МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ
ФАКУЛЬТЕТ ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА МАТЕМАТИКИ ТА ПРОГРАМУВАННЯ

Пояснювальна записка

до дипломної роботи

________ бакалавр ________
(освітньо-кваліфікаційний рівень)

на тему «Створення программного додатку, що реалізує модель Лотки-Вольтерри»

Виконав: студент 4 курсу, групи ІТ-641
напряму підготовки 6.040302 „Інформатика”
спеціальності 7.04030201 „Інформатика”
_____________ Євтушенко А.С.
(підпис)

Керівник,
доцент, к.т.н._____________ Іванов В.Г.
(підпис)

Рецензент,
доцент, к.т.н._____________ Ковальов Ю.Г.
(підпис)

СЄВЄРОДОНЕЦЬК
2018 року

СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВОЛОДИМИРА ДАЛЯ

Факультет програмної інженерії
Кафедра програмування та математики
Освітньо-кваліфікаційний рівень бакалавр
Напрям підготовки 6.040302 „Інформатика”
Спеціальність 7.04030201 „Інформатика”

ЗАТВЕРДЖУЮ
Завідувач кафедри ПМ,
к.т.н., доцент
_____________ Лифар В.О.
«___» ____________ 2018 р.

ЗАВДАННЯ
НА ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Євтушенко Артуру Сергійовичу

1.Тема роботи «**Створення програмного додатку,що реалізує модель Лотки-Вольтерри**»
керівник роботи Віталій Геннадійович Іванов доцент, к.т.н.

затверджені наказом вищого навчального закладу від “__”____2018 р. №____
2. Строк подання студентом роботи 9.06.2018
3. Вихідні дані до роботи «Створення програмного додатку,що реалізує модель Лотки-Вольтерри»
4. Зміст роботи
Вступ
1. Аналітичний огляд
1.1. Основні поняття моделі «хижак-жертва»
1.2. Математичний опис моделі «хижак-жертва»
1.3. Моделі Вольтера
2. Інформаційна модель
2.1. Загальне уявлення моделі
2.2. Модель взаємозв’язку чисельності популяцій хижаків і жертв
2.3. Інформаційна модель обєкту
3. Опис програми
3.1. Вибір інструментальних засобів розробки
3.2. Опис програми
Висновки
Список літератури
Додаток
5. Перелік графічного матеріалу немає

6. Дата видачі завдання 4 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	10.03.18	
3	Проектування інфологічної моделі задачі що реалізується.	01.04.18	
5	Укладання та тестування програмного продукту	20.04.18	
6	Укладання, оформлення та погодження пояснювальної записки з керівником	10.05.18	
7	Укладання, оформлення та погодження з консультантом розділу «Охорона праці»	15.05.18	
7	Здача готової пояснювальної записки на кафедру	20.05.18	
8	Укладання доповіді і презентації	01.06.18	

Студент _____________ А.С. Євтушенко
(підпис) (ініціали, прізвище)

Керівник роботи _____________ В.Г. Іванов
(підпис) (ініціали, прізвище)

*Консультантом не може бути зазначено керівника дипломної роботи.

ЛИСТ ПОГОДЖЕННЯ І ОЦІНЮВАННЯ

дипломної роботи студента гр. ІТ-641 Євтушенко А.С.

Науковий керівник

Доцент, к.т.н. __________________ Іванов В.Г.

Оцінка наукового керівника: __________________

Рецензент Ковальов Юрій Григорович

ПІБ, місто роботи, посада

Оцінка рецензента: __________________________________

Кінцева оцінка за результатами захисту:

Голова ЕК

__________________ Лифар В.О.

підпис

РЕФЕРАТ

Текст – 60, рис. – 8, , табл. – 2, додатків – 1, літературних джерел – 11

У ході виконання даної роботи було проведено дослідження предметної області, вивчені основні джерела й особливості розробки моделі системи „хижак-жертва”.

Була розроблена програма, що реалізує алгоритм роботи моделі системи „хижак-жертва”

Одержана програма може бути використані у навчальному процесі при перевірки та демонстрації роботи моделі системи „хижак-жертва”.

Ключові слова: КОМ’’ПЮТЕРНА МОДЕЛЬ, СИСТЕМ „ХИЖАК-ЖЕРТВА”.

ЗМІСТ

ВСТУП

Актуальність досліджень. Математичне моделювання динаміки біологічних популяцій, взаємодія хижаків не тільки актуальна, а й надзвичайно цікава проблема. Існування об'єкта в складі екосистеми обумовлюється як закономірними внутрішніми процесами (репродукція, ріст, харчування, смертність та ін.), Так і випадковими зовнішніми явищами, які безпосередньо впливають на перебіг процесів життєдіяльності. Для опису процесів відтворення і смертності існує ряд аналітичних моделей Мальтуса (1798 р), Ферхюльста - Пірла (1838 р) і Ріккера (1954 р.) Найпростіша модель харчування була запропонована Лотки (1925 г.) і Вольтерра (1926, 1931 г.) і послужила поштовхом до розвитку сучасної математичної екології. Розроблено клас матричних і безперервних моделей, що враховують внутрішню вікову структуру популяції, найпростішої з яких є модель Леслі (1945 г.).

Системи звичайних диференціальних рівнянь Лотки - Вольтерри використовуються для моделювання взаємодії між конкуруючими біологічними видами.

Взаємодія популяцій, взаємодія хижаків і жертв, зміна їх чисельності з часом цікава з точки моделювання завдання. Аналітичні рішення подібних завдань були розглянуті нами на лабораторних роботах. Проблема аналітичних рішень полягає в тому, що ми можемо сказати скільки буде хижаків і скільки буде жертв в певний момент часу, але не можемо сказати як вони будуть розподілені по площі. У даній роботі розглядається модель взаємодії хижаків і жертв на площині.

Об’єкт досліджень: Алгоритми розв’язання моделі „хижак-жертва”.

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 методи розв’язання моделі „хижак-жертва”;

в) скласти програму розв’язання моделі „хижак-жертва”;

Методологічна та теоретична основа дослідження: методи розв’язання моделі „хижак-жертва”

Методи дослідження: програмування за допомогою візуального середовища Delphi.

Практичне значення отриманих результатів. Одержана програма може бути використані у навчальному процесі при перевірки та демонстрації роботи моделі „хижак-жертва”.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД

1.1. Основні поняття моделі «хижак-жертва» [1].

Спроби математичного моделювання динаміки як розписування окремих біологічних популяцій, і співтовариств, які включають взаємодіючі популяції різних видів, робилися давно. Один із перших моделей зростання ізольованій популяції було запропоновано ще 1798 р. Томасом Мальтусом:

$$\frac{dn}{dt} = \mu N \, , \qquad (1.1.1)$$

Ця модель задається такими параметрами:

N — чисельність популяції;

μ — різницю між коефіцієнтами народжуваності і смертності.

Інтегруючи це рівняння отримуємо:

$$N(t) = N(0)e^{\mu t} \, , \qquad (1.1.2)$$

Де N(0) – чисельність популяції в останній момент t = 0. Вочевидь, що модель Мальтуса при $\mu > 0$ дає нескінченний зростання чисельності, що немає у природних популяціях, де ресурси, які цей зростання, завжди обмежені. Зміни чисельності популяцій рослинного й тваринного світу не можна описувати простим закономМальтуса, на динаміку зростання впливають багато взаємозалежні причини – зокрема, розмноження кожного видусаморегулируется і видозмінюється те щоб цей вид зберігався у процесі еволюції. [1]

Математическим описом цих закономірностей займається математична екологія – наука про відносини рослинних і тварин організмів і утворених ними співтовариств між собою й навколишнім середовищем.

Найґрунтовніше дослідження моделей біологічних співтовариств, які включають у собі кілька популяцій різних видів, провели італійським математиком Вольтерра [2-3]

1.2 Математичний опис моделі «хижак-жертва»

Розглянемо модель трофічного взаємодії на кшталт «хижак-жертва», побудовану У.Вольтерром. Нехай є система, що складається з два види, з яких одна поїдає інший [4-6].

Розглянемо випадок, коли один видів є хижаком, а інший — жертвою, й вважатимемо, що хижак харчується лише жертвою. Приймемо таку просту гіпотезу:

$\varepsilon_1 - \gamma_1 N_2$ — коефіцієнт приросту жертви;

$\gamma_2 N_1 - \varepsilon_2$ — коефіцієнт приросту хижака;

N_1 — чисельність популяції жертви;

N_2 — чисельність популяції хижака;

ε_1 — коефіцієнт природний приріст жертви;

γ_1 — швидкість споживання жертви хижаком;

ε_2 — коефіцієнт смертності хижака за відсутності жертви;

γ_2 — коефіцієнт «переробки» хижаком біомаси жертви на власне біомасу.

Тоді динаміка чисельності популяцій у системі хижак — жертва описуватиметься системою диференційних рівнянь (1.2.1):

$$\begin{cases} \dfrac{dN_1}{dt} = N_1(\varepsilon_1 - \gamma_1 N_2) \\ \dfrac{dN_2}{dt} = N_2(-\varepsilon_2 + \gamma_2 N_1) \end{cases} \qquad (1.2.1)$$

де всі коефіцієнти позитивні та постійні.

Модель має рівноважний рішення (1.2.2):

$$\begin{cases} N_1 = \dfrac{\varepsilon_2}{\gamma_2} \\ N_2 = \dfrac{\varepsilon_1}{\gamma_1} \end{cases} \tag{1.2.2}$$

За моделлю (1.2.1) хижаків загалом тварин виражається формулою (1.2.3):

$$\frac{N_2}{N_1 + N_2} = \frac{\varepsilon_1 / \gamma_1}{\varepsilon_1 + \varepsilon_2 \gamma_1 / \gamma_2} \tag{1.2.3}$$

Аналіз стійкості стану рівноваги стосовно малим збурюванням показав, що така точка (1.2.2) є «нейтрально» стійкою (типу «центр»), тобто будь-які відхилення від рівноваги не загасають, але переводять систему в коливальний режим з амплітудою, залежною від величини обурення. Траєкторії системи на фазовій площині (N_1, N_2) мають вид замкнутих кривих, розташованих на різних відстанях від точки рівноваги (рис. 1).

