Міністерство освіти і науки України

Національний університет «Львівська політехніка»

Інститут телекомунікацій, радіоелектроніки та електронної техніки

Кафедра електронних засобів інформаційно-комп’ютерних технологій



**ЗВІТ**

**Про виконання лабораторної роботи № 3**

**на тему: Гра «Битва дроїдів»**

**з курсу «Основи програмування , частина 2»**

**Виконав:**

студент групи ОІ-26

**Маланій Олексій Миколайович**

**Прийняв:**

Мельник Р.В.

Львів – 2024

**Мета роботи**: Написати свою реалізацію консольної гри “Битва дроїдів”.

**Хід роботи**

1. Створіть базовий клас Droid, від якого будуть походити інші підкласи (види дроїдів), які будуть відрізнятися різними характеристиками. Мінімальний набір характеристик: name, health, damage.

2. Додайте можливість різних видів бою: 1 на 1, або команда на команду.

3. Класи потрібно грамотно розкласти по пакетах.

4. У програмі має бути консольне меню. Мінімальний набір команд:

− створити дроїда (обраного виду);

− показати список створених дроїдів;

− запустити бій 1 на 1 (вибрати дроїдів, які будуть змагатися);

− запустити бій команда на команду (сформувати команди суперників з дроїдів, яких ви створили у першому пункті);

− записати проведений бій у файл;

− відтворити проведений бій зі збереженого файлу;

− вийти з програми.

**Мета роботи:** У процесі виконання лабораторної роботи я навчаться використовувати об'єктно-орієнтований підхід, застосовуючи наслідування, поліморфізм та інкапсуляцію для створення класів дроїдів і боїв. Також програма забезпечує збереження результатів боїв у файл і їх відтворення, що дає змогу працювати з файловими операціями в Java.

**Хід роботи**

| **Main:** | |
| --- | --- |
| package main; import droids.\*; import battle.Battle; import java.io.FileWriter; import java.io.IOException; import java.io.BufferedReader; import java.io.FileReader; import java.util.ArrayList; import java.util.List; import java.util.Scanner;  public class Main {  private static List<Droid> droids = new ArrayList<>();  private static List<Droid> team1 = new ArrayList<>();  private static List<Droid> team2 = new ArrayList<>();  public static void main(String[] args) {  Scanner scanner = new Scanner(System.in);  while (true) {  System.out.println("1. Create a droid");  System.out.println("2. Show list of droids");  System.out.println("3. Launch a 1-on-1 battle");  System.out.println("4. Launch a team-on-team battle");  System.out.println("5. Record the battle in a file");  System.out.println("6. Play the battle from the saved file");  System.out.println("7. Exit");  int choice = scanner.nextInt();  scanner.nextLine();   switch (choice) {  case 1:  createDroid(scanner);  break;  case 2:  showDroids();  break;  case 3:  oneOnOneBattle(scanner);  break;  case 4:  teamOnTeamBattle(scanner);  break;  case 5:  System.out.println("Enter filename to record the battle:");  String recordFilename = scanner.nextLine();  System.out.println("Is it a team battle? (true/false):");  boolean isTeamBattle = scanner.nextBoolean();  recordBattleToFile(recordFilename, team1, team2, isTeamBattle);  break;  case 6:  System.out.println("Enter filename to play the battle:");  String playFilename = scanner.nextLine();  playBattleFromFile(playFilename);  break;  case 7:  System.exit(0);  break;  default:  System.out.println("Invalid choice. Try again.");  }  }  }   private static void createDroid(Scanner scanner) {  System.out.println("Enter droid type (1 for BattleDroid, 2 for SupportDroid):");  int type = scanner.nextInt();  scanner.nextLine();   System.out.println("Enter droid name:");  String name = scanner.nextLine();   Droid droid;  if (type == 1) {  droid = new BattleDroid(name);  } else {  droid = new SupportDroid(name);  }  droids.add(droid);  System.out.println("Droid created: " + droid);  }   private static void showDroids() {  for (Droid droid : droids) {  System.out.println(droid);  }  }   private static void oneOnOneBattle(Scanner scanner) {  System.out.println("Select first droid by index:");  int index1 = scanner.nextInt();  System.out.println("Select second droid by index:");  int index2 = scanner.nextInt();   Droid droid1 = droids.get(index1);  Droid droid2 = droids.get(index2);   team1.clear();  team2.clear();  team1.add(droid1);  team2.add(droid2);   Battle.oneOnOne(droid1, droid2);  System.out.println("Battle finished. Results:");  System.out.println(droid1);  System.out.println(droid2);  }   private static void teamOnTeamBattle(Scanner scanner) {  System.out.println("Select number of droids in each team:");  int teamSize = scanner.nextInt();   List<Droid> team1 = new ArrayList<>();  List<Droid> team2 = new ArrayList<>();   System.out.println("Select droids for Team 1:");  for (int i = 0; i < teamSize; i++) {  System.out.println("Select droid by index:");  int index = scanner.nextInt();  team1.add(droids.get(index));  }   System.out.println("Select droids for Team 2:");  for (int i = 0; i < teamSize; i++) {  System.out.println("Select droid by index:");  int index = scanner.nextInt();  team2.add(droids.get(index));  }   Battle.teamOnTeam(team1, team2);   System.out.println("Battle finished. Results:");  System.out.println("Team 1:");  for (Droid droid : team1) {  System.out.println(droid);  }  System.out.println("Team 2:");  for (Droid droid : team2) {  System.out.println(droid);  }  }  private static void recordBattleToFile(String filename, List<Droid> team1, List<Droid> team2, boolean isTeamBattle) {  try (FileWriter writer = new FileWriter(filename)) {  if (isTeamBattle) {  writer.write("Team Battle:\n");  writer.write("Team 1:\n");  for (Droid droid : team1) {  writer.write(droid.toString() + "\n");  }  writer.write("Team 2:\n");  for (Droid droid : team2) {  writer.write(droid.toString() + "\n");  }  } else {  writer.write("1-on-1 Battle:\n");  writer.write("Droid 1:\n");  writer.write(team1.get(0).toString() + "\n");  writer.write("Droid 2:\n");  writer.write(team2.get(0).toString() + "\n");  }  } catch (IOException e) {  System.out.println("An error occurred while recording the battle: " + e.getMessage());  }  }  private static void playBattleFromFile(String filename) {  try (BufferedReader reader = new BufferedReader(new FileReader(filename))) {  String line;  while ((line = reader.readLine()) != null) {  System.out.println(line);  }  } catch (IOException e) {  System.out.println("An error occurred while playing the battle: " + e.getMessage());  }  } } |  |

