МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Кафедра вычислительной техники

**Отчёт**

**по лабораторной работе №1**

**по дисциплине «Разработка клиент-серверных приложений»**

Тема: Параллелизм и потоки

Группа: АВТ-909

Студенты: Буйко К.А. Пилецкий Д.К.

Преподаватель: Токарев В.Г.

НОВОСИБИРСК

2022

**Цель работы**

Реализовать процесс движения объектов (вид движения по вариантам для каждого типа), остановки и возобновления движения по отдельности и всех вместе. Внести в базовый класс функционал движения, в производный – алгоритм движения для конкретного типа.

**Задание**

1. Броуновское движение со случайным изменением значения скорости и направления
2. Синусоидальная пульсация размера фигуры

**Ход выполнения работы.**

В лабораторной работе дополнялись ранее разработанные классы объектов: Car (базовый класс), Cargo и PassCar (наследники) – грузовые и легковые машины соответственно.

В каждом из этих классов реализован метод move(), который отвечает за движение объекта.

Листинг 1. Метод move() класса Cargo.

Синусоидальная пульсация размера фигуры.

public void move() {  
 if (isMove) {  
 t += 0.01;  
 w = (int) (W \* (*sin*(t) + 1.0));  
 h = (int) (H \* (*sin*(t) + 1.0));  
 }  
}

Листинг 2. Метод move() класса PassCar.

Броуновское движение.

public void move() {  
 if (isMove) {  
 Random rand = new Random();  
 int step = rand.nextInt(30) + 1;  
 cx += rand.nextInt(step) - step / 2;  
 cy += rand.nextInt(step) - step / 2;  
 }  
}

Для реализации одновременного движения объектов и их отрисовка необходимо создать несколько потоков, отвечающих каждый за свою задачу соответственно.

Таким образом, были разработаны классы потоков движения и отрисовки: AIThread, PaintThread.

Листинг 3. Класс PaintThread

public static class PaintThread extends Thread {  
 Habitat h;  
 PaintThread() {  
 super("Paint Thread");  
  
 }  
 public void setH(Habitat hab)  
 {  
 h=hab;  
 }  
 @Override  
 public void run() {  
 while (!Check.*isExit*)  
 while (Check.*isWork*) {  
 h.repaint();  
 // System.out.println("Print");  
 try {  
 *sleep*(1);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

Листинг 4. Класс AIThread.

public static class AIThread extends Thread {  
 Habitat h;  
 AIThread() {  
 super("GenThread");  
 }  
 public void setH(Habitat hab)  
 {  
 h=hab;  
 }  
 @Override  
 public void run() {  
 while (Check.*isWork*) {  
 if (Check.*isMove*) {  
 for (int i = 0; i < h.cars.size(); i++)  
 h.cars.get(i).move();  
 }  
 try {  
 *sleep*(10);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }  
}

Для реализации остановки отдельно каждого объекта было добавлено свойство isMove в базовый класс, которое при каждой попытке сместить объект проверяется на истинность. Значение данного свойства меняется с помощью клика правой кнопкой мыши по объекту.

Листинг 5. Кейс срабатывания правого клика мышью.

case 3:{  
 for (int i = 0; i < h.cars.size(); i++) {  
 if (h.cars.get(i).isIn(e.getX(), e.getY())) {  
 synchronized (h.cars) {  
 h.cars.get(i).isMove = !h.cars.get(i).isMove;  
 }  
 }  
 }  
}

\*для проверки того, что пользователь попал в какой-либо объект была ранее описана функция isIn(int x, int y), которая проверяет принадлежность какой-либо точки объекту.

Для остановки всех объектов одновременно необходимо использовать нажатие на клавишу пробела.

Листинг 6. Вырезка из KeyListener.

@Override  
public void keyTyped(KeyEvent e) {  
 if (e.getKeyChar() == ' ')  
 Check.*isMove* = !Check.*isMove*;  
  
}

\*для учета поведения объектов (двигаются или нет) было добавлено свойство isMove в основной класс учета переключателей.

**Вывод**

В ходе выполнения лабораторной работы были практически закреплены навыки распараллеливания процессов.

Также была создана программа моделирования создания, удаления и движения объектов.