифту }  
begin  
form1.FontDialog1.Execute;  
if form1.FontDialog1.Font.Name<>'MS Sans Serif' then  
form1.Font:=form1.FontDialog1.Font;  
end;
```

```
procedure TForm1.N31Click(Sender: TObject); { додати в таб.1 }  
begin  
form2.ShowModal;  
end;
```

```
procedure TForm1.N30Click(Sender: TObject);{ видалити }  
var  
i,k:integer;  
begin  
form1.stringgrid1.Rows[form1.stringgrid1.Row].Clear;  
k:=form1.stringgrid1.Row;  
for i:=form1.stringgrid1.row to form1.stringgrid1.rowcount do  
begin  
form1.StringGrid1.Rows[k]:= form1.StringGrid1.Rows[k + 1];  
k:=k+1;  
end;  
form1.StringGrid1.Rows[k + 1].Clear;  
  
end;
```

```
procedure TForm1.ApplicationEvents1ActionExecute(Action: TBasicAction;  
var Handled: Boolean);  
begin
```

```
with StatusBar1 do
begin
  SimpleText := Application.Hint;
  SimplePanel := SimpleText <> ";
end;
end;
```

```
procedure TForm1.Timer1Timer(Sender: TObject);
var
  DateTime: TDateTime;
begin
  DateTime:=Time;
  Label1.Caption:=TimeToStr(DateTime);
end;
```

```
procedure TForm1.Button6Click(Sender: TObject);{ прописана кнопка змінити }
begin
form2.show;
  i:=form1.StringGrid1.Row;
  with form1.StringGrid1 do begin
    form2.Edit1.Text:=cells[0,i];
    form2.Edit2.Text:=cells[1,i];
    form2.Edit3.Text:=cells[2,i];
    form2.Edit4.Text:=cells[3,i];
    form2.Edit5.Text:=cells[4,i];
  end;
end;
```

```
procedure TForm1.ToolButton11Click(Sender: TObject);
begin
  if form1.StringGrid1.Cells[0,1]<>" then begin
```

```
form12.Show;
end else showmessage('Недостатньо даних для побудови діаграми');
end;
```

```
procedure TForm1.ToolButton4Click(Sender: TObject);{прописана кнопка вихід з програми }
```

```
begin
If Tag = 0 Then
Begin
ShowMessage('Ви дійсно хочете вийти?');
Tag:=1;
Close;
End;
end;
```

```
procedure TForm1.ToolButton1Click(Sender: TObject); {прописана кнопка відкрити файл }
```

```
var
FileOut:file of zap;
z:zap;
i,j:integer;
route:string;
begin
if OpenFileDialog1.Execute then
begin
route := OpenFileDialog1.FileName;
with stringgrid1 do
begin
for i:=1 to rowcount do
rows[i].Clear;
assignfile(fileout,route);
reset(fileout);
i:=0;
while not eof(fileout) do
```

```

begin
  i:=i+1;
  read(fileout,z);
  cells[0,i]:=z.den.Fam;
  cells[1,i]:=z.den.Im;
  cells[2,i]:=z.den.ot;
  cells[3,i]:=z.den.kol_hour;
  cells[4,i]:=z.den.tarif;
  if z.sp.fam<>" then
  with stringgrid2 do
  begin
    if i<>1 then form1.stringgrid2.rowcount:=form1.stringgrid2.rowcount+1;
      cells[0,i]:=z.sp.fam;
      cells[1,i]:=z.sp.name;
      cells[2,i]:=z.sp.otch;
      cells[3,i]:=z.sp.datau;
      cells[4,i]:=z.sp.dolg;
      cells[5,i]:=z.sp.INN;
      cells[6,i]:=z.sp.Pens;
      cells[7,i]:=z.sp.DR;
      cells[8,i]:=z.sp.spas;
      cells[9,i]:=z.sp.Npas;
      cells[10,i]:=z.sp.vpas;
      cells[11,i]:=z.sp.city;
      cells[12,i]:=z.sp.str;
      cells[13,i]:=z.sp.dom;
      cells[14,i]:=z.sp.kv;
      cells[15,i]:=z.sp.tel;
      cells[16,i]:=z.sp.sem;
      cells[17,i]:=z.sp.dety;
    end;
  end;
  closefile(fileout);
  end;
end;
end;

```