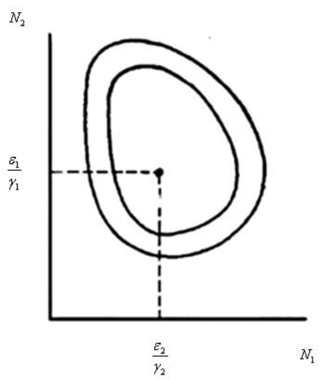


Рисунок 1.1 – Фазовий «портрет» класичної вольтерровой системи «хижак-жертва»

Розділивши перше рівняння системи (1.2.1) на друге, отримаємо диференціальне рівняння (1.2.4) для кривій на фазовій площині (N_1, N_2).

$$\frac{dN_1}{dN_2} = \frac{N_1}{-\varepsilon_2 + \gamma_2 N_1} \frac{\varepsilon_1 - \gamma_1 N_2}{N_2} \tag{1.2.4}$$

Інтегруючи дане рівняння отримаємо:

$$N_1^{-\varepsilon_2} N_2^{-\varepsilon_1} e^{\gamma_2 N_1} e^{\gamma_1 N_2} = C \tag{1.2.5}$$

де C - постійна інтеграції, де $C = N_1^{-\varepsilon_2} e^{\gamma_2 N_1} \Big/ N_2^{\varepsilon_1} e^{-\gamma_1 N_2}$

Нескладно показати, що рух крапки по фазовій площині відбуватиметься тільки в один бік. Для цього зручно зробити заміну функцій N_1 і N_2 перенісши початок координат на площині (N_1, N_2) у стаціонарну крапку (1.2.2) і ввівши потім полярні координати:

$$\begin{cases} N_1 = \dfrac{\varepsilon_2}{\gamma_2}(1+\rho\cos\varphi) > 0 \\ N_2 = \dfrac{\varepsilon_1}{\gamma_1}(1+\rho\sin\varphi) > 0 \end{cases} \tag{1.2.6}$$

У такому разі, підставивши значення системи (1.2.6) в систему (1.2.1), матимемо:

$$\begin{cases} \dfrac{d\rho}{dt}\cos\varphi - \dfrac{d\varphi}{dt}(\sin\varphi)\rho = -\varepsilon_1\rho\sin\varphi - \varepsilon_1\rho^2\sin\varphi\cos\varphi \\ \dfrac{d\rho}{dt}\sin\varphi + \dfrac{d\varphi}{dt}(\cos\varphi)\rho = \varepsilon_2\rho\cos\varphi + \varepsilon_2\rho^2\sin\varphi\cos\varphi \end{cases} \tag{1.2.7}$$

Помноживши перше рівняння на $\sin\varphi$ а друге — на $\cos\varphi$ і склавши їх, отримаємо:

$$\frac{d\varphi}{dt} = \varepsilon_2\cos^2\varphi(1+\rho\sin\varphi) + \varepsilon_1\sin^2\varphi(1+\rho\cos\varphi) \tag{1.2.8}$$

Після аналогічних перетворень алгебри отримаємо рівняння для ρ:

$$\frac{d\rho}{dt} = \sin\varphi(\cos\varphi)\rho(\varepsilon_2 - \varepsilon_1) - \rho^2\sin^2\varphi(\varepsilon_1\cos\varphi - \varepsilon_2\sin\varphi) \tag{1.2.9}$$

Величина $\frac{d\varphi}{dt}$ як видно з (1.2.9), завжди більше нуля. Таким чином $\frac{d\varphi}{dt}$ не міняє знаку, і обертання весь час йде в один бік.

Інтегруючи (1.2.9) знайдемо період:

$$T = \int_0^{2\pi} \frac{d\varphi}{\varepsilon_2\cos^2\varphi(1+\rho\sin\varphi) + \varepsilon_1\sin^2\varphi(1+\rho\cos\varphi)} \tag{1.2.10}$$

Коли ρ мало, то рівняння (1.2.8) і (1.2.9) переходять в рівняння еліпса. Період звернення в цьому випадку рівний:

$$T = \int_0^{2\pi} \frac{d\varphi}{\varepsilon_2 \cos^2 \varphi + \varepsilon_1 \sin^2 \varphi} = \frac{2\pi}{\sqrt{\varepsilon_1 \varepsilon_2}} \tag{1.2.11}$$

Виходячи з періодичності вирішень рівнянь (1.2.1), можна отримати деякі следствия. Представимо для цього (1.2.1) у вигляді:

$$\begin{cases} \dfrac{d \lg N_1}{dt} = \varepsilon_1 - \gamma_1 N_2 \\ \dfrac{d \lg N_2}{dt} = -\varepsilon_2 - \gamma_2 N_1 \end{cases} \tag{1.2.12}$$

і проінтегруємо по періоду:

$$\begin{cases} d \lg N_1 \Big|_{t_0}^{t_0+T} = \varepsilon_1 T - \gamma_1 \int\limits_{t_0}^{t_0+T} N_2 dt \\ d \lg N_2 \Big|_{t_0}^{t_0+T} = -\varepsilon_2 T + \gamma_2 \int\limits_{t_0}^{t_0+T} N_1 dt \end{cases} \tag{1.2.13}$$

Оскільки підстановки від $\lg N_1$ і $\lg N_2$ через періодичність звертаються в нуль, середні по періоду виявляються рівними стаціонарним станам (1.2.14):

$$\begin{cases} \dfrac{1}{T} \int\limits_{t_0}^{t_0+T} N_2 dt = \dfrac{\varepsilon_1}{\gamma_1} \\ \dfrac{1}{T} \int\limits_{t_0}^{t_0+T} N_1 dt = \dfrac{\varepsilon_2}{\gamma_2} \end{cases} \tag{1.2.14}$$

Прості рівняння моделі «хищник—жертва» (1.2.1) володіють поряд істотних недоліків. Так, в них передбачається необмеженість харчових ресурсів для жертви і необмежене зростання хижака, що противоречит експериментальним даним. Крім того, як видно з мал. 1, жодна з фазових кривих не виділена з погляду стійкості. За наявності навіть невеликих обурюючих дій траєкторія системи все далі йтиме від положення рівноваги, амплітуда коливань рости, і система достатньо швидко руйнуватиметься.

Не дивлячись на недоліки моделі (1.2.1), уявлення про принципово коливальний характер динаміки системи «хищник— жертва» набули широкого поширення в екології. Взаємодіями «хищник—жертва» пояснювали такі явища, як коливання чисельності хижих і мирних тварин в промислових зонах, коливання в популяціях риб, комах і так далі Насправді коливання чисельності можуть бути обумовлені і іншими причинами.

Припустимо, що в системі хижак — жертва відбувається штучне знищення особин обох видів, і розглянемо питання про те, яким чином знищення особин впливає на середні значення їх чисельності, якщо здійснюється пропорційно цій чисельності з коефіцієнтами пропорційності α_1 і α_2 відповідно для жертви і хижака. З урахуванням зроблених припущень систему рівнянь (1.2.1) перепишемо у вигляді:

$$\begin{cases} \dfrac{dN_1}{dt} = N_1(\varepsilon_1 - \alpha_1 - \gamma_1 N_2) & \alpha_1 > 0 \\ \dfrac{dN_2}{dt} = N_2(-\varepsilon_2 - \alpha_2 + \gamma_2 N_1) & \alpha_2 > 0 \end{cases} \tag{1.2.15}$$

Припустимо, що $\alpha_1 < \varepsilon_1$ тобто коефіцієнт винищування жертви менше коефіцієнта її природного приросту. В цьому випадку також спостерігатимуться періодичні коливання чисельності. Обчислимо середні значення численностей:

$$\begin{cases} \dfrac{1}{T}\int\limits_{t_0}^{t_0+T} N_2(t)dt = \dfrac{\varepsilon_1 - \alpha_1}{\gamma_1} \\ \dfrac{1}{T}\int\limits_{t_0}^{t_0+T} N_1(t)dt = \dfrac{\varepsilon_2 + \alpha_2}{\gamma_2} \end{cases} \tag{1.2.16}$$

Таким чином, якщо $\alpha_1 < \varepsilon_1$ то середня чисельність популяцій жертви зростає, а хижака — убуває.

Розглянемо випадок, коли коефіцієнт винищування жертви більше коефіцієнта її природного приросту, т. Е $\alpha_1 > \varepsilon_1$. В цьому випадку $\varepsilon_1 - \alpha_1 - \gamma_1 N_2 < 0$ при будь-яких $N_2 > 0$ і, отже, вирішення першого рівняння (1.2.15) обмежене зверху експоненціально убуваючою функцією $N_1(t) \le N_1 e^{(\varepsilon_1 - \alpha_1)}$ тобто $N_1(t) \to 0$ при $t \to \infty$.

Починаючи з деякого моменту часу t, при якому $\gamma_2 N_1(t) - \varepsilon_2 - \alpha_2 = 0$ вирішення другого рівняння (1.2.15) також починає убувати і при $t \to \infty$ прагне до нуля. Таким чином, у випадку $\alpha_1 > \varepsilon_1$ обидва види зникають.

1.3 Моделі Вольтера

Перші моделі В. Вольтерра, природно, не могли відображати всі сторони взаємодії в системі хижак — жертва, оскільки вони були значною мірою спрощені щодо реальних умов. Наприклад, якщо чисельність хижака N_2 рівна нулю, то з рівнянь (1.3.1) виходить, що чисельність жертви необмежено зростає, що не відповідає дійсності. Проте цінність цих моделей полягає саме в тому, що вони були основою, на якій швидкими темпами почала розвиватися математична екологія [7-8].

З'явилося велике число досліджень різних модифікацій системи хижак — жертва, де були побудовані більш загальні моделі, що враховують в тому або іншому ступені реальну ситуацію в природі.

У 1936 р. А.Н. Колмогоров запропонував використовувати для опису динаміки системи хижак — жертва наступну систему рівнянні:

$$\begin{cases} \dfrac{dN_1}{dt} = N_1 g_1(N_1, N_2) \\ \dfrac{dN_2}{dt} = N_2 g_2(N_1, N_2) \end{cases} \qquad (1.3.1)$$

де g_1 убуває із зростанням чисельності хижаків, а g_2 зростає із збільшенням чисельності жертви.

Ця система диференціальних рівнянь через її достатню спільність дозволяє добре враховувати реальну поведінку популяцій і разом з тим проводити якісний аналіз її рішень.

Пізніше в своїй роботі, Колмогоров досліджував детально менш загальну модель:

$$\begin{cases} \dfrac{dN_1}{dt} = g_1(N_1)N_1 - L(N_1)N_2 \\ \dfrac{dN_2}{dt} = g_2(N_1, N_2) \end{cases} \qquad (1.3.2)$$

Різні окремі випадки системи диференціальних рівнянь (1.3.2) досліджувалися багатьма авторами. У таблиці 1.3.1 приведені різні окремі випадки функцій $g(N_1)$ $L(N_1)$

Таблиця 1.3.1 - Різні моделі співтовариства «хижак-жертва»

$g(N_1)$	$L(N_1)$	$g_2(N_1,N_2)$	Автори
ε_1	$a_{12}N_1$	$-\varepsilon_2+a_{21}N_1$	Вольтерра-лотка
$\varepsilon_1-\alpha_1 N_1$	$a_{12}N_1$	$\varepsilon_2(1-e^{-\gamma N_1})$	Гаузе
ε_1	$a_{12}N_1$	$\varepsilon_2-a_{21}\frac{N_2}{N_1}$	Післоу
ε_1	$\frac{\alpha N_1}{1+\alpha h N_1}$	$\varepsilon_2-a_{21}\frac{N_2}{N_1}$	Холінг
ε_1	$b(1-e^{-\gamma N_1})$	$\varepsilon_2-a_{21}\frac{N_2}{N_1}$	Івльов
ε_1	$\frac{\alpha(N_1)N_1}{1+\alpha(N_1)hN_1}$	$\varepsilon_2-a_{21}\frac{N_2}{N_1}$	Рояма
$\varepsilon_1-\frac{N_1}{K_1}$	$\frac{\alpha N_1}{1+\alpha h N_1}$	$1-\frac{N_2}{K_1}N_1$	Шимазу
$\varepsilon_1-\alpha_1 N_1$	$a_{12}(1-e^{-\gamma N_1})$	$\varepsilon_2(1-a_{21}e^{-\mu N_1})$	Мей

РОЗДІЛ 2 ІНФОРМАЦІЙНА МОДЕЛЬ

2.1 Загальне уявлення моделі та моделювання

Модель – речова, знакова або уявна (мислена) система, що відтворює, імітує, відображає принципи внутрішньої організації або функціонування, певні властивості, ознаки чи характеристики об’єкта дослідження (оригіналу).

Значення терміна “модель” багатопланове:

– зразок, взірцевий примірник чогось;

– тип, марка конструкції;

– те, що є матеріалом, натурою для відтворення;

– зразок, з якого знімається форма для відливання в іншому матеріалі;

– комп’ютерна модель,

– розрахункова модель,

– теоретична модель (процесу, конструкції тощо).

