Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования
"Национальная научно-образовательная корпорация ИТМО"

Факультет Систем Управления и Робототехники

Лабораторная работа №2
по дисциплине
«Программирование»

Вариант №874290

Выполнил:
Студент группы R3138
Велюго Кирилл Олегович

Преподаватель:
Иманзаде Фахри Рашидович

Санкт-Петербург, 2022

Оглавление

Условие задания

На основе базового класса Pokemon написать свои классы для заданных видов покемонов. Каждый вид покемона должен иметь один или два типа и стандартные базовые характеристики:

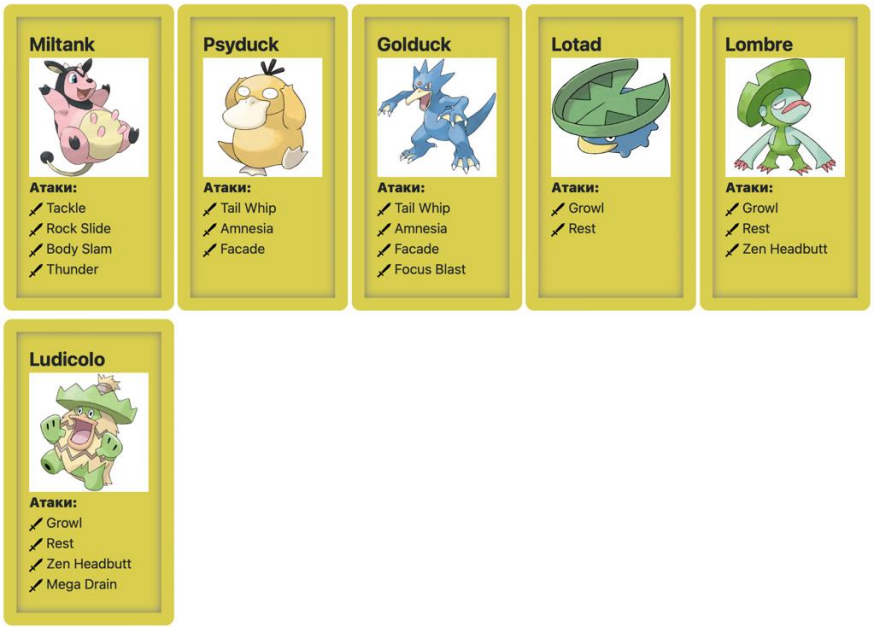
- очки здоровья (HP)
- атака (attack)
- защита (defense)
- специальная атака (special attack)
- специальная защита (special defense)
- скорость (speed)

Классы покемонов должны наследоваться в соответствии с цепочкой эволюции покемонов. На основе базовых классов PhysicalMove, SpecialMove и StatusMove реализовать свои классы для заданных видов атак.

Атака должна иметь стандартные тип, силу (power) и точность (accuracy). Должны быть реализованы стандартные эффекты атаки. Назначить каждому виду покемонов атаки в соответствии с вариантом. Уровень покемона выбирается минимально необходимым для всех реализованных атак.

Используя класс симуляции боя Battle, создать 2 команды покемонов (каждый покемон должен иметь имя) и запустить бой.

Базовые классы и симулятор сражения находятся в jar-архиве. Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах.



Исходный код:

Репозиторий GitHub: https://github.com/kirillvelyugo/ITMOLabs/tree/main/02.%20Lab

Class Program:

```
import ru.ifmo.se.pokemon.*;
import pokemons.*;

// Lab2 var. 874290

public class main
{
    public static void main(String[] args)
    {
        Battle b = new Battle();

        Miltank miltank = new Miltank("Счастливая корова", 20);
        Psyduck psyduck = new Psyduck("Утка-кря", 24);
        Golduck golduck = new Golduck("Монстрик", 12);
        Lotad lotad = new Lotad("Кувшинка", 44);
        Lombre lombre = new Lombre("Кувшимби", 43);
        Ludicolo ludicolo = new Ludicolo("Яйцешинка", 24);

        b.addAlly(miltank);
        b.addAlly(golduck);
        b.addAlly(lombre);

        b.addFoe(psyduck);
        b.addFoe(lotad);
        b.addFoe(ludicolo);

        b.go();
    }
}
```