```
procedure TForm1.N12Click(Sender: TObject);
begin
  StatusBar1.Visible:=false;
  Button1.Visible:=false;
  Button2.Visible:=false;
  Button3.Visible:=false;
  Button5.Visible:=false;
  Button6.Visible:=false;
  DateTimePicker1.Visible:=false;
  Label1.Visible:=false;
  ToolButton1.Visible:=false;
  ToolButton2.Visible:=false;
  ToolButton3.Visible:=false;
  ToolButton4.Visible:=false;
  ToolButton5.Visible:=false;
  ToolButton6.Visible:=false;
  ToolButton7.Visible:=false;
  ToolButton8.Visible:=false;
  ToolButton9.Visible:=false;
  ToolButton10.Visible:=false;
  ToolBar1.Visible:=false;
  StringGrid2.Visible:=false;

  PageControl1.Visible:=true;
  StringGrid1.Visible:=true;

  Form1.Print;

  StatusBar1.Visible:=true;
  Button1.Visible:=true;
  Button2.Visible:=true;
  Button3.Visible:=true;
  Button5.Visible:=true;
  Button6.Visible:=true;
  DateTimePicker1.Visible:=true;
  Label1.Visible:=true;
```

```
ToolButton1.visible:=true;
ToolButton2.visible:=true;
ToolButton3.visible:=true;
ToolButton4.visible:=true;
ToolButton5.visible:=true;
ToolButton6.visible:=true;
ToolButton7.visible:=true;
ToolButton8.visible:=true;
ToolButton9.visible:=true;
ToolButton10.visible:=true;
ToolBar1.visible:=true;
StringGrid2.visible:=true;
end;
```

```
procedure TForm1.ToolButton6Click(Sender: TObject);
begin
form1.FontDialog1.Execute;
if form1.FontDialog1.Font.Name<>'MS Sans Serif' then
form1.Font:=form1.FontDialog1.Font;
end;
```

```
procedure TForm1.N9Click(Sender: TObject); { графік }
begin
if form1.StringGrid1.Cells[0,1]<>" then begin
form12.Show;
end else showmessage ('Недостатньо даних для побудови діаграми ');
end;
```

```
procedure TForm1.ToolButton9Click(Sender: TObject); begin
Form10.showmodal;
end;
```

```
procedure TForm1.N33Click(Sender: TObject);
begin
form2.show;
i:=form1.StringGrid1.Row;
```

```
with form1.StringGrid1 do begin
  form2.Edit1.Text:=cells[0,i];
  form2.Edit2.Text:=cells[1,i];
  form2.Edit3.Text:=cells[2,i];
  form2.Edit4.Text:=cells[3,i];
  form2.Edit5.Text:=cells[4,i];
end;
end;
procedure TForm1.N2Click(Sender: TObject);
begin
form11.ShowModal;
end;

procedure TForm1.ToolButton10Click(Sender: TObject);
begin
form13.showmodal;
end;

procedure TForm1.N6Click(Sender: TObject);
begin
ShellExecute (Form1.Handle, nil, 'helpp.chm', nil, nil, SW_show);
end;

procedure TForm1.ToolButton5Click(Sender: TObject);
begin
ShellExecute (Form1.Handle, nil, 'helpp.chm', nil, nil, SW_show);
end;

end.

-----

unit Unit2;
```

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
Dialogs, StdCtrls;

type

TForm2 = class(TForm)

Label1: TLabel;

Label2: TLabel;

Label3: TLabel;

Button1: TButton;

Button2: TButton;

Edit2: TEdit;

Edit3: TEdit;

Edit1: TEdit;

Label4: TLabel;

Label5: TLabel;

Edit4: TEdit;

Edit5: TEdit;

Button3: TButton;

Button4: TButton;

procedure Button1Click(Sender: TObject);

procedure Button2Click(Sender: TObject);

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

procedure Button3Click(Sender: TObject);

procedure Button4Click(Sender: TObject);

procedure Edit1Change(Sender: TObject);

procedure FormActivate(Sender: TObject);

procedure FormCreate(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

end;

var

