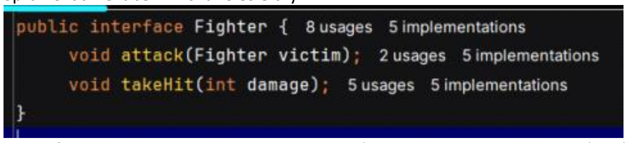
3

wtorek, 4 marca 2025 09:55

Sprawozdanie lab3 - Michał Ciesielczyk

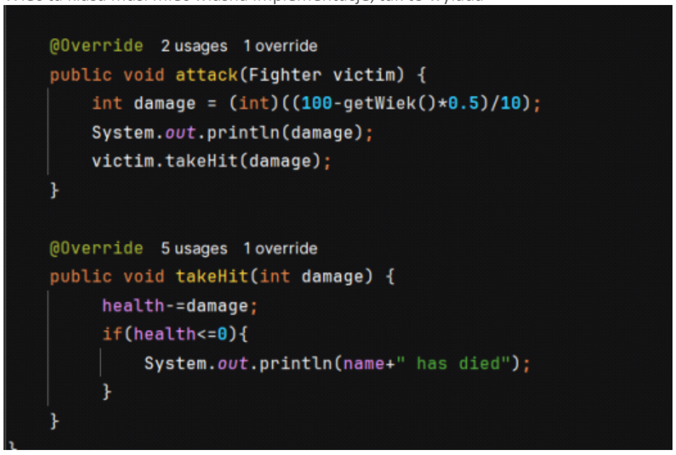


```
public interface Fighter {  8 usages  5 implementations
    void attack(Fighter victim);  2 usages  5 implementations
    void takeHit(int damage);  5 usages  5 implementations
}
```

a interfejs jest zbiorem abstrakcyjnych metod(czyli takic bez implementacji), które muszą zostać zaimplementowane przez każdą klasę osobno np.:

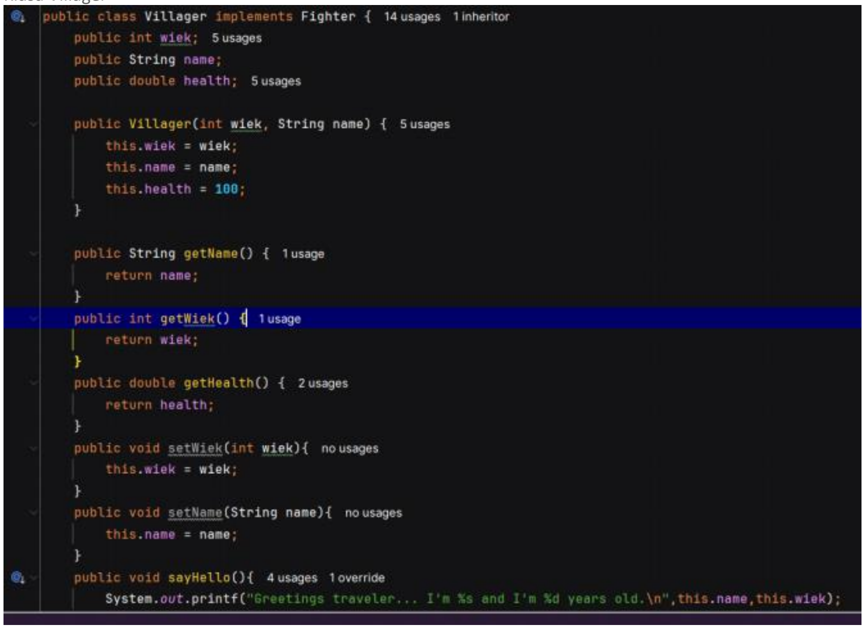
public class Villager implements Fighter {

Wiec ta klasa musi mieć wlasna implementacje, tak to wylada



```
    @Override  2 usages  1 override
    public void attack(Fighter victim) {
        int damage = (int)((100-getWiek()*0.5)/10);
        System.out.println(damage);
        victim.takeHit(damage);
    }

    @Override  5 usages  1 override
    public void takeHit(int damage) {
        health-=damage;
        if(health<=0){
            System.out.println(name+" has died");
        }
    }
}
```