Покемоны:

Class Golduck

```
package pokemons;

import attacks.specialAttacks.FocusBlast;
public class Golduck extends Psyduck
{
    public Golduck(String name, int lvl)
    {
        super(name, lvl);

        // set stats
        this.setStats(80, 82, 78, 95, 80, 85);

        // add attacks
        this.addMove(new FocusBlast());
    }
}
```

Class Lombre

```
package pokemons;

import attacks.physicalAttacks.ZenHeadbutt;

public class Lombre extends Lotad
{
    public Lombre (String name, int lvl)
```

```
    {
        super(name, lvl);

        // set stats
        this.setStats(60, 50, 50, 60, 70, 50);

        // add attacks
        this.addMove(new ZenHeadbutt());
    }
}
```

Class Lotad

```
package pokemons;

import attacks.statusAttacks.Growl;
import attacks.statusAttacks.Rest;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Lotad extends Pokemon
{
    public Lotad (String name, int lvl)
    {
        super(name, lvl);

        // set stats
        this.setStats(40, 30, 30, 40, 50, 30);
        this.setType(Type.WATER, Type.GRASS);

        // add attacks
        this.addMove(new Growl());
        this.addMove(new Rest());
    }
}
```

Class Ludicolo

```
package pokemons;

import attacks.specialAttacks.MegaDrain;

public class Ludicolo extends Lombre
{
    public Ludicolo (String name, int lvl)
    {
        super(name, lvl);

        // add stats
        this.setStats(80, 70, 70, 90, 100, 70);

        // add attacks
        this.addMove(new MegaDrain());
    }
}
```

Class Miltank

```
package pokemons;

import attacks.physicalAttacks.BodySlam;
import attacks.physicalAttacks.RockSlide;
import attacks.physicalAttacks.Tackle;
```

```
import attacks.specialAttacks.Thunder;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Miltank extends Pokemon
{
    public Miltank (String name, int lvl)
    {
        super(name, lvl);

        // set stats
        this.setStats(95, 80, 105, 40, 70, 100);
        this.setType(Type.NORMAL);

        // add attacks
        this.addMove(new Tackle());
        this.addMove(new RockSlide());
        this.addMove(new BodySlam());
        this.addMove(new Thunder());
    }
}
```

Class Psyduck

```
package pokemons;

import attacks.physicalAttacks.Facade;
import attacks.statusAttacks.Amnesia;
import attacks.statusAttacks.TailWhip;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Psyduck extends Pokemon
{
    public Psyduck(String name, int lvl)
    {
        super(name, lvl);

        //set stats
        this.setStats(50, 52, 48, 65, 50, 55);
        this.addType(Type.WATER);

        // add atacks
        this.addMove(new TailWhip());
        this.addMove(new Amnesia());
        this.addMove(new Facade());
    }
}
```

Атаки:

Физические атаки:

Class BodySlam

```
package attacks.physicalAttacks;

import ru.ifmo.se.pokemon.*;

import java.lang.module.ModuleReference;

public class BodySlam extends PhysicalMove
{
    public BodySlam ()
```

```
    {
        super(Type.NORMAL, 85, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon)
    {
        Effect e = new Effect().chance(0.3).condition(Status.PARALYZE);
        pokemon.addEffect(e);

        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe()
    {
        return "применяет Body Slam";
    }
}
```

Class Facade

```
package attacks.physicalAttacks;

import ru.ifmo.se.pokemon.*;

public class Facade extends PhysicalMove
{
    public Facade ()
    {
        super(Type.NORMAL, 70, 100);
    }

    @Override
    protected void applyOppDamage (Pokemon pokemon, double damage)
    {
        if (pokemon.getCondition().equals(Status.BURN) ||
pokemon.getCondition().equals(Status.POISON) ||
pokemon.getCondition().equals(Status.PARALYZE))
        {
            pokemon.setMod(Stat.HP, 2 * (int) Math.round(damage));
        }

        super.applyOppDamage(pokemon, damage);
    }

    @Override
    protected String describe(){
        return "применяет Facade";
    }
}
```