```
Form2: TForm2;
```

```
implementation
```

```
uses Unit3, Unit1;
```

```
{ $R *.dfm }
```

```
procedure TForm2.Button1Click(Sender: TObject);
```

```
var
```

```
    fam, im, ot, i :String;
```

```
    kol_hour, tarif :integer;
```

```
    k:integer;
```

```
begin
```

```
    If (form2.Edit1.text<>") { зберігає дані в таб.1 }
```

```
    and(form2.Edit2.text<>")
```

```
    and(form2.Edit3.text<>")
```

```
    and(form2.Edit4.text<>")
```

```
    and(form2.Edit5.text<>") then
```

```
        begin
```

```
            fam := Edit1.Text;
```

```
            im := Edit2.Text;
```

```
            ot := Edit3.Text;
```

```
            kol_hour := StrToInt(form2.Edit4.text);
```

```
            tarif := StrToInt(form2.Edit5.text);
```

```
            form1.StringGrid1.Cells[0,k] := fam;
```

```
            form1.StringGrid1.Cells[1,k] := im;
```

```
            form1.StringGrid1.Cells[2,k] := ot;
```

```
            form1.StringGrid1.Cells[3,k] := inttostr (kol_hour) ;
```

```
            form1.StringGrid1.Cells[4,k] := inttostr (tarif);
```

```
            form1.StringGrid1.RowCount:=form1.StringGrid1.RowCount+1;
```

```
            end else showmessage('Введено не всі дані');
```

```

OKBottomDlg.Showmodal;
    end;
procedure TForm2.Button2Click(Sender: TObject); { очищає поля у вікні вхідні дані }
begin
    edit1.text:="";
    edit2.text:="";
    edit3.text:="";
    edit4.text:="";
    edit5.text:="";
    form2.Close;
end;

procedure TForm2.FormClose(Sender: TObject; var Action: TCloseAction);
begin
form2.Edit1.Clear;
    form2.Edit2.Clear;
    form2.Edit3.Clear;
    form2.Edit4.Clear;
    form2.Edit5.Clear;
end;

procedure TForm2.Button3Click(Sender: TObject);
var
    fam, im, ot, i :String;
    kol_hour, tarif :integer;
    k:integer;
begin

k:=form1.StringGrid1.Row; { змінює дані в таб.1 }
im := Edit2.Text;
fam := Edit1.Text;
ot := Edit3.Text;
kol_hour := StrToInt(form2.Edit4.text);
tarif := StrToInt(form2.Edit5.text);

form1.StringGrid1.Cells[0,k] := fam;
form1.StringGrid1.Cells[1,k] := im;

```

```
form1.StringGrid1.Cells[2,k] := ot;
form1.StringGrid1.Cells[3,k] := inttostr (kol_hour) ;
form1.StringGrid1.Cells[4,k] := inttostr (tarif);
form2.Close;
end;
```

```
procedure TForm2.Button4Click(Sender: TObject); { видалляє дані з таб.1 }
```

```
var
  i,k:integer;
begin
  form1.stringgrid1.Rows[form1.stringgrid1.Row].Clear;
  k:=form1.stringgrid1.Row;
  for i:=form1.stringgrid1.row to form1.stringgrid1.rowcount do
  begin
    form1.StringGrid1.Rows[k]:= form1.StringGrid1.Rows[k + 1];
    k:=k+1;
  end;
  form1.StringGrid1.Rows[k + 1].Clear;
form2.Close;
end;
```

```
procedure TForm2.Edit1Change(Sender: TObject);
```

```
begin
  if (edit1.Text<>") and (edit2.Text<>") and (edit3.Text<>") and (edit4.Text<>") and
(edit5.Text<>")
  then
  begin
    form2.Button3.Enabled:=true;
    form2.Button4.Enabled:=true;
  end else
  begin
    form2.Button3.Enabled:=false;
    form2.Button4.Enabled:=false;
  end;
end;
```