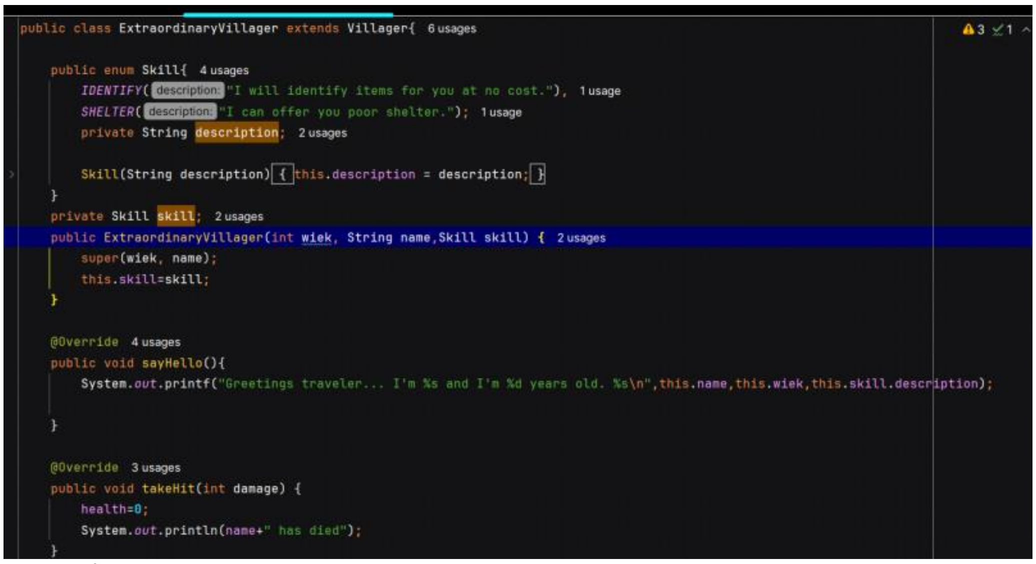
Klasa Villager



```
public class Villager implements Fighter {  14 usages  1 inheritor
    public int wiek;  5 usages
    public String name;
    public double health;  5 usages

    public Villager(int wiek, String name) {  5 usages
        this.wiek = wiek;
        this.name = name;
        this.health = 100;
    }

    public String getName() {  1 usage
        return name;
    }
    public int getWiek() {  1 usage
        return wiek;
    }
    public double getHealth() {  2 usages
        return health;
    }
    public void setWiek(int wiek){  no usages
        this.wiek = wiek;
    }
    public void setName(String name){  no usages
        this.name = name;
    }
    public void sayHello(){  4 usages  1 override
        System.out.printf("Greetings traveler... I'm %s and I'm %d years old.\n",this.name,this.wiek);
```

Koonstruktor z wartoscia domyslna zdrowie z getterami i setterami

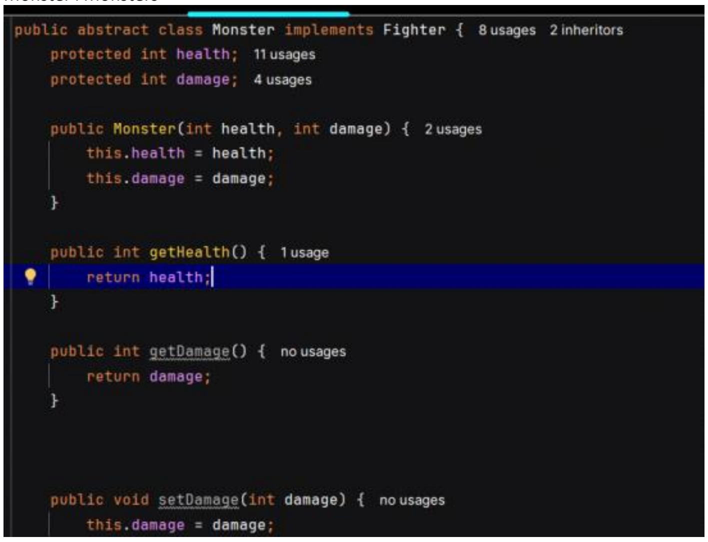
klasa ExtraordinaryVillager dziedzicząca po klasie Villager