**Battle:**

| **package battle;  import droids.Droid; import java.util.List;  public class Battle {  public static void oneOnOne(Droid droid1, Droid droid2) {  while (droid1.isAlive() && droid2.isAlive()) {  droid1.takeDamage(droid2.getDamage());  if (droid1.isAlive()) {  droid2.takeDamage(droid1.getDamage());  }  }  }   public static void teamOnTeam(List<Droid> team1, List<Droid> team2) {  while (team1.stream().anyMatch(Droid::isAlive) && team2.stream().anyMatch(Droid::isAlive)) {  for (Droid droid1 : team1) {  for (Droid droid2 : team2) {  if (droid1.isAlive() && droid2.isAlive()) {  droid1.takeDamage(droid2.getDamage());  if (droid1.isAlive()) {  droid2.takeDamage(droid1.getDamage());  }  }  }  }  }  } }** |
| --- |

**BattleDroid:**

| **package droids;  public class BattleDroid extends Droid {  public BattleDroid(String name) {  super(name, 100, 20);  } }** |
| --- |

**Droid:**

| **package droids;  public abstract class Droid {  protected String name;  protected int health;  protected int damage;   public Droid(String name, int health, int damage) {  this.name = name;  this.health = health;  this.damage = damage;  }   public String getName() {  return name;  }   public int getHealth() {  return health;  }   public int getDamage() {  return damage;  }   public void takeDamage(int damage) {  this.health -= damage;  }   public boolean isAlive() {  return this.health > 0;  }   @Override  public String toString() {  return "Droid{name='" + name + "', health=" + health + ", damage=" + damage + "}";  } }** |
| --- |

**SupportDroid:**

| **package droids;  public class SupportDroid extends Droid {  public SupportDroid(String name) {  super(name, 80, 10);  } }** |
| --- |

**Висновки:** У ході виконання роботи було реалізовано консольну гру "Битва дроїдів", що включає базові принципи об'єктно-орієнтованого програмування: наслідування, поліморфізм і інкапсуляцію. Я розширив свої навички в проектуванні класів, організації взаємодії між ними та структуризації коду через пакети. Також було опрацьовано механізми зчитування і запису даних у файли для збереження результатів боїв. Робота дозволила закріпити знання в області роботи з абстрактними класами, методами та створенням багатофункціональних програм у Java.