```
procedure TForm2.FormActivate(Sender: TObject);
```

```
begin
form2.Button3.visible:=true;
form2.Button4.Visible:=true;
form2.Button3.Enabled:=false;
form2.Button4.Enabled:=false;
if (edit1.Text<>") and (edit2.Text<>") and (edit3.Text<>") and (edit4.Text<>") and
(edit5.Text<>")
then begin
form2.Button3.Enabled:=true;
form2.Button4.Enabled:=true;
end;
end;

procedure TForm2.FormCreate(Sender: TObject);
begin

end;

end.
unit Unit4;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls;

type
  TForm4 = class(TForm)
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
```


Label9: TLabel;
Label10: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Label17: TLabel;
Label18: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit9: TEdit;
Edit10: TEdit;
Edit12: TEdit;
Edit13: TEdit;
Edit14: TEdit;
Edit15: TEdit;
Edit16: TEdit;
Edit17: TEdit;
Edit18: TEdit;
Button1: TButton;
Button2: TButton;
Button3: TButton;
Button4: TButton;
Edit11: TEdit;
Edit8: TEdit;
Label11: TLabel;
Button5: TButton;
procedure Button2Click(Sender: TObject);
procedure Button3Click(Sender: TObject);
procedure Button1Click(Sender: TObject);
procedure Button4Click(Sender: TObject);

```
    procedure Button5Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
//  procedure FormCreate(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form4: TForm4; i:integer;

implementation

uses Unit1, Unit5;

{$R *.dfm}

procedure TForm4.Button2Click(Sender: TObject);
begin
    edit1.text:="";
    edit2.text:="";
    edit3.text:="";
    edit4.text:="";
    edit5.text:="";
    edit6.text:="";
    edit7.text:="";
    edit8.text:="";
    edit11.text:="";
    edit9.text:="";
    edit10.text:="";
    edit12.text:="";
    edit13.text:="";
    edit14.text:="";
    edit15.text:="";
    edit16.text:="";
    edit17.text:="";
```

```
edit18.text:='';
```

```
end;
```

```
procedure TForm4.Button3Click(Sender: TObject); { печать справка о сотруднике }  
begin
```

```
Button1.visible:=false;
```

```
Button2.visible:=false;
```

```
Button3.visible:=false;
```

```
Button4.visible:=false;
```

```
Button5.visible:=false;
```

```
label11.visible:=true;
```

```
label12.visible:=true;
```

```
label13.visible:=true;
```

```
label14.visible:=true;
```

```
label15.visible:=true;
```

```
label16.visible:=true;
```

```
label17.visible:=true;
```

```
label18.visible:=true;
```

```
label19.visible:=true;
```

```
label10.visible:=true;
```

```
label12.visible:=true;
```

```
label13.visible:=true;
```

```
label14.visible:=true;
```

```
label15.visible:=true;
```

```
label16.visible:=true;
```

```
label17.visible:=true;
```

```
label18.visible:=true;
```

```
Edit1.visible:=true;
```

```
Edit2.visible:=true;
```

```
Edit3.visible:=true;
```

```
Edit4.visible:=true;
```

```
Edit5.visible:=true;
```

```
Edit6.visible:=true;
Edit7.visible:=true;
Edit11.visible:=true;
Edit9.visible:=true;
Edit10.visible:=true;
Edit12.visible:=true;
Edit13.visible:=true;
Edit14.visible:=true;
Edit15.visible:=true;
Edit16.visible:=true;
Edit17.visible:=true;
Edit18.visible:=true;
```

```
Form4.Print;
```

```
Button1.visible:=true;
Button2.visible:=true;
Button3.visible:=true;
Button4.visible:=true;
Button5.visible:=true;
    Form4.Close;
end;
```

```
procedure TForm4.Button1Click(Sender: TObject);
begin
if (form1.StringGrid2.RowCount=2) and (form1.StringGrid2.Cells[0,1]='')
then i:=form1.StringGrid2.RowCount-1 else i:=form1.StringGrid2.RowCount;
with s do begin
    fam:=form4.Edit1.Text;
    name:=form4.Edit2.Text;
    otch:=form4.Edit3.Text;
    dataU:=form4.Edit4.Text;
    dolg:=form4.Edit5.Text;
    INN:=form4.Edit6.Text;
    pens:=form4.Edit7.Text;
    dr:=form4.Edit8.Text;
    spas:=form4.Edit9.Text;
```