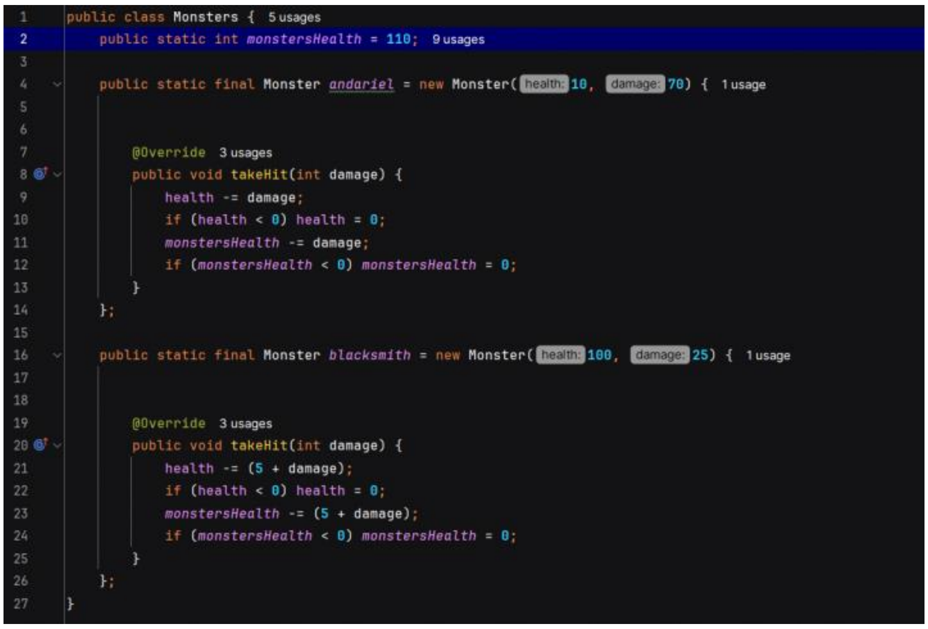
```java
public class ExtraordinaryVillager extends Villager{  6 usages

    public enum Skill{  4 usages
        IDENTIFY( description: "I will identify items for you at no cost."),  1 usage
        SHELTER( description: "I can offer you poor shelter.");  1 usage
        private String description;  2 usages

        Skill(String description) { this.description = description; }
    }
    private Skill skill;  2 usages
    public ExtraordinaryVillager(int wiek, String name,Skill skill) {  2 usages
        super(wiek, name);
        this.skill=skill;
    }

    @Override  4 usages
    public void sayHello(){
        System.out.printf("Greetings traveler... I'm %s and I'm %d years old. %s\n",this.name,this.wiek,this.skill.description);

    }

    @Override  3 usages
    public void takeHit(int damage) {
        health=0;
        System.out.println(name+" has died");
    }
```

Nadpisane funckja sayhello i ine, obiekt tej klasy zadaje 0 i od razu umiera

Monster i Monsters



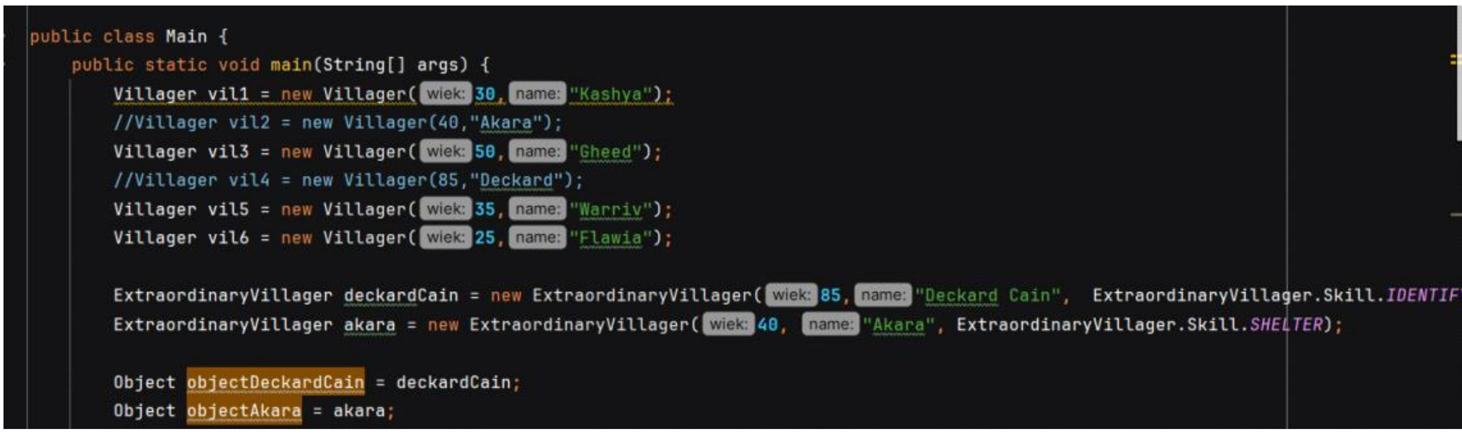
```java
public abstract class Monster implements Fighter {  8 usages  2 inheritors
    protected int health;  11 usages
    protected int damage;  4 usages

    public Monster(int health, int damage) {  2 usages
        this.health = health;
        this.damage = damage;
    }

    public int getHealth() {  1 usage
        return health;
    }

    public int getDamage() {  no usages
        return damage;
    }



    public void setDamage(int damage) {  no usages
        this.damage = damage;
```



```java
public class Monsters {  5 usages
    public static int monstersHealth = 110;  9 usages

    public static final Monster andariel = new Monster( health: 10,  damage: 70) {  1 usage


        @Override  3 usages
        public void takeHit(int damage) {
            health -= damage;
            if (health < 0) health = 0;
            monstersHealth -= damage;
            if (monstersHealth < 0) monstersHealth = 0;
        }
    };

    public static final Monster blacksmith = new Monster( health: 100,  damage: 25) {  1 usage


        @Override  3 usages
        public void takeHit(int damage) {
            health -= (5 + damage);
            if (health < 0) health = 0;
            monstersHealth -= (5 + damage);
            if (monstersHealth < 0) monstersHealth = 0;
        }
    };
}
```

