Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних технологій, автоматики та метрології

Кафедра ЕОМ



**Звіт**

**До лабораторної роботи №6**

# З дисципліни: «Кросплатформні засоби програмування»

На тему «Параметризоване програмування»

# Варіант №18

Виконала: ст. гр. КІ-305

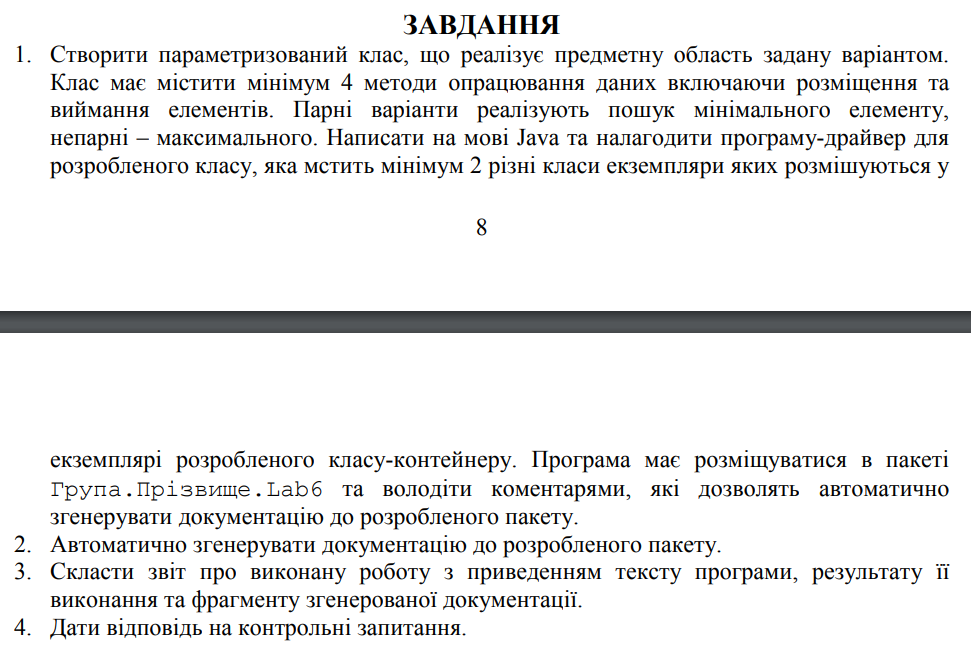
Лесяк Х. В.

Прийняв:

Іванов Ю.С.

Львів – 2023

**Мета:** оволодіти навиками параметризованого програмування мовою Java.





**Виконання:**

**Лістинг програми:**

package KI305.Lesiak.Lab6;  
import java.util.ArrayList;  
  
public class WeaponStorage<T extends Weapon> {  
 private final ArrayList<T> weapons;  
 private static int *sizeCounter* = 0;  
  
  
 public static int getSizeCounter() {  
 return *sizeCounter*;  
 }  
  
 public WeaponStorage() {  
 weapons = new ArrayList<>();  
 }  
  
 public void addWeapon(T weapon) {  
 weapons.add(weapon);  
 System.*out*.println("Weapon added: " + weapon.getName());  
 *sizeCounter*++;  
 }  
 public int size() {  
 return *sizeCounter*; // Return the size  
 }  
  
 public void removeWeapon(int index) {  
 if (index >= 0 && index < weapons.size()) {  
 T removedWeapon = weapons.remove(index);  
 System.*out*.println("Weapon removed: " + removedWeapon.getName());  
 } else {  
 System.*out*.println("Invalid index");  
 }  
 }  
  
 public T findMin() {  
 if (!weapons.isEmpty()) {  
 T minWeapon = weapons.get(0);  
 for (int i = 1; i < weapons.size(); i++) {  
 if (weapons.get(i).compareTo(minWeapon) < 0) {  
 minWeapon = weapons.get(i);  
 }  
 }  
 return minWeapon;  
 }  
 return null;  
 }  
  
 public T findMax() {  
 if (!weapons.isEmpty()) {  
 T maxWeapon = weapons.get(0);  
 for (int i = 1; i < weapons.size(); i++) {  
 if (weapons.get(i).compareTo(maxWeapon) > 0) {  
 maxWeapon = weapons.get(i);  
 }  
 }  
 return maxWeapon;  
 }  
 return null;  
 }  
}  
  
interface Weapon extends Comparable<Weapon> {  
 String getName();  
  
 int getDamage();  
  
 void setName(String name);  
  
 void setDamage(int damage);  
}  
  
class Gun implements Weapon {  
 private String name;  
 private int damage;  
  
 public Gun(String name, int damage) {  
 this.name = name;  
 this.damage = damage;  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public int getDamage() {  
 return damage;  
 }  
  