Розрізняють фізичні, математичні та ін. моделі.

Наприклад, модель — опис об'єкта (предмета, явища або процесу) на якій-небудь формалізованій мові, складений з метою вивчення його властивостей. Такий опис особливо корисний у випадках, коли дослідження самого об'єкта ускладнене або фізично неможливе.

Найчастіше в ролі моделі виступає інший матеріальний або уявний об'єкт, що замінює в процесі дослідження об'єкт – оригінал. Таким чином, модель виступає як своєрідний інструмент для пізнання, який дослідник ставить між собою і об'єктом, і за допомогою якого вивчає об'єкт, що його цікавить.

Математична модель — це система математичних співвідношень, які описують досліджуваний процес або явище. Математична модель має важливе значення для таких наук, як: економіка, екологія, соціологія, фізика, хімія, механіка, інформатика, біологія та ін.

При одержанні математичної моделі використовують загальні закони природознавства, спеціальні закони конкретних наук, результати пасивних та активних експериментів, імітаційне моделювання за допомогою ЕОМ.

Математичні моделі дозволяють передбачити хід процесу, розрахувати цільову функцію (вихідні параметри процесу), керувати процесом, проектувати системи з бажаними характеристиками. Для їх створення можна використовувати будь які математичні засоби — мову диференційних або інтегральних рівнянь, теорії множин, абстрактної алгебри, математичну логіку, теорії ймовірностей, графи та інші.

Якщо відношення задаються аналітично, то їх можна розв'язати в замкнутому вигляді (явно) відносно шуканих змінних як функції від параметрів моделі, або в частково замкнутому вигляді (неявно), коли шукані змінні залежать від одного або багатьох параметрів моделі. До моделей цього класу належать диференційні, інтегральні, різницеві рівняння, ймовірнісні моделі, моделі математичного програмування та інші. Якщо не можна здобути точний розв'язок математичної моделі, використовуються чисельні (обчислювальні) методи або інші види моделювання.

У залежності від того, якими є параметри системи та зовнішні збурення математичні моделі можуть бути детермінованими та стохастичними. Детерміновані - пов’язані з дослідженням моделей з чітко заданими параметрами (задання початкових і граничних умов значень функцій на вході і т. д.). Стохастичні – пов’язані з випадковими значеннями. Останні мають особливо важливе значення при дослідженні і проектуванні великих систем зі складними зв’язками і властивостями, які важко врахувати.

Для розробки математичних моделей широко використовується диференційне числення, теорія множин, матриці і графи, а також планування експерименту. Відповідно розрізняють теоретико-множинні, матричні, топологічні та поліномні математичні моделі.

Приклади математичних моделей:

1. Модель Мальтуса – закон про пропорційну залежність між швидкістю росту і розміром популяції.

2. Система хижак-жертва (Вольтера-Лотки) – показує залежність між чисельністю хижаків та жертв.

3. Модель оптимальної поведінки покупця – виражає вибір покупця між множиною продуктів при обмеженому бюджеті.

Процес побудови, вивчення й використання математичних моделей називається математичним моделюванням. Це найзагальніший та найбільш використовуваний в науці, зокрема, в кібернетиці, метод досліджень. Це метод дослідження процесів або явищ шляхом створення їхніх математичних моделей і дослідження цих моделей. Він тісно поєднаний з такими категоріями, як абстракція, аналогія, гіпотеза тощо.

В основу методу покладено ідентичність форми рівнянь і однозначність співвідношень між змінними в рівняннях оригіналу і моделі, тобто, їх аналогії. Математичні моделі досліджуються, як правило, із допомогою аналогових обчислювальних машин, цифрових обчислювальних машин, комп'ютерів.

Математичне моделювання тією чи іншою мірою застосовують всі природничі і суспільні науки, що використовують математичний апарат для одержання спрощеного опису реальності за допомогою математичних понять. Воно дозволяє замінити реальний об’єкт його моделлю і потім вивчати останню.

В залежності від характеру процесів, що вивчаються, в системі всі види моделювання можуть бути розділені на аналітичні та комп’’ютерні (рис. 2.1)

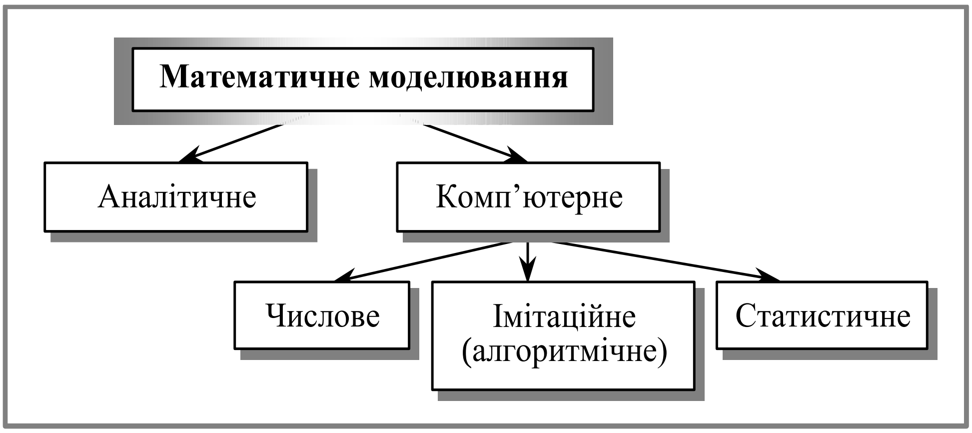


Рисунок 2.1 – Аналітичне та комп’ютерне моделювання

Для аналітичного моделювання характерним є те, що процеси функціонування елементів системи записують у вигляді деяких математичних співвідношень (алгебраїчних, інтегро-диференційних, кінцево-різницевих тощо) чи логічних умов.

Аналітична модель може досліджуватися такими методами:

1. Аналітичним, коли прагнуть у загальному вигляді отримати деякі залежності для шуканих характеристик;

2. Числовим;

3. Якісним, коли, не маючи явного розв’язку, все ж знаходять деякі властивості рішень.

Комп’ютерне моделювання характеризується тим, що математична модель системи (використовуючи основні співвідношення аналітичного моделювання) подається у вигляді деякого алгоритму та програми, придатної для її реалізації на комп’ютері, що дозволяє проводити з нею обчислювальні експерименти. Залежно від математичного інструментарію, що використовується в побудові моделі, та способу організації обчислювальних експериментів можна виокремити три взаємопов’язані види моделювання: числове, алгоритмічне (імітаційне) та статистичне.

За числового моделювання для побудови комп’ютерної моделі використовуються методи обчислювальної математики.

Алгоритмічне (імітаційне) моделювання (може бути як детермінованим, так і стохастичним) — це вид комп’ютерного моделювання, для якого характерним є відтворення на комп’ютері (імітація) процесу функціонування досліджуваної складної системи. Тут імітуються (з використанням аналітичних залежностей і моделей) елементарні явища, що становлять процес, зі збереженням їхньої логічної та семантичної структури, послідовності плину в часі, що дозволяє отримати нову інформацію про стан системи S у задані моменти часу.

Статистичне моделювання — це вид комп’ютерного моделювання, який дозволяє отримати статистичні дані відносно процесів у модельованій системі S.

Все частіше використовується комбіноване моделювання, системотвірним елементом якого є аналітичні моделі. У побудові та використанні комбінованих моделей попередньо проводять декомпозицію процесу функціонування моделі на складові елементи.

З розвитком математичних досліджень ускладнюється й проблема класифікації моделей, що використовуються. Разом із виникненням нових типів моделей (особливо змішаних типів) і нових ознак їх класифікації здійснюється процес інтеграції моделей різних типів у більш складні модельні конструкції.

Існує клас об'єктів, для яких з різних причин не розроблені аналітичні моделі або не розроблені методи розв'язування задач про такі моделі. В цьому випадку математична модель замінюється імітатором або імітаційною моделлю.

Імітаційна модель — логіко-математичний опис об'єкту, який може бути використаний для експериментування на комп'ютері в цілях проектування, аналізу і оцінки функціонування об'єкту. Імітаційне моделювання — це окремий випадок математичного моделювання, метод дослідження, заснований на тому, що система, яка вивчається, замінюється імітатором і з ним проводяться експерименти з метою отримання інформації про цю систему. Експериментування з імітатором називають імітацією (імітація — це збагнення суті явища, не вдаючись до експериментів на реальному об'єкті).

Імітаційне моделювання (машинна імітація) - особлива форма проведення експериментів на ЕОМ з математичними моделями, які з певним ступенем ймовірності описують закономірності функціонування реальних систем і об’єктів. Це метод, що дозволяє будувати моделі процесів, що описують, як ці процеси проходили б насправді. Таку модель можна «програти» в часі як для одного випробування, так і заданої їх кількості. При цьому результати визначатимуться випадковим характером процесів. За цими даними можна отримати достатньо стійку статистику.

Імітація як метод розв'язування нетривіальних задач отримала початковий розвиток у зв'язку із створенням ЕОМ в 1950х — 1960х роках.

Імітаційні (алгоритмічні) моделі можуть бути детермінованими і стохастичними. В останньому випадку за допомогою датчиків (генераторів) випадкових чисел імітується вплив (дія) невизначених і випадкових чинників. Такий метод імітаційного моделювання дістав назву методу статистичного моделювання (статистичних прогонів, чи методу Монте-Карло). На даний час цей метод вважають одним із найефективніших методів дослідження складних систем, а часто і єдиним практично доступним методом отримання нової інформації щодо поведінки гіпотетичної системи (на етапі її проектування).

Поняття про комп’ютерну модель і комп’ютерне моделювання.

Людина в будь якій діяльності постійно користується моделями. У дитинстві люди граються з ляльками, будиночками, машинами – зменшеними копіями реальних об'єктів. Дорослі також використовують моделі підчас спорудження будинку або пошиття костюму, створення ілюстрованого журналу або розрахунку польоту ракети.

Модель – це прообраз, опис або зображення якогось об'єкту.

Крім матеріальних об'єктів (іграшки, глобуса, макету будинку), існують абстрактні моделі: описи, формули, зображення, схеми, креслення, графіки тощо. За допомогою математичних формул описуються, скажімо, арифметичні операції. Хімічні формули допомагають уявити атомний склад хімічних речовин і реакцій, в які вони вступають.

Багато явищ і процесів різної природи описуються аналогічними співвідношеннями, наприклад, електроакустична аналогія, електро-, магнітно- і гідродинаміка. Тому для аналізу (розв'язання, розрахунку) математичних моделей необхідно володіти розвиненим математичним апаратом, що охоплює всі види типових завдань прикладної математики. Стосовно використання комп'ютерів основним етапом розрахунку математичних моделей є їх алгоритмізація, тобто розробка структури алгоритму, представленого у вигляді блок-схеми, граф-схеми або програмно-реалізованого з використанням принципів структурного програмування.

Комп'ютерне моделювання – метод розв'язування задачі аналізу або синтезу складної системи, що ґрунтується на використанні її комп'ютерної моделі. Сутність комп'ютерного моделювання полягає у відшуканні кількісних і якісних результатів із залученням наявної моделі.

Комп'ютерна модель складної системи має якомога повніше відбивати всі основні фактори й взаємозв'язки, що характеризують реальні ситуації, критерії та обмеження. До того ж модель має бути настільки універсальною (щоб охоплювати якнайширше коло близьких за призначенням об'єктів) настільки й простою (щоб сприяти виконанню необхідних досліджень із мінімальними витратами).