public static int monstersHealth = 110; - pole statyczne
Tworzymy sobie dwa obiekty mosnter i nadpisujemy dla nich takehit
Anadirl i blacksmit - anonimowe klasy

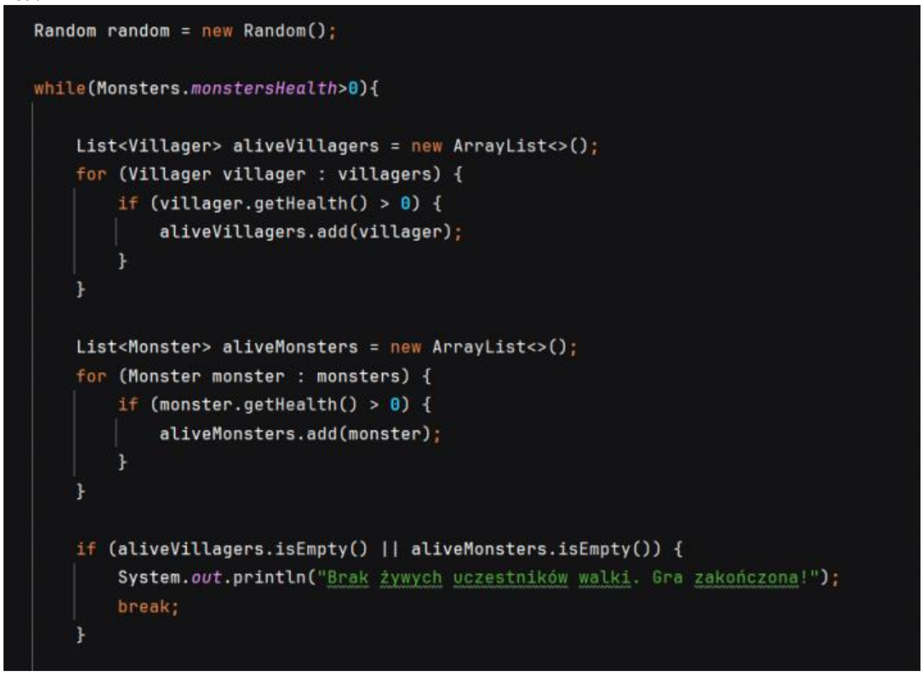
Main



```java
public class Main {
    public static void main(String[] args) {
        Villager vil1 = new Villager( wiek: 30, name: "Kashya");
        //Villager vil2 = new Villager(40,"Akara");
        Villager vil3 = new Villager( wiek: 50, name: "Gheed");
        //Villager vil4 = new Villager(85,"Deckard");
        Villager vil5 = new Villager( wiek: 35, name: "Warriv");
        Villager vil6 = new Villager( wiek: 25, name: "Flawia");

        ExtraordinaryVillager deckardCain = new ExtraordinaryVillager( wiek: 85, name: "Deckard Cain",  ExtraordinaryVillager.Skill.IDENTIF
        ExtraordinaryVillager akara = new ExtraordinaryVillager( wiek: 40, name: "Akara", ExtraordinaryVillager.Skill.SHELTER);

        Object objectDeckardCain = deckardCain;
        Object objectAkara = akara;
```