```

    npas:=form4.Edit10.Text;
    vpas:=form4.Edit11.Text;
    city:=form4.Edit12.Text;
    str:=form4.Edit13.Text;
    dom:=form4.Edit14.Text;
    kv:=form4.Edit15.Text;
    sem:=form4.Edit17.Text;
    dety:=form4.Edit18.Text;
    tel:=form4.Edit16.Text;
end;
if
(form4.Edit1.text<>")and(form4.Edit2.text<>")and(form4.Edit3.text<>")and(form4.Edit4.text<>'
')and(form4.Edit5.text<>")
    and(form4.Edit6.text<>")and(form4.Edit7.text<>")and(form4.Edit11.text<>")and(form4.
Edit8.text<>")
    and(form4.Edit9.text<>")and(form4.Edit10.text<>") then
begin
    with form1.StringGrid2, s do begin
    cells[0,i]:=fam;
    cells[1,i]:=name;
    cells[2,i]:=otch;
    cells[3,i]:=datau;
    cells[4,i]:=dolg;
    cells[5,i]:=inn;
    cells[6,i]:=pens;
    cells[7,i]:=dr;
    cells[8,i]:=spas;
    cells[9,i]:=npas;
    cells[10,i]:=vpas;
    cells[11,i]:=city;
    cells[12,i]:=str;
    cells[13,i]:=dom;
    cells[14,i]:=kv;
    cells[15,i]:=tel;
    cells[16,i]:=sem;
    cells[17,i]:=dety;
end;

```

```

i:=i+1;if i>1 then form1.StringGrid2.RowCount:=i;
messedlg('Запис додана!',mtinformation,[mbOK],0);
{ form1.StatusBar1.Panels.Items[0].Text:='Кількість записів:
'+INTtostr(form1.Stringgrid1.RowCount-1); }
end else (messedlg('Введіть будь ласка дані!',mtInformation,[mbOK],1));
Form4.Close;
end;

procedure TForm4.Button4Click(Sender: TObject);
begin
with s do
fam:=form4.Edit1.Text;
name:=form4.Edit2.Text;
otch:=form4.Edit3.Text;
dataU:=form4.Edit4.Text;
dolg:=form4.Edit6.Text;
INN:=form4.Edit5.Text;
pens:=form4.Edit7.Text;
dr:=form4.Edit11.Text;
spas:=form4.Edit8.Text;
npas:=form4.Edit9.Text;
vpas:=form4.Edit10.Text;
city:=form4.Edit12.Text;
str:=form4.Edit13.Text;
dom:=form4.Edit14.Text;
kv:=form4.Edit15.Text;
sem:=form4.Edit17.Text;
dety:=form4.Edit18.Text;
tel:=form4.Edit16.Text;
end;
if
(form4.Edit1.text<>")and(form4.Edit2.text<>")and(form4.Edit3.text<>")and(form4.Edit4.text<>')
)and(form4.Edit5.text<>")
and(form4.Edit6.text<>")and(form4.Edit7.text<>")and(form4.Edit11.text<>")and(form4.
Edit8.text<>")
and(form4.Edit9.text<>")and(form4.Edit10.text<>") then
begin

```

```

i:=form1.StringGrid2.Row;
begin
with form1.stringgrid2,s do begin
cells[0,i]:=fam;
cells[1,i]:=name;
cells[2,i]:=otch;
cells[3,i]:=datau;
cells[4,i]:=dolg;
cells[5,i]:=inn;
cells[6,i]:=pens;
cells[7,i]:=dr;
cells[8,i]:=spas;
cells[9,i]:=npas;
cells[10,i]:=vpas;
cells[11,i]:=city;
cells[12,i]:=str;
cells[13,i]:=dom;
cells[14,i]:=kv;
cells[15,i]:=tel;
cells[16,i]:=sem;
cells[17,i]:=dety;
end;
i:=i+1;
end;
messagedlg('Запис змінена!',mtinformation,[mbOK],0);
end else (messagedlg('Введіть дані!',mtInformation,[mbOK],1));
Form4.Close;
end;

```

```

procedure TForm4.Button5Click(Sender: TObject); { видаляє дані з таб. довідка про
співробітників }

```

```

var
i,k:integer;

```

```
begin
  form1.stringgrid2.Rows[form1.stringgrid2.Row].Clear;
  k:=form1.stringgrid2.Row;
  for i:=form1.stringgrid2.row to form1.stringgrid2.rowcount do
    begin
      form1.StringGrid2.Rows[k]:= form1.StringGrid2.Rows[k + 1];
      k:=k+1;
    end;
  form1.StringGrid2.Rows[k + 1].Clear;
  Form4.Close;
end;
```

```
procedure TForm4.FormCreate(Sender: TObject);
begin

end;

end.
```