Комп'ютерна модель:

- структурно-функціональна - створений за допомогою комп'ютера умовний (віртуальний) образ об'єкта, процесу, явища, який якісно і кількісно відбиває структуру, його внутрішні властивості та зв'язки між елементами об'єкта, що моделюється;

- імітаційна - окремі програми, їх сукупності, програмні комплекси, за допомогою яких можна шляхом здійснення послідовності обчислень і графічного відображення їх результатів відтворювати (імітувати) процеси функціонування об'єкта при умові дії на об'єкт різних факторів.

Основні етапи комп'ютерного моделювання.

- постановка задачі;
- опис;
- мотивація;
- попередній аналіз;
- розробка моделі;
- виділення суттєвих факторів;
- створення алгоритму;
- програмування;
- комп'ютерний експеримент;
- тестування моделі;
- налагодження моделі;
- розрахунок моделі при різноманітних вхідних даних
- аналіз результатів;
- обчислювальний експеримент.

Комп'ютерний напрям моделювання в науці отримав назву обчислювального експерименту.

Обчислювальний експеримент (computationalexperiment) – це методологія дослідження, засновані на вивченні математичної (інформаційної) моделі за допомогою логіко-математичних алгоритмів на комп'ютері.

Комп'ютерне моделювання (обчислювальний експеримент) має істотні переваги перед натурним експериментом.

По-перше, не потрібно проводити експеримент на реальних фізичних, економічних чи інших об'єктах, тому затрати на різні комп'ютерні експерименти набагато менші, ніж на натурні експерименти. Масштаби експериментів можна вибрати на свій розсуд, при цьому є можливість проведення багатократних дослідів із поступовими змінами вхідних даних задачі.

По-друге, проведення реальних експериментів у деяких галузях науки небезпечне (екологія, ядерна фізика) або неможливе (астрофізика).

По-третє, у процесі побудови математичних моделей для проведення обчислювального експерименту і під час її дослідження можна проаналізувати і зрозуміти характеристики досліджуваного об'єкта.

Галузі застосування комп’ютерного моделювання. Приклади.

Фізика. При дослідженні енергетичної проблеми; космічна техніка (розрахунок траєкторії руху планет, комет і т.д.); технологічні процеси (технології створення матеріалів за даними властивостями); екологічні проблеми; гео – і астрофізичні явища (моделювання клімату, землетрусів, цунамі, процесу розвитку зірок і сонячної активності, проблеми розвитку Всесвіту).

Хімія. Розрахунок хімічних реакцій; дослідження хімічних реакцій на мокро і макрорівні.

Біологія. Вивчення фундаментальних проблем генетики; розвиток біотехнологій.

Екологія, соціологія, економіка, математика.

2.2 Модель взаємозв’язку чисельності популяцій хижаків і жертв

Модель взаємозв’язку чисельності популяцій хижаків і жертв описується системою рівнянь Лотки-Вольтера:

$$\begin{cases} \dfrac{dx_1}{dt} = +a_1 x_1 - b_1 x_1 x_2, \\ \dfrac{dx_2}{dt} = -a_2 x_2 + b_2 x_1 x_2. \end{cases} \qquad (2.2.1)$$

Нехай задано значення параметрів a1,a2,b1,b2 і початкові умови – значення перемінних при t = 0.

Треба провести чисельне інтегрування системи методом Рунге-Кути, одержати таблицю даних і побудувати графік залежності чисельності кожної популяції від часу і фазову траєкторію.

Проводячи зміни початкових умов, встановити, при якому співвідношенні початкових кількостей популяцій спостерігається нестійкість циклу (чисельність у точці мінімуму дорівнює 0 або менше 0)

Дослідити ускладнену модель Лотки-Вольтера (2.2.2):

$$\begin{cases} \dfrac{dx_1}{dt} = +a_1 x_1^{C_1} - b_1 x_1 x_2, \\ \dfrac{dx_2}{dt} = -a_2 x_2^{C_2} + b_2 x_1 x_2. \end{cases} \tag{2.2.2}$$

Дослідити, яким чином якісно впливає величина C_1 та C_2 , наприклад, в діапазоні змін від 0,8 до 1,2. Визначити, чи є наявною точка біфуркації. Якщо вона є – визначити її тип.

Популяцією називають будь-яку групу організмів одного виду, що протягом великого числа поколінь займає певний простір і може функціонувати (розмножуватись і розвиватись) за певних умов навколишнього природного середовища. Популяція — це елементарна одиниця еволюційного процесу і форма існування виду, це множина (група) організмів, що мають властивість саморазмноження.

Популяції мають властивості:

а) притаманні як самій популяції (щільність, народжуваність, смертність, вікова структура, генетика, пристосованість),

б) притаманні окремим організмам (життєвий цикл, репродукція та ін.).

Під щільністю популяції розуміють число особин (організмів) на одиницю площі або об'єму

Всі популяції дуже мінливі, тобто їх чисельність (щільність) з часом зазнає змін. Навіть у тому випадку, коли угруповання (популяція) та екосистема вбачаються незмінними, щільність, народжуваність, рівень виживання (смертність), вікова структура, швидкість росту та інші характеристики, як правило, змінюються залежно від зміни сезону, клімату та інших факторів навколишнього середовища.

Екологів найбільше цікавить те, з якою швидкістю відбувається зміна чисельності популяції або її певної вікової групи.

а) Середня швидкість визначається як. відношення зміни кількості організмів (особин) ΔN до проміжку часу Δt, протягом якого відбулася ця зміна (збільшення або зменшення, в останньому випадку ΔN береться і знаком мінус).

б) Миттєва (істинна) швидкість розмноження популяції $V = \frac{dN}{dt}$ або просто швидкість розмноження організмів у деякий момент часу t визначається як межа середньої швидкості за умови необмеженого зменшення проміжку часу Δt, тобто:

$$V = \frac{dN}{dt} = \lim_{\Delta t \to 0} \frac{\Delta N}{\Delta t}. \tag{2.2.3}$$

Таким чином, швидкість розмноження популяції є похідною за часом від кількості особин, що розглядається як функція часу (віку особин або віку популяції), тобто N = f(t), V = f'(t).

Багато явищ природи, такі як «цвітіння» води у водоймищах, спалах (вибух) чисельності шкідників, розмноження бактерій та інших мікроорганізмів, супроводжується зростанням чисельності організмів згідно з експоненціальним законом. Відзначимо також, що експоненціальний характер спостерігається для багатьох інших процесів і явищ природи, таких як поглинання світла, мономолекулярні хімічні реакції, зростання складних процентів та ін.

У випадку взаємовідносин типу «хижацтво» динаміка популяцій опишеться такою системою диференціальних рівнянь:

$$\begin{cases} \dfrac{dN_1}{dt} = \varepsilon_1 N_1 - \dfrac{\varepsilon_1}{h_1} N_1^2 + \gamma_1 N_1 N_2; \\ \dfrac{dN_2}{dt} = \varepsilon_2 N_2 - \dfrac{\varepsilon_2}{h2} N_2^2 - \gamma_2 N_1 N_2 \end{cases} \tag{2.2.4}$$

Для описання біоценозу двох популяцій з чисельністю N1(t) і N2(t) Вольтерра запропонував систему диференціальних рівнянь

$$\frac{dN_i}{dt} = N_i \cdot f_i(N_1, N_2), \quad i = 1,2 \qquad (2.2.5)$$

Система диференціальних рівнянь має вид:

$$\frac{dN_1}{dt} = \varepsilon_1 N_1 - \gamma_1 N_1 N_2, \qquad (2.2.6)$$

$$\frac{dN_2}{dt} = -\varepsilon_2 N_2 - \gamma_2 N_1 N_2, \qquad (2.2.7)$$

$\{\varepsilon_1, \varepsilon 2\} > 0$; $\{\gamma 1, \gamma 2\} > 0$

У рівнянні (2.2.6) для жертви:

а) Перший член відображає звичайний природний приріст чисельності жертви внаслідок перевищення народжуваності над смертністю.

б) Другій член відповідає зменшенню приросту жертви внаслідок знищення її хижаком. Це зменшення пропорційно кількості зустрічей між хижаком і жертвою.

У рівняння для хижака:

а) Перший член відображає природне зменшення кількості хижаків.

б) Другий член пояснює, що розмноження хижаків пропорційно кількості зустрічей з жертвою, тобто, кількістю їжі для хижаків.

Система рівнянь (2.2.6), (2.2.7) має два стаціонарних рішення:

$$\varepsilon_1 N_1 - \gamma_1 N_1 N_2 = 0, \qquad (2.2.8)$$

$$-\varepsilon_2 N_2 - \gamma_2 N_1 N_2 = 0. \qquad (2.2.9)$$

Звідсі – наявні два рішення:

а) $(x_1, x_2)_1 = 0$;

б) $(x_1, x_2)_2 = \left(\frac{\varepsilon_2}{\gamma_2}; \frac{\varepsilon_1}{\gamma_2}\right)$

Виконаємо перетворення, подібні тим, що описані у 2.2.2 б), одержимо , що компоненти будь-якого рішення системи (2.2.8), (2.2.9) з позитивними початковими даними задовільність умові:

$$N_1^{\varepsilon_2} N_2^{\varepsilon_1} e^{-\gamma_2 N_1} e^{-\gamma_1 N2} = c. \qquad (2.2.10)$$

Рівняння (2.2.10) при певному значенні С є рівнянням деякої замкненої траєкторії (кривої), що містить , як внутрішню, точку М = $\left(\frac{\varepsilon_2}{\gamma_2}; \frac{\varepsilon_1}{\gamma_2}\right)$ (рис. 2.2.1). Таким чином, будь-яка траєкторія у позитивному квадранті (крім точки М являє собою цикл. Це означає, що рішення є періодичним. Тобто хижак і жертва можуть співіснувати будь-кільки довго, змінюючи свою чисельність за періодичним законом.

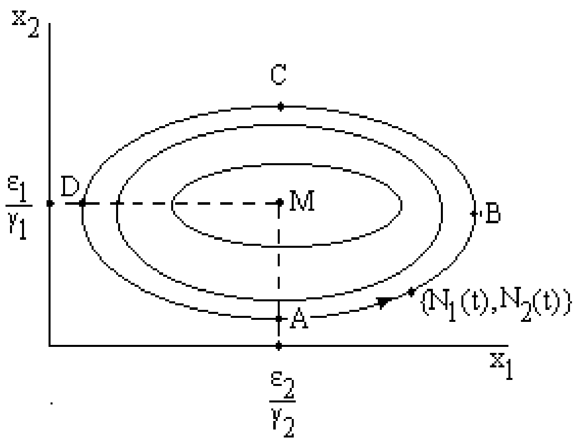


Рисунок 2.2.1 – Траєкторія

Інтерпретація співіснування:

а) збільшуючи свою чисельність, жертва збільшує ресурси живлення для хижака. Тому на ділянці АВ зростає чисельність не тільки жертви але й хижака тощо. До того моменту, коли точка (N_1(t),N_2(t)) попадає у точку В, хижаків становиться так багато, що вони знищують жертву швидше, ніж вона відтворює себе. Тому, починаючи з точки В, чисельність N_1(t) починає убувати. Чисельність N_2(t) при цьому все ще зростає. Однак з часом нестача їжі становиться настільки суттєвою, що, починаючи з точки С – точки максимальної кількості хижака, остання падає. Так продовжується до точки D, де хижаків залишається настільки мало, що жертва встигає поповнити свою чисельність і рухається по дузі DA. Після попадання у точку А цикл повторюється.

2.3 Інформаційна модель обєкту

Взаємодія популяцій, взаємодія хижаків і жертв, зміна їх чисельності з часом цікаве з точки моделювання завдання. Проблема аналітичних рішень полягає в тому, що ми можемо сказати, скільки буде хижаків і скільки буде жертв в певний момент часу, але не можемо сказати як вони будуть розподілені за площею. У даній роботі розглядається модель взаємодії хижаків і жертв на площині.