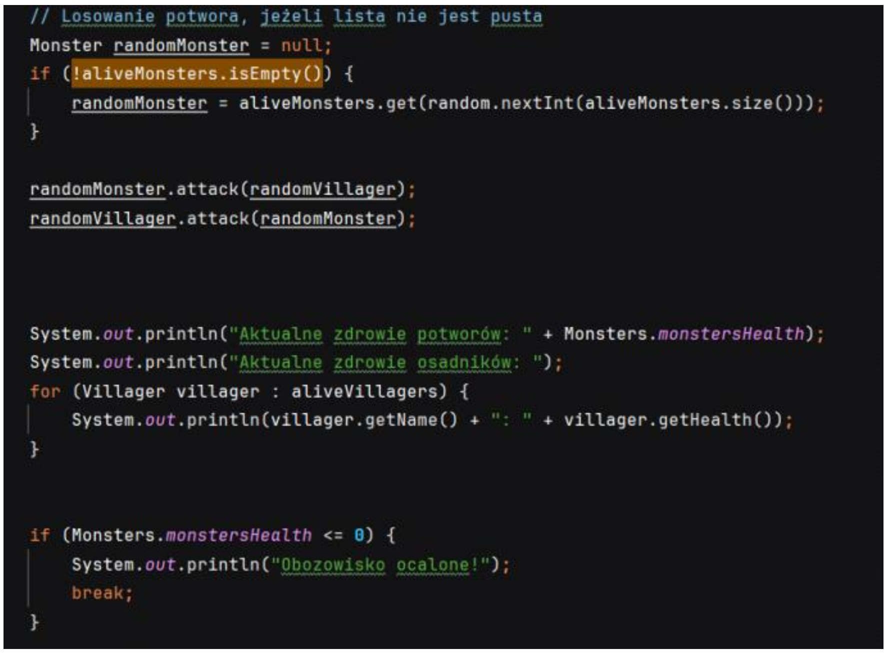
Tworze sobie obikety klasy villager i extraordinary
Ukrycie akara i deckara
Nie można na nic wywolywac netody klas

Petla



```java
Random random = new Random();

while(Monsters.monstersHealth>0){

    List<Villager> aliveVillagers = new ArrayList<>();
    for (Villager villager : villagers) {
        if (villager.getHealth() > 0) {
            aliveVillagers.add(villager);
        }
    }

    List<Monster> aliveMonsters = new ArrayList<>();
    for (Monster monster : monsters) {
        if (monster.getHealth() > 0) {
            aliveMonsters.add(monster);
        }
    }

    if (aliveVillagers.isEmpty() || aliveMonsters.isEmpty()) {
        System.out.println("Brak żywych uczestników walki. Gra zakończona!");
        break;
    }
```

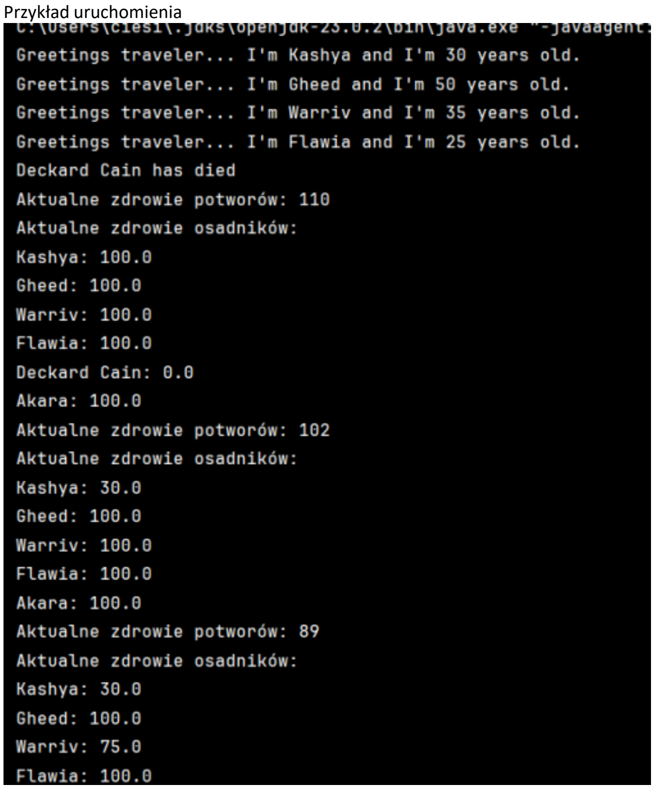
Twooerze sobie nowa liste i wpisuje tam tylko tych którzy jeszcze zyja, żeby losoowac z zywych