 @Override  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void setDamage(int damage) {  
 this.damage = damage;  
 }  
  
 @Override  
 public int compareTo(Weapon other) {  
 return Integer.*compare*(this.damage, other.getDamage()); // Пошук мінімального елемента  
 }  
}  
  
class Sword implements Weapon {  
 private String name;  
 private int damage;  
  
 public Sword(String name, int damage) {  
 this.name = name;  
 this.damage = damage;  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public int getDamage() {  
 return damage;  
 }  
  
 @Override  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void setDamage(int damage) {  
 this.damage = damage;  
 }  
  
 @Override  
 public int compareTo(Weapon other) {  
 return Integer.*compare*(this.damage, other.getDamage()); // Пошук мінімального елемента  
 }  
}  
  
class Bow implements Weapon {  
 private String name;  
 private int damage;  
  
 public Bow(String name, int damage) {  
 this.name = name;  
 this.damage = damage;  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public int getDamage() {  
 return damage;  
 }  
  
 @Override  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void setDamage(int damage) {  
 this.damage = damage;  
 }  
  
 @Override  
 public int compareTo(Weapon other) {  
 return Integer.*compare*(this.damage, other.getDamage()); // Пошук мінімального елемента  
 }  
}  
  
class Axe implements Weapon {  
 private String name;  
 private int damage;  
  
 public Axe(String name, int damage) {  
 this.name = name;  
 this.damage = damage;  
 }  
  
 @Override  
 public String getName() {  
 return name;  
 }  
  
 @Override  
 public int getDamage() {  
 return damage;  
 }  
  
 @Override  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public void setDamage(int damage) {  
 this.damage = damage;  
 }  
  
 @Override  
 public int compareTo(Weapon other) {  
 return Integer.*compare*(this.damage, other.getDamage()); // Пошук мінімального елемента  
 }  
 public static class WeaponDemo {  
 public static void main(String[] args) {  
 WeaponStorage<Weapon> storage = *createAndPopulateStorage*();  
  
  
  
 Weapon minWeapon = storage.findMin();  
 Weapon maxWeapon = storage.findMax();  
  
 if (minWeapon != null) {  
 System.*out*.println("Мінімальна зброя: " + minWeapon.getName() + " з шкодою " + minWeapon.getDamage());  
 } else {  
 System.*out*.println("Контейнер пустий");  
 }  
  
 if (maxWeapon != null) {  
 System.*out*.println("Максимальна зброя: " + maxWeapon.getName() + " з шкодою " + maxWeapon.getDamage());  
 } else {  
 System.*out*.println("Контейнер пустий");  
 }  
  
 // Приклад використання методу setDamage  
 if (minWeapon != null) {  
 minWeapon.setDamage(50); // Встановлюємо нове значення damage для мінімальної зброї  
 System.*out*.println("Нова шкода мінімальної зброї: " + minWeapon.getDamage());  
 }  
  
 // Приклад використання методу setName  
 if (maxWeapon != null) {  
 maxWeapon.setName("New Max Weapon"); // Встановлюємо нове ім'я для максимальної зброї  
 System.*out*.println("Нова назва максимальної зброї: " + maxWeapon.getName());  
 }  
  
 // Приклад використання методу removeWeapon  
 storage.removeWeapon(1); // Видаляємо зброю за індексом 1\*/  
 }  
 }  
  
  
 public static WeaponStorage<Weapon> createAndPopulateStorage() {  
 WeaponStorage<Weapon> storage = new WeaponStorage<>();  
  
 // Створення об'єктів зброї і додавання їх до контейнера  
 Gun shotgun = new Gun("Shotgun", 100);  
 Sword katana = new Sword("Katana", 80);  
 Bow longbow = new Bow("Longbow", 70);  
 Axe battleAxe = new Axe("Battle Axe", 90);  
  
 storage.addWeapon(shotgun);  
 storage.addWeapon(katana);  
 storage.addWeapon(longbow);  
 storage.addWeapon(battleAxe);  
 return storage;  
  
  
 }  
}

**Результати:**

**Зображення, що містить текст, знімок екрана, Шрифт

Автоматично згенерований опис**

**Висновок:** оволоділа навиками параметризованого програмування мовою Java.