Спрощуючі припущення.

Спробуємо зіставити жертві і хижакові деякий алгоритм (примітивний інтелект), щоб взаємодія виглядала якомога правдоподібнішою.

1. Жертви і хижаки за одну ітерацію ходять на 1 клітку.
2. Жертви знаходяться на одній клітці поки на ній достатньо їжі.
3. Якщо їжа закінчується, то жертва випадковим чином переходить на сусідню вільну клітку.
4. Жертви утримують зайняту територію (тобто інша жертва не може встати на ту ж клітку).
5. Якщо поряд немає вільних кліток, то жертва залишається на поточній клітці.
6. Хижак бачить на відстань однієї клітки і якщо поряд є жертви, то випадковим чином з'їдає одну.
7. Якщо поряд немає жертв, то випадковим чином встає на вільну поряд клітку.
8. Хижак захищає зайняту територію (тобто інший хижак не може встати на зайняту родичем клітку)
9. Якщо їжу не досить, то жертви і хижаки починають голодувати (аж до смерті).
10. З'ївши жертву, хижак повністю відновлює свої сили, а жертва відновлює свої сили тільки на 1 умовну одиницю.
11. Хижаки і жертви розмножуються після закінчення певного часу (за умови, що з предыдыдуших пологів пройшов певний термін і той, хто народжує- ситий (голод вгамований на 100%).)
12. Хижаки і жертви не розмножуються, якщо всі прилеглі поля зайняті.

Відповідно до вищезгаданих припущень була побудована модель, яка дає візуальне уявлення про взаємодію хижаків і жертв, що відбувається. У даній моделі можна міняти ряд параметрів (на жаль, тільки при компіляції):

1. Голод жертв (скільки ходів може без їжі).

2. Голод хижаків (скільки ходів може без їжі).
3. Скільки трави з'їдає за хід жертва.
4. Скільки трави зростає за хід на клітці (швидкість відновлення ресурсів).
5. Через який час після пологів жертва може знову народжувати.
6. Через який час після пологів хижак може знову народжувати.

РОЗДІЛ 3 ОПИС ПРОГРАМИ

3.1 Вибір інструментальних засобів розробки

Нині існує чимало якісних засобів розробки програмного забезпечення. Розглянемо особливості деяких з них.

В даному проекті використовується Borland Delphi.

Borland Delphi — це інтегроване середовище швидкої розробки програмного забезпечення в Microsoft Windows. Delphi являє собою актуальну і легку у використанні програму, яка необхідна для генерації автономних програм графічного інтерфейсу або 32-бітових консольних додатків - програм, які існують поза рамками GUI, замість цього, відповідно до так званим «DOS вікна».

Delphi є першою мовою програмування, що забезпечує знищення бар'єру між додатками комплексного і спрощеного характеру у використанні і низькорівневими бітовими програмними засобами.

Створюючи GUI-додатки за допомогою Delphi, трансльований мова програмування існує в рамках RAD-середовища (мова Паскаль). Delphi включає в себе такі компоненти, як основні елементи графічного інтерфейсу користувача системи Windows, які представлені у вигляді екранного бланка, кнопок і ін. Це означає, що користувачеві не потрібно організовувати написання кодування в разі приєднання цих елементів до визначеному додатком. Користувач просто розробляє їх в програмі малювання. Можливо також використання керуючих елементів ActiveX з метою створення таких спеціальних програм, як веб-браузери. Delphi дозволяє користувачеві розробляти весь інтерфейс візуально, а також швидко складати код.

Delphi існує в безлічі конфігурацій, які використовуються як у відомчих, так і у виробничих установах. За допомогою Delphi користувач може написати програму в рамках ОС Windows на багато швидше і простіше, ніж це коли-небудь було можливо.

В основі Delphi-середовища лежить мова програмування Паскаль. Delphi має можливість використання безлічі баз даних. Прикладами можуть бути локальні бази даних - Paradox, Dbase, мережеві серверні бази даних SQL - InterBase, SysBase.

Незважаючи на те, що написання комп'ютерної програми за допомогою Delpi представляється полегшеним заходом, не варто забувати, що подібне програмний засіб вимагає упевнених знань системи Windows.

3.2 Опис програми

Програма для моделювання проживання зайців і вовків (Хижака і Жертви) на одній території, протягом якого-небудь часу.

Лістинг програми показаний в додатку А.

Блок-схема програми:

Середовищем їхнього життя є матриця, розмірністю 10 на 10. Тимчасовим кроком є одна доба (в програмі - один цикл).

За добу тварина здатна переміститися на одну клітку в будь-якому напрямку.

Якщо заєць зустрічається з вовком, то він гине;

Якщо зустрічаються різностатеві тварини, то з'являється ще одне;

Якщо вовк не знайшов собі зайця ,то він гине від голоду;

При відкритті програми відкривається головне вікно.

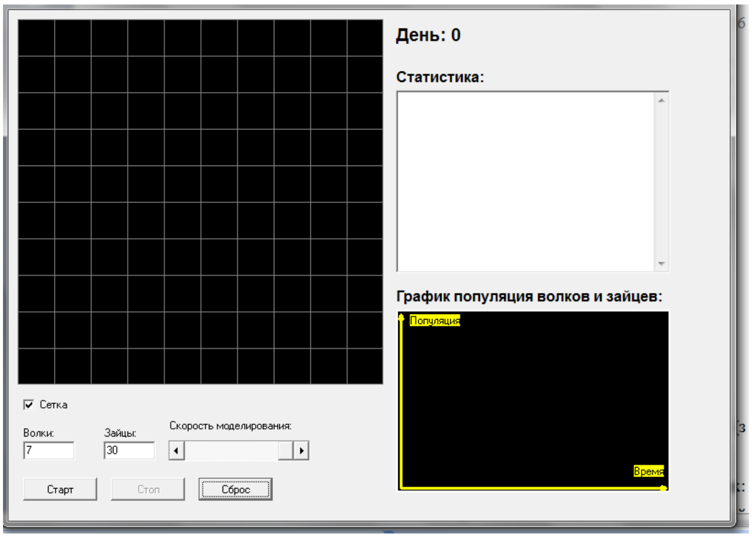


Рисунок 3.2.1 – Вікно програми

У таблиці клітини можуть бути в трьох станах: порожня (чорна клітинка), заєць (зелений колір клітинки), вовк (червоний колір клітинки) (рис.3.2.2).

Для зручності аналізу одержуваної інформації в програму додана статистика, у якій виводиться кількість днів та тварин.

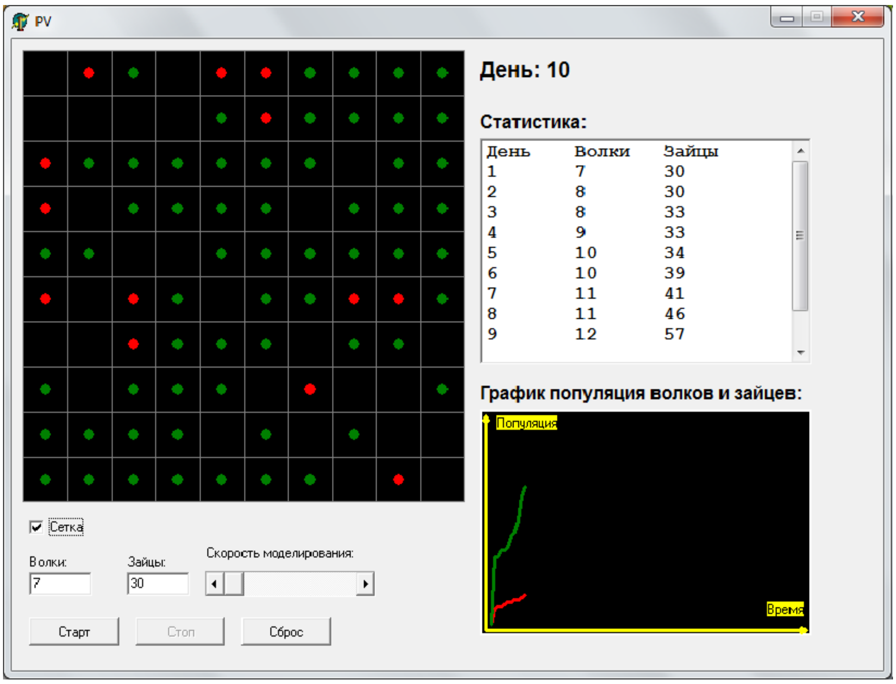


Рисунок 3.2.2 – Программа у процессі

Поведінка системи характеризується наступними параметрами:

Особина може переміститися на будь-яку клітку, що має з даною спільну сторону. Напрямок вибирається випадково з можливих вільних.

Особина може залишити потомство в тій клітці, з якої вона перемістилася. Потомство з'являється у будь-який час, параметр не існує.

Якщо особина є хижою, то вона може поглинути свою жертву, при цьому переміщаючись на місце жертви.

Особина живе необмежену кількість часу, параметра також не існує.

Якщо «хижа» особина не знаходить собі їжі протягом певного часу (час голодної смерті), то вона гине.

Час голодної смерті не є параметром.

Також є графік, на якому зоображенна життєва діяльність(популяція) обох видів тварин, червоним кольором позначенні вовки, зеленим зайці. Після багатох спостережень було помітно, що коли зелена лінія перетинається з червоною, зелена лінія(зайці) стрімко падає вниз, що означає загибель представників цього виду. (рис. 3.2.3)

У програмі існує клавіши «Старт», «Стоп», «Сброс». Кожна з них має свій фунціонал. Клавіша «Старт» починає відлік днів та розпочинає роботу графіка. Клавіша «Стоп» припиняє ведіння статистики та графіку, але не видаляє їх. Клавіша «Сброс» має дві функції: припиняє ведіння статистики та видаляє її. (рис. 3.2.4)