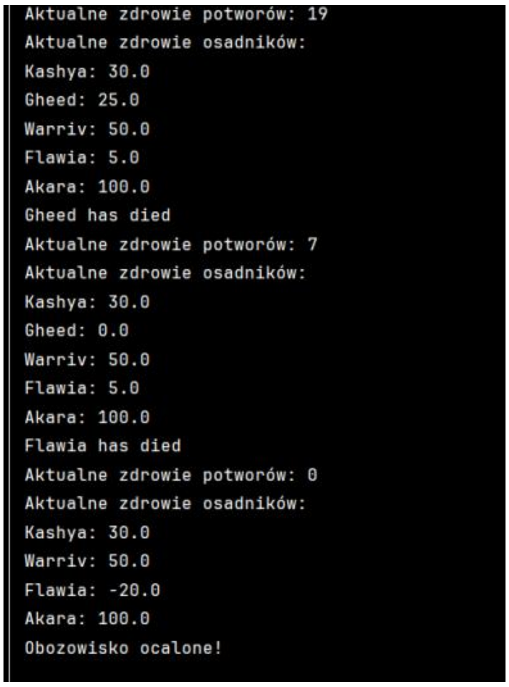
```java
// Losowanie potwora, jeżeli lista nie jest pusta
Monster randomMonster = null;
if (!aliveMonsters.isEmpty()) {
    randomMonster = aliveMonsters.get(random.nextInt(aliveMonsters.size()));
}

randomMonster.attack(randomVillager);
randomVillager.attack(randomMonster);



System.out.println("Aktualne zdrowie potworów: " + Monsters.monstersHealth);
System.out.println("Aktualne zdrowie osadników: ");
for (Villager villager : aliveVillagers) {
    System.out.println(villager.getName() + ": " + villager.getHealth());
}


if (Monsters.monstersHealth <= 0) {
    System.out.println("Obozowisko ocalone!");
    break;
}
```

Sprawdzam czy nie pusta dla pewnosci
Zadaja sobie obrazenia i wypisuje kto zyje z iloscia hp

Przykład uruchomienia



```
C:\Users\ciesl\.jdks\openjdk-23.0.2\bin\java.exe "-javaagent:
Greetings traveler... I'm Kashya and I'm 30 years old.
Greetings traveler... I'm Gheed and I'm 50 years old.
Greetings traveler... I'm Warriv and I'm 35 years old.
Greetings traveler... I'm Flawia and I'm 25 years old.
Deckard Cain has died
Aktualne zdrowie potworów: 110
Aktualne zdrowie osadników:
Kashya: 100.0
Gheed: 100.0
Warriv: 100.0
Flawia: 100.0
Deckard Cain: 0.0
Akara: 100.0
Aktualne zdrowie potworów: 102
Aktualne zdrowie osadników:
Kashya: 30.0
Gheed: 100.0
Warriv: 100.0
Flawia: 100.0
Akara: 100.0
Aktualne zdrowie potworów: 89
Aktualne zdrowie osadników:
Kashya: 30.0
Gheed: 100.0
Warriv: 75.0
Flawia: 100.0
```



```
Aktualne zdrowie potworów: 19
Aktualne zdrowie osadników:
Kashya: 30.0
Gheed: 25.0
Warriv: 50.0
Flawia: 5.0
Akara: 100.0
Gheed has died
Aktualne zdrowie potworów: 7
Aktualne zdrowie osadników:
Kashya: 30.0
Gheed: 0.0
Warriv: 50.0
Flawia: 5.0
Akara: 100.0
Flawia has died
Aktualne zdrowie potworów: 0
Aktualne zdrowie osadników:
Kashya: 30.0
Warriv: 50.0
Flawia: -20.0
Akara: 100.0
Obozowisko ocalone!
```