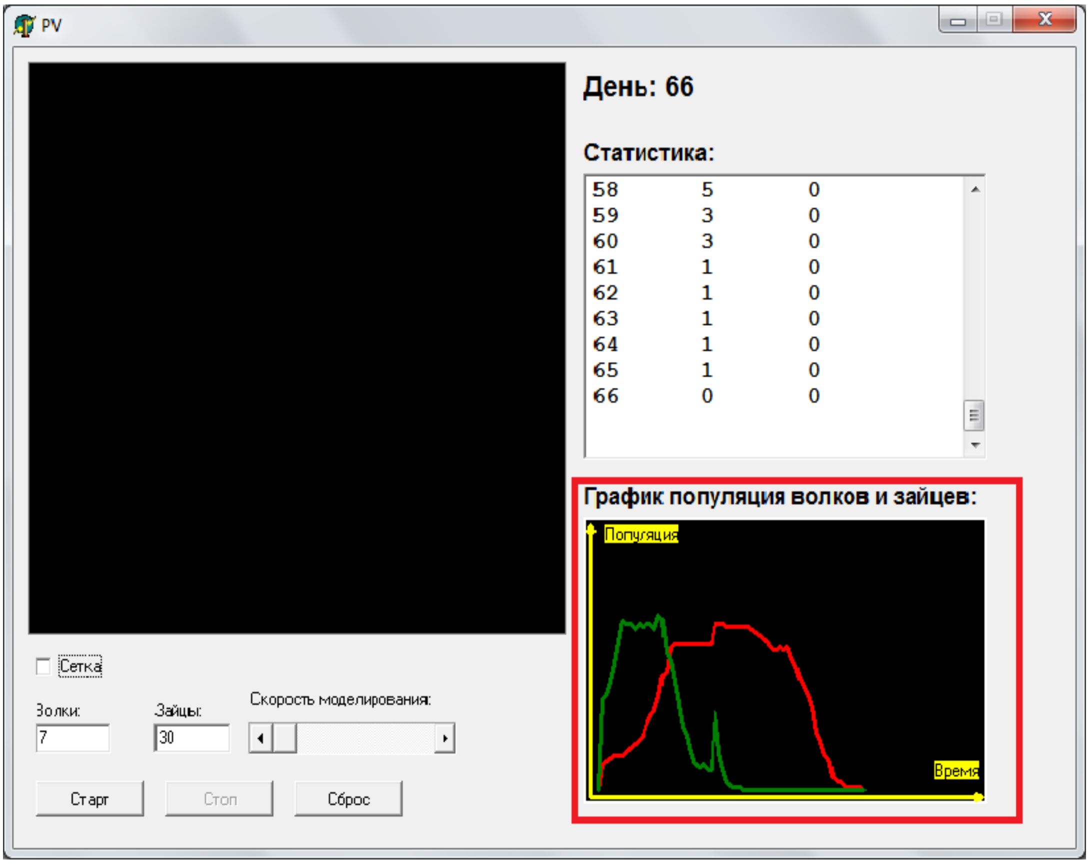
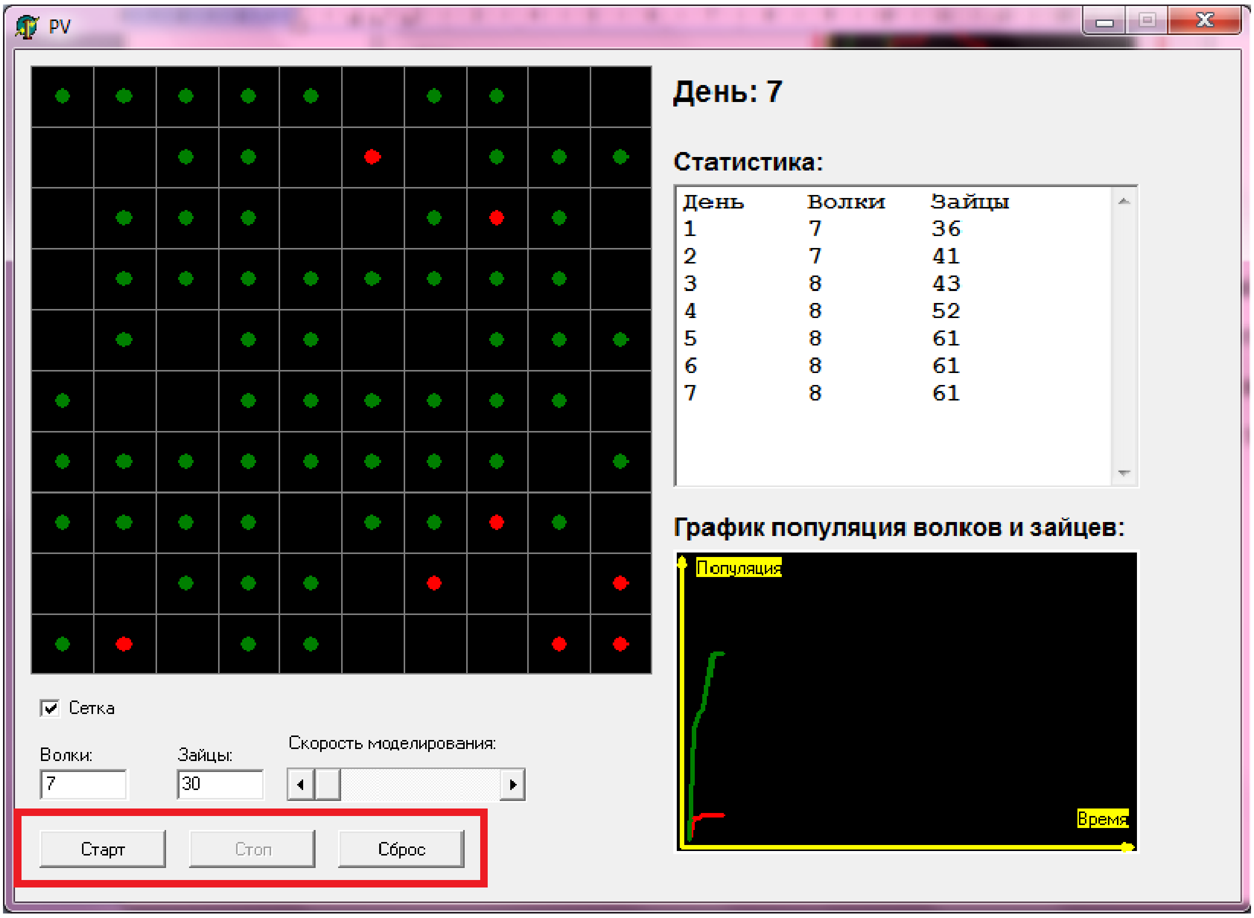


Рисунок 3.2.4 – Клавіші «Старт», «Стоп», «Сброс».



Також можна визначити кількість хижаків та жертв, також можна визначити швидкість моделювання. Це дозволяє визначати швидкість подій у моделі. Можно зробити як максимальну швидкість, так і минімальну, щоб детальніше роздивитись увесь процесс.

Для зручності у користуванні можно зробити сітку на полі, або зробити поле без неї. (рис. 3.2.5)

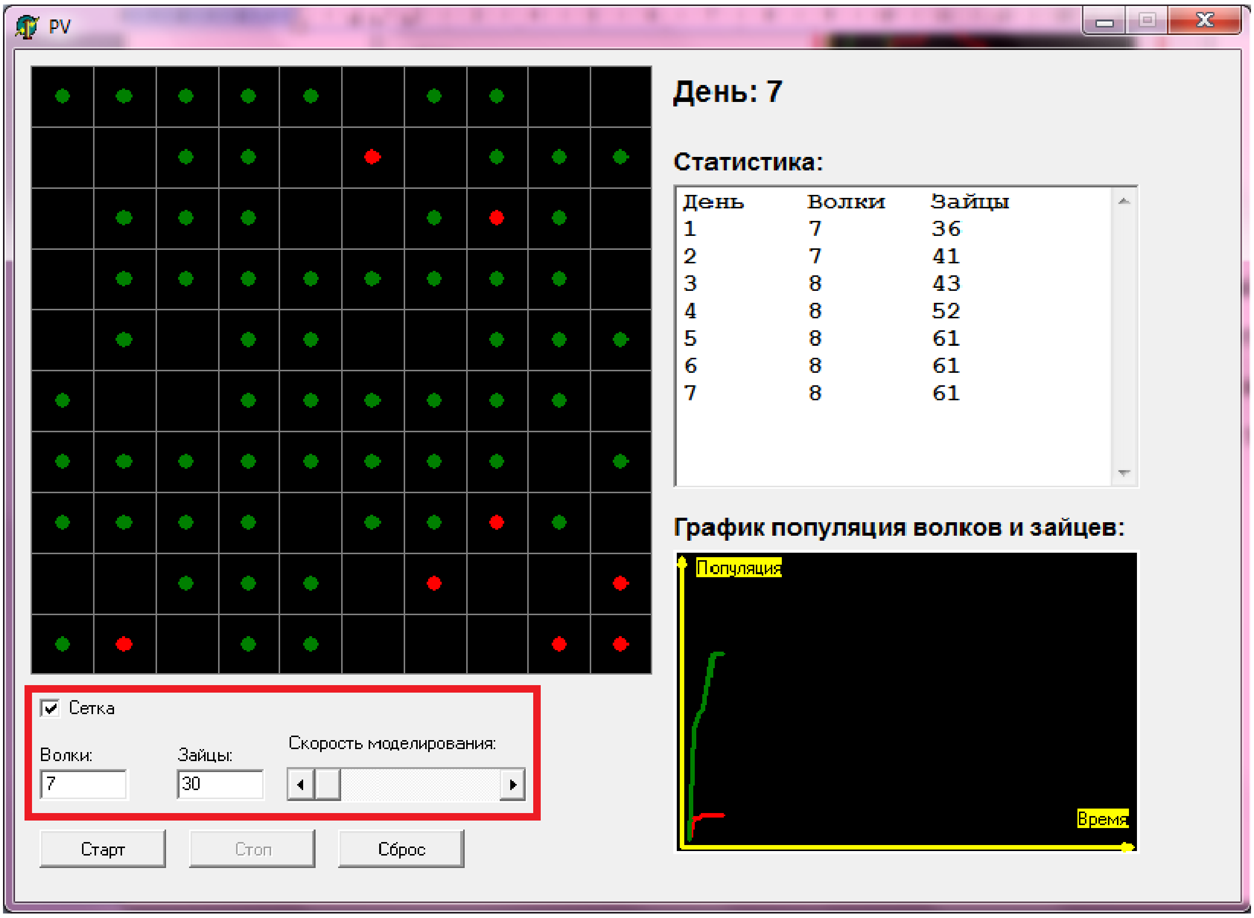


Рисунок 3.2.5 – Вигляд программи.

Після проведення експерименту з данною программою, завдяки підручних засобів ми визначили цикл життя хижакі та жертв. Спочатку було 7 вовків(хижаків) та 30 зайців(жертв). Увесь цикл життя тривав 56 днів. На перший же день популяція вовків збільшилась на 1, а зайців – на два. На 7, 9, 10-12 день було найбільше зайців, а саме 61, на 18 день їх становилося все меньше. З 21 по 27 день популяція жертв стрімко спадала вниз, за день зникало більше 10 представників. На 36 жертв не стало. З 24 по 38 день було найбільше вовків, а саме 51. На 39 день популяція почала помирати голодною смертю, бо зайців вже не було. На останній день, 56, не стало вовків. Після цього ведіння статистки припинилося.

Складаємо таблицю:

	На початку експерименту	У середнині	На кінці експерименту	Середня кількість
Хижаки(Вовки)	7	51	0	28
Жертви(Зайці)	30	4	0	24

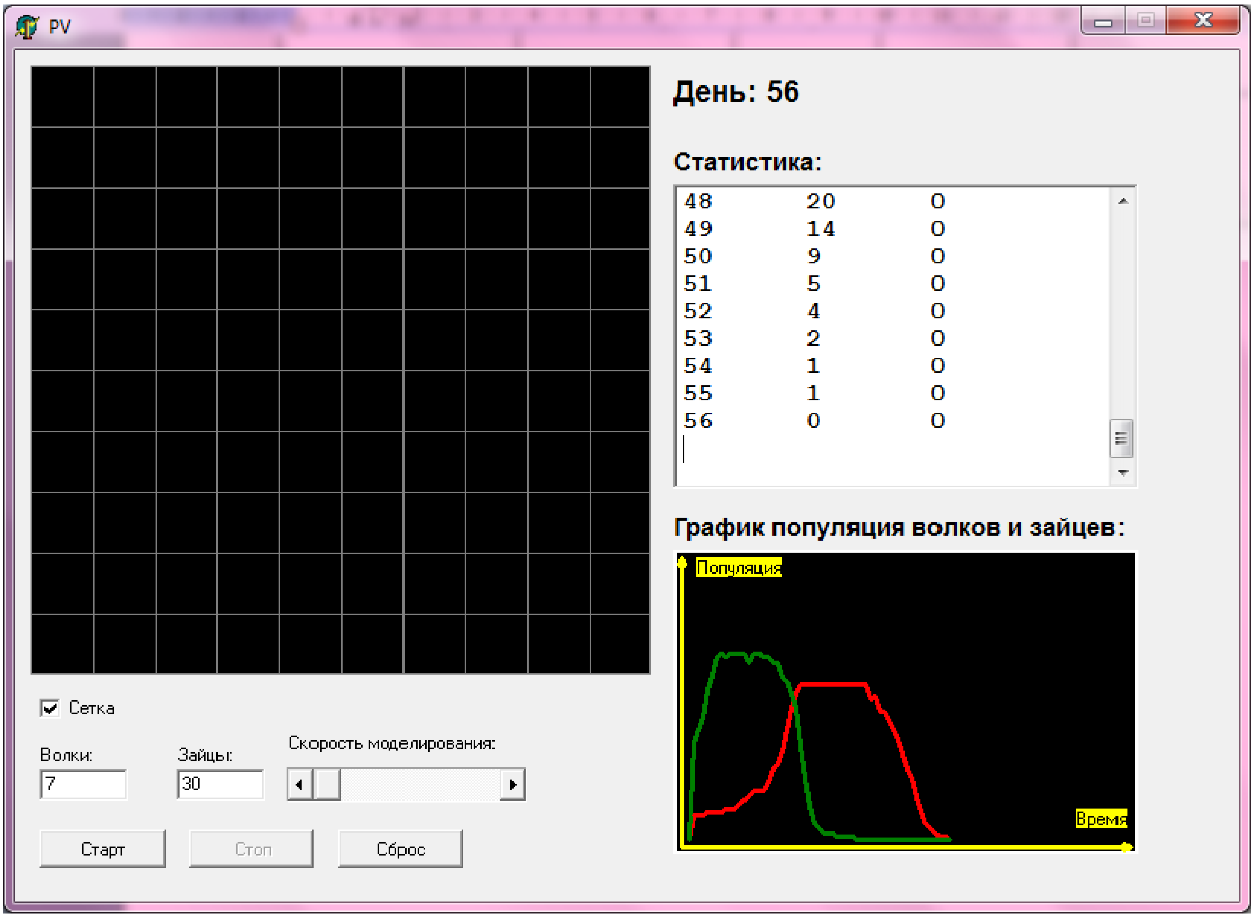


Рисунок 3.2.6 - Кінцевий результат.

ВИСНОВКИ

1. Зроблено аналіз предметної області, вивчені основні джерела й особливості розробки комп'ютерної моделі „хижак-жертва”

2. У роботі була розроблена програма, що реалізує модель „хижак-жертва”.

3. Одержана програма може бути використані у навчальному процесі при перевірки та демонстрації роботи моделі „хижак-жертва”.

СПИСОК ЛІТЕРАТУРИ

1. Томашевський В.М. Моделювання систем [Текст]. – К.: Видавнича група ВН, 2005. – 352 с.

2. Войлов ЮМ. Элементы теории систем и системного анализа [Текст]. – Луганск: ВНУ им. В.Даля, 2002 – 310 с.

3. Кузин Л.Т. Основы кибернетики: В2-х тт. Т. 2. Основы кибернетических моделей [Текст]. Учеб. Пособие для вузов. – М.: Энергия,1979. – 584 с.

4. Кендэл М. Ранговые корреляции [Текст]. М.: Статистика, 1975 – 216 с

5. Лаврік В.І.. Методи математичного моделювання в екології [Текст]. Киії, ВД “Академія”, 2002 – 203 с.

7. Шварц М. Сети ЭВМ. Анализ и проектирование [Текст]. – М.: Радио и свіязь, 1981. – 336 с.

8. Вентцель Е.С. Исследование операций [Текст]. Изд-во «Советское радио»,1972.

9. Ризниченко Г. Ю., Рубин А. Б. Математические модели биологических продукционных процессов, Изд. МГУ, Москва, 1993.

10. Семакин И.Г., Хеннер Е.К. Информатика. Задачник-практикум в 2 т.: ЛБЗ, 1999 г., с. 188-189.

11. Вольтерра В. Математическая теория борьбы за существование. Наука, Москва, 1976.

Додаток

Лістинг програми

```
unit MainUnit;
interface
uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, ExtCtrls, StdCtrls, Buttons;
const
  WolfLifes = 10;
  PointSize = 10;
  CellCount = 10;
  GridColor = clGray;
type
  TAnimal = record
    Old: Integer;
    X, Y: Integer;
    Lifes: Integer;
    Animal: Integer;
    D: Boolean;
    S: Integer;
  end;
  TStatistics = record
    WolfCount: Integer;
    HareCount: Integer;
  end;
  TMainForm = class(TForm)
    AreaPaintBox: TPaintBox;
    ShowGrid: TCheckBox;
    StartBtn: TBitBtn;
    StopBtn: TBitBtn;
    BitBtn3: TBitBtn;
    WolfEdit: TLabeledEdit;
    HareEdit: TLabeledEdit;
    Timer: TTimer;
    Days: TLabel;
    Memo: TMemo;
    Graph3: TPaintBox;
    Label1: TLabel;
    Label2: TLabel;
    ScrollBar: TScrollBar;
    Label3: TLabel;
    procedure FormCreate(Sender: TObject);
    procedure AreaPaintBoxPaint(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure ShowGridClick(Sender: TObject);
```

```
    procedure StartBtnClick(Sender: TObject);
    procedure StopBtnClick(Sender: TObject);
    procedure TimerTimer(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure ScrollBarChange(Sender: TObject);
    procedure Graph3Paint(Sender: TObject);
  private
    { Private declarations }
  public
    GW, Index1: Integer;
    First: Boolean;
    Days1: Integer;
    WolfCount: Integer;
    HareCount: Integer;
    CellWidth: Integer;
    CellHeight: Integer;
    DayCount: Integer;
    Buffer: TBitMap;
    GraphBuffer: TBitMap;
    Area: array[1..10, 1..10] of TAnimal;
    Statistics: TStatistics;
    OldX, OldY, OldX1, OldY1: Integer;
    procedure DrawGrid;
    procedure DrawGraph;
    procedure DrawBuffer;
    procedure GetStatistics(var Statistics: TStatistics);
    procedure NewAnimal(Animal: Integer; Life: Integer);
  end;
var
  MainForm: TMainForm;
implementation
{$R *.dfm}

procedure TMainForm.GetStatistics(var Statistics: TStatistics);
var
  X, Y: Integer;
begin
  Statistics.WolfCount := 0;
  Statistics.HareCount := 0;
  for Y := 1 to 10 do
  for X := 1 to 10 do
  begin
    if Area[X, Y].Animal = 1 then Inc(Statistics.WolfCount);
    if Area[X, Y].Animal = 2 then Inc(Statistics.HareCount);
  end;
end;
```

```
procedure TMainForm.NewAnimal(Animal: Integer; Life: Integer);
var
  B: Boolean;
  X, Y, I: Integer;
begin
  GetStatistics(Statistics);
  if (Statistics.WolfCount > 50) and (Animal = 1) then Exit;
  if (Statistics.HareCount > 60) and (Animal = 2) then Exit;
  I := 0;
  B := False;
  Repeat
    Inc(I);
    X := 1 + Random(10);
    Y := 1 + Random(10);
    if Area[X, Y].Animal = 0 then
    begin
      B := True;
      Area[X, Y].Animal := Animal;
      Area[X, Y].S := Random(2);
      Area[X, Y].Lifes := Life;
      Area[X, Y].Old := 1;
    end;
    if I > 100 then Break;
  Until B;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
  Days1 := 0;
  GW := 0;
  Index1 := 0;
  DayCount := 0;
  First := True;
  FillChar(Area, SizeOf(Area), 0);
  Buffer := TBitMap.Create;
  Buffer.Width := AreaPaintBox.Width;
  Buffer.Height := AreaPaintBox.Height;
  GraphBuffer := TBitMap.Create;
  GraphBuffer.Width := 300;
  GraphBuffer.Height := 200;
  GraphBuffer.Canvas.Pen.Color := clWhite;
  GraphBuffer.Canvas.Brush.Color := clBlack;
  GraphBuffer.Canvas.Rectangle(0, 0, 300, 200);
  GraphBuffer.Canvas.Pen.Color := clYellow;
```

```
  GraphBuffer.Canvas.MoveTo(5, 195);
  GraphBuffer.Canvas.LineTo(5, 5);
  GraphBuffer.Canvas.LineTo(3, 10);
  GraphBuffer.Canvas.LineTo(7, 10);
  GraphBuffer.Canvas.LineTo(5, 5);
  GraphBuffer.Canvas.MoveTo(5, 195);
  GraphBuffer.Canvas.LineTo(295, 195);
  GraphBuffer.Canvas.LineTo(290, 193);
  GraphBuffer.Canvas.LineTo(290, 197);
  GraphBuffer.Canvas.LineTo(295, 195);
  GraphBuffer.Canvas.Brush.Color := clYellow;
  GraphBuffer.Canvas.TextOut(15, 5, 'Популяция');
  GraphBuffer.Canvas.TextOut(260, 170, 'Время');

  Graph3.Canvas.Draw(0, 0, GraphBuffer);
  CellWidth := AreaPaintBox.Width  div CellCount;
  CellHeight := AreaPaintBox.Height div CellCount;
end;

procedure TMainForm.DrawBuffer;
var
  X, Y, CellX, CellY: Integer;
begin
  Buffer.Canvas.Pen.Color := GridColor;
  Buffer.Canvas.Brush.Color := clBlack;
  Buffer.Canvas.Rectangle(0, 0, AreaPaintBox.Width, AreaPaintBox.Height);
  if ShowGrid.Checked then DrawGrid;
  if Timer.Enabled then
  begin
    CellX := CellWidth div 2 - (PointSize div 2);
    CellY := CellHeight div 2  - (PointSize div 2);
    for Y := 0 to CellCount - 1 do
    for X := 0 to CellCount - 1 do
    begin
      if Area[X + 1, Y + 1].Animal <> 0 then
      begin
        case Area[X + 1, Y + 1].Animal of
          1: begin
               Buffer.Canvas.Pen.Color := clRed;
               Buffer.Canvas.Brush.Color := clRed;
             end;
          2: begin
               Buffer.Canvas.Pen.Color := clGreen;
               Buffer.Canvas.Brush.Color := clGreen;
             end;
        end;
```

```
        Buffer.Canvas.Ellipse(X * CellWidth + CellX, Y * CellHeight + CellY, X *
CellWidth + CellX + PointSize, Y * CellHeight + CellY + PointSize);
      end;
    end;
  end;
  AreaPaintBox.Canvas.Draw(0, 0, Buffer);
  end;

procedure TMainForm.AreaPaintBoxPaint(Sender: TObject);
begin
  DrawBuffer;
end;
procedure TMainForm.FormClose(Sender: TObject; var Action: TCloseAction);
begin
  Buffer.Free;
  GraphBuffer.Free;
end;

procedure TMainForm.DrawGrid;
var
  X, Y: Integer;
begin
  Buffer.Canvas.Pen.Color := clGray;
  for Y := 0 to CellCount - 1 do
  for X := 0 to CellCount - 1 do
  begin
    Buffer.Canvas.MoveTo(0, Y * CellHeight);
    Buffer.Canvas.LineTo(AreaPaintBox.Width, Y * CellHeight);
    Buffer.Canvas.MoveTo(X * CellWidth, 0);
    Buffer.Canvas.LineTo(X * CellWidth, AreaPaintBox.Height);
  end;
end;

procedure TMainForm.ShowGridClick(Sender: TObject);
begin
  DrawBuffer;
end;

procedure TMainForm.StartBtnClick(Sender: TObject);
var
  B: Boolean;
  I, X, Y: Integer;
begin
  StopBtn.Enabled := True;
  StartBtn.Enabled := False;
  WolfCount := StrToInt(WolfEdit.Text);
```

```
  HareCount := StrToInt(HareEdit.Text);

  Randomize;
  for I := 1 to WolfCount do
  begin
    repeat
      B := False;
      X := 1 + Random(CellCount);
      Y := 1 + Random(CellCount);
      if Area[X, Y].Animal = 0 then
      begin
        B := True;
        Area[X, Y].Animal := 1;
        Area[X, Y].D := True;
        Area[X, Y].Lifes := WolfLifes;
        Area[X, Y].S := Random(2);
      end;
    until B;
  end;

  Randomize;
  for I := 1 to HareCount do
  begin
    repeat
      B := False;
      X := 1 + Random(CellCount);
      Y := 1 + Random(CellCount);
      if Area[X, Y].Animal = 0 then
      begin
        B := True;
        Area[X, Y].Animal := 2;
        Area[X, Y].D := True;
        Area[X, Y].S := Random(2);
        Area[X, Y].Old := 1;
      end;
    until B;
  end;
  Timer.Enabled := True;
end;

procedure TMainForm.StopBtnClick(Sender: TObject);
begin
  StopBtn.Enabled := False;
  StartBtn.Enabled := True;
  Timer.Enabled := False;
end;
```

```
procedure TMainForm.DrawGraph;
begin
  GraphBuffer.Canvas.Pen.Color := clWhite;
  GraphBuffer.Canvas.Brush.Color := clBlack;
  GraphBuffer.Canvas.Rectangle(0, 0, 300, 200);
  GraphBuffer.Canvas.Pen.Color := clYellow;
  GraphBuffer.Canvas.MoveTo(5, 195);
  GraphBuffer.Canvas.LineTo(5, 5);
  GraphBuffer.Canvas.LineTo(3, 10);
  GraphBuffer.Canvas.LineTo(7, 10);
  GraphBuffer.Canvas.LineTo(5, 5);
  GraphBuffer.Canvas.MoveTo(5, 195);
  GraphBuffer.Canvas.LineTo(295, 195);
  GraphBuffer.Canvas.LineTo(290, 193);
  GraphBuffer.Canvas.LineTo(290, 197);
  GraphBuffer.Canvas.LineTo(295, 195);
  GraphBuffer.Canvas.Brush.Color := clYellow;
  GraphBuffer.Canvas.TextOut(15, 5, 'Популяция');
  GraphBuffer.Canvas.TextOut(260, 170, 'Время');
  Graph3.Canvas.Draw(0, 0, GraphBuffer);
end;

procedure TMainForm.TimerTimer(Sender: TObject);
var
  I, X, Y, D, R: Integer;
begin
  Inc(DayCount);
  GetStatistics(Statistics);
  for Y := 1 to CellCount do
  for X := 1 to CellCount do
  begin
    if (Area[X, Y].Animal <> 0) and (Area[X, Y].D) then
    begin
      D := Random(4);
      case D of
        0: begin
             if Y > 1 then
             begin
               case Area[X, Y].Animal of
                 1: begin
                      if (Area[X, Y - 1].Animal = 1) and (Area[X, Y - 1].S <> Area[X, Y].S) and
(Area[X, Y].Lifes > 4) and (Area[X, Y - 1].Lifes > 4) then NewAnimal(1, (Area[X, Y].Lifes
+ Area[X, Y - 1].Lifes) div 2);
                      if (Area[X, Y - 1].Animal = 0) or ((Area[X, Y - 1].Animal = 2){and
(Statistics.HareCount>20)}) then
```

```
            begin
             if (Area[X, Y - 1].Animal = 2) and (Statistics.HareCount>10) then Area[X,
Y].Lifes := WolfLifes else Dec(Area[X, Y].Lifes);
             if Area[X, Y].Lifes > 0 then
             begin
               Area[X, Y - 1] := Area[X, Y];
               Area[X, Y - 1].D := False;
             end;
             FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
            end;
          end;
        2: begin
            Inc(Area[X, Y].Old);
            if Area[X, Y].Old > 20 then FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0) else
            begin
             if (Area[X, Y - 1].Animal = 2) and (Area[X, Y - 1].S <> Area[X, Y].S) and
(Area[X, Y - 1].Old < 19) then NewAnimal(2, 1);
             if Area[X, Y - 1].Animal = 0 then
             begin
               Area[X, Y - 1] := Area[X, Y];
               Area[X, Y - 1].D := False;
               FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
             end;
            end;
          end;
       end;
      end;
    end;
   1: begin
      if Y < CellCount then
      begin
       case Area[X, Y].Animal of
        1: begin
            if (Area[X, Y + 1].Animal = 1) and (Area[X, Y + 1].S <> Area[X, Y].S)and
(Area[X, Y].Lifes > 4) and (Area[X, Y + 1].Lifes > 4) then NewAnimal(1, (Area[X, Y].Lifes
+ Area[X, Y + 1].Lifes) div 2);
            if (Area[X, Y + 1].Animal = 0) or ((Area[X, Y + 1].Animal = 2){and
(Statistics.HareCount>20)}) then
            begin
             if (Area[X, Y + 1].Animal = 2) and (Statistics.HareCount>10) then Area[X,
Y].Lifes := WolfLifes else Dec(Area[X, Y].Lifes);
             if Area[X, Y].Lifes > 0 then
             begin
               Area[X, Y + 1] := Area[X, Y];
               Area[X, Y + 1].D := False;
             end;
```

```
          FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
         end;
        end;
      2: begin
         Inc(Area[X, Y].Old);

         if Area[X, Y].Old > 20 then FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0) else
         begin
          if (Area[X, Y + 1].Animal = 2) and (Area[X, Y + 1].S <> Area[X, Y].S) and
(Area[X, Y + 1].Old < 19) then NewAnimal(2, 1);
          if Area[X, Y + 1].Animal = 0 then
          begin
           Area[X, Y + 1] := Area[X, Y];
           Area[X, Y + 1].D := False;
           FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
          end;
         end;
        end;
     end;
    end;
   end;
  2: begin
    if X > 1 then
    begin
     case Area[X, Y].Animal of
      1: begin
         if (Area[X - 1, Y].Animal = 1) and (Area[X - 1, Y].S <> Area[X, Y].S) and
(Area[X, Y].Lifes > 4) and (Area[X - 1, Y].Lifes > 4)then NewAnimal(1, (Area[X - 1,
Y].Lifes + Area[X, Y].Lifes) div 2);
         if (Area[X - 1, Y].Animal = 0) or ((Area[X - 1, Y].Animal = 2) {
(Statistics.HareCount > 20)})then
         begin
          if (Area[X - 1, Y].Animal = 2) then Area[X, Y].Lifes := WolfLifes else
Dec(Area[X, Y].Lifes);
          if Area[X, Y].Lifes > 0 then
          begin
           Area[X - 1, Y] := Area[X, Y];
           Area[X - 1, Y].D := False;
          end;
          FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
         end;
        end;
      2: begin
         Inc(Area[X, Y].Old);
         if Area[X, Y].Old > 20 then FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0) else
         begin
```

```
                    if (Area[X - 1, Y].Animal = 2) and (Area[X - 1, Y].S <> Area[X, Y].S) and
(Area[X - 1, Y].Old < 19) then NewAnimal(2, 1);
                    if Area[X - 1, Y].Animal = 0 then
                    begin
                      Area[X - 1, Y] := Area[X, Y];
                      Area[X - 1, Y].D := False;
                      FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
                    end;
                  end;
                end;
              end;
            end;
          end;
        3: begin
            if X < CellCount then
            begin
              case Area[X, Y].Animal of
                1: begin
                    if (Area[X + 1, Y].Animal = 1) and (Area[X + 1, Y].S <> Area[X, Y].S) and
(Area[X, Y].Lifes > 4) and (Area[X + 1, Y].Lifes > 4) then NewAnimal(1, (Area[X + 1,
Y].Lifes + Area[X, Y].Lifes) div 2);
                    if (Area[X + 1, Y].Animal = 0) or ((Area[X + 1, Y].Animal = 2){and
(Statistics.HareCount>20)}) then
                    begin
                      if (Area[X + 1, Y].Animal = 2) and (Statistics.HareCount>10) then Area[X,
Y].Lifes := WolfLifes else Dec(Area[X, Y].Lifes);
                      if Area[X, Y].Lifes > 0 then
                      begin
                        Area[X + 1, Y] := Area[X, Y];
                        Area[X + 1, Y].D := False;
                      end;
                      FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
                    end;
                  end;
                2: begin
                    Inc(Area[X, Y].Old);
                    if Area[X, Y].Old > 20 then FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0) else
                    begin
                      if (Area[X + 1, Y].Animal = 2) and (Area[X + 1, Y].S <> Area[X, Y].S) and
(Area[X + 1, Y].Old < 19) then NewAnimal(2, 1);
                      if Area[X + 1, Y].Animal = 0 then
                      begin
                        Area[X + 1, Y] := Area[X, Y];
                        Area[X + 1, Y].D := False;
                        FillChar(Area[X, Y], SizeOf(Area[X, Y]), 0);
                      end;
```

```
                end;
              end;
            end;
          end;
        end;
      end;
    end;
    end;
    for Y := 1 to CellCount do
    for X := 1 to CellCount do
    begin
      if Area[X, Y].Animal <> 0 then Area[X, Y].D := True;
    end;
    DrawBuffer;
    Days.Caption := 'День: ' + IntToStr(DayCount);
      if First then
    begin
      First := False;
      OldX := 10;
      OldY := 190;
      OldX1 := 10;
      OldY1 := 190;
      Memo.Lines.Add('День' + #9 + 'Волки' + #9 + 'Зайцы');
      Memo.Lines.Add(IntToStr(DayCount) + #9 + IntToStr(Statistics.WolfCount) + #9 +
IntToStr(Statistics.HareCount));
    end else
    begin
      Memo.Lines.Add(IntToStr(DayCount) + #9 + IntToStr(Statistics.WolfCount) + #9 +
IntToStr(Statistics.HareCount));
    end;
    GraphBuffer.Canvas.Pen.Width := 3;
    GraphBuffer.Canvas.Pen.Color := clRed;
    GraphBuffer.Canvas.MoveTo(OldX, OldY);
    OldX := DayCount * 3 + 10;
    OldY := 190 - (Statistics.WolfCount * 2);
    GraphBuffer.Canvas.LineTo(OldX, OldY);
    GraphBuffer.Canvas.Pen.Color := clGreen;
    GraphBuffer.Canvas.MoveTo(OldX1, OldY1);
    OldX1 := DayCount * 3 + 10;
    OldY1 := 190 - (Statistics.HareCount * 2);
    GraphBuffer.Canvas.LineTo(OldX1, OldY1);
    Graph3.Canvas.Draw(0, 0, GraphBuffer);
    if (Statistics.WolfCount = 0) and (Statistics.HareCount = 0) then StopBtn.Click;
end;

procedure TMainForm.BitBtn3Click(Sender: TObject);
```

```
begin
  Days1 := 0;
  GW := 0;
  Index1 := 0;
  First := True;
  StopBtn.Enabled := False;
  StartBtn.Enabled := True;
  Timer.Enabled := False;
  DayCount := 0;

  Days.Caption := 'День: ' + IntToStr(DayCount);
  FillChar(Area, SizeOf(Area), 0);
  RePaint;
  Memo.Clear;
  DrawGraph;
  end;

procedure TMainForm.ScrollBarChange(Sender: TObject);
begin
  Timer.Interval := ScrollBar.Position;
end;

procedure TMainForm.Graph3Paint(Sender: TObject);
begin
  Graph3.Canvas.Draw(0, 0, GraphBuffer);
end;
end.
```