Class RockSlide

```
package attacks.physicalAttacks;

import ru.ifmo.se.pokemon.*;

public class RockSlide extends PhysicalMove {
    public RockSlide ()
    {
```

```
        super(Type.ROCK, 75, 90);
    }

    @Override
    protected void applyOppEffects (Pokemon pokemon){
        if (Math.random() <= 0.3) Effect.flinch(pokemon);

        super.applyOppEffects(pokemon);
    }


    @Override
    protected String describe(){
        return "применяет Rock Slide";
    }
}
```

Class Tackle

```
package attacks.physicalAttacks;

import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Type;

public class Tackle extends PhysicalMove {
    public Tackle (){
        super(Type.NORMAL, 40, 100);
    }

    @Override
    protected String describe(){
        return "применяет Tackle";
    }
}
```

Class ZenHeadbutt

```
package attacks.physicalAttacks;

import ru.ifmo.se.pokemon.Effect;
import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class ZenHeadbutt extends PhysicalMove {
    public ZenHeadbutt(){
        super(Type.PSYCHIC, 80, 90);
    }

    @Override
    protected void applyOppEffects (Pokemon pokemon){
        if (Math.random() <= 0.2) Effect.flinch(pokemon);

        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe(){
        return "применяет Zen Headbutt";
    }
}
```

Специальные атаки:

Class FocusBlask

```
package attacks.specialAttacks;

import ru.ifmo.se.pokemon.*;

public class FocusBlast extends SpecialMove {
    public FocusBlast (){
        super(Type.FIGHTING, 120, 70);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon) {
        Effect e = new Effect().chance(0.1).stat(Stat.SPECIAL_DEFENSE, 1).turns(1);
        pokemon.addEffect(e);

        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe(){
        return "применяет Focus Blast";
    }
}
```

Class MegaDrain

```
package attacks.specialAttacks;

import ru.ifmo.se.pokemon.*;

public class MegaDrain extends SpecialMove {
    public MegaDrain (){
        super(Type.GRASS, 40, 100);
    }

    @Override
    protected void applySelfDamage (Pokemon pokemon, double damage){
        pokemon.setMod(Stat.HP, - (int) (damage * 0.5));
    }

    @Override
    protected String describe(){
        return "применяет Mega Drain";
    }
}
```

Class Thunder

```
package attacks.specialAttacks;

import ru.ifmo.se.pokemon.*;

public class Thunder extends SpecialMove {
    public Thunder (){
        super(Type.ELECTRIC, 110, 70);
    }

    @Override
    protected void applyOppEffects (Pokemon pokemon){
        Effect e = new Effect().chance(0.3).condition(Status.PARALYZE);
        pokemon.addEffect(e);

        super.applyOppEffects(pokemon);
    }
```

```
    @Override
    protected String describe(){
        return "применяет Thunder";
    }
}
```

Статусные атаки:

Class Amnesia

```
package attacks.statusAttacks;

import ru.ifmo.se.pokemon.*;

public class Amnesia extends StatusMove
{
    public Amnesia ()
    {
        super(Type.PSYCHIC, 0, 100);
    }

    @Override
    protected void applySelfEffects(Pokemon pokemon)
    {
        pokemon.setMod(Stat.SPECIAL_DEFENSE, 2);

        super.applySelfEffects(pokemon);
    }

    @Override
    protected String describe()
    {
        return "Применяет Amnesia";
    }
}
```

Class Growl

```
package attacks.statusAttacks;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;
import ru.ifmo.se.pokemon.Type;

public class Growl extends StatusMove
{
    public Growl ()
    {
        super(Type.NORMAL, 0, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon)
    {
        pokemon.setMod(Stat.ATTACK, -2);

        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe()
    {
        return "Применяет Growl";
    }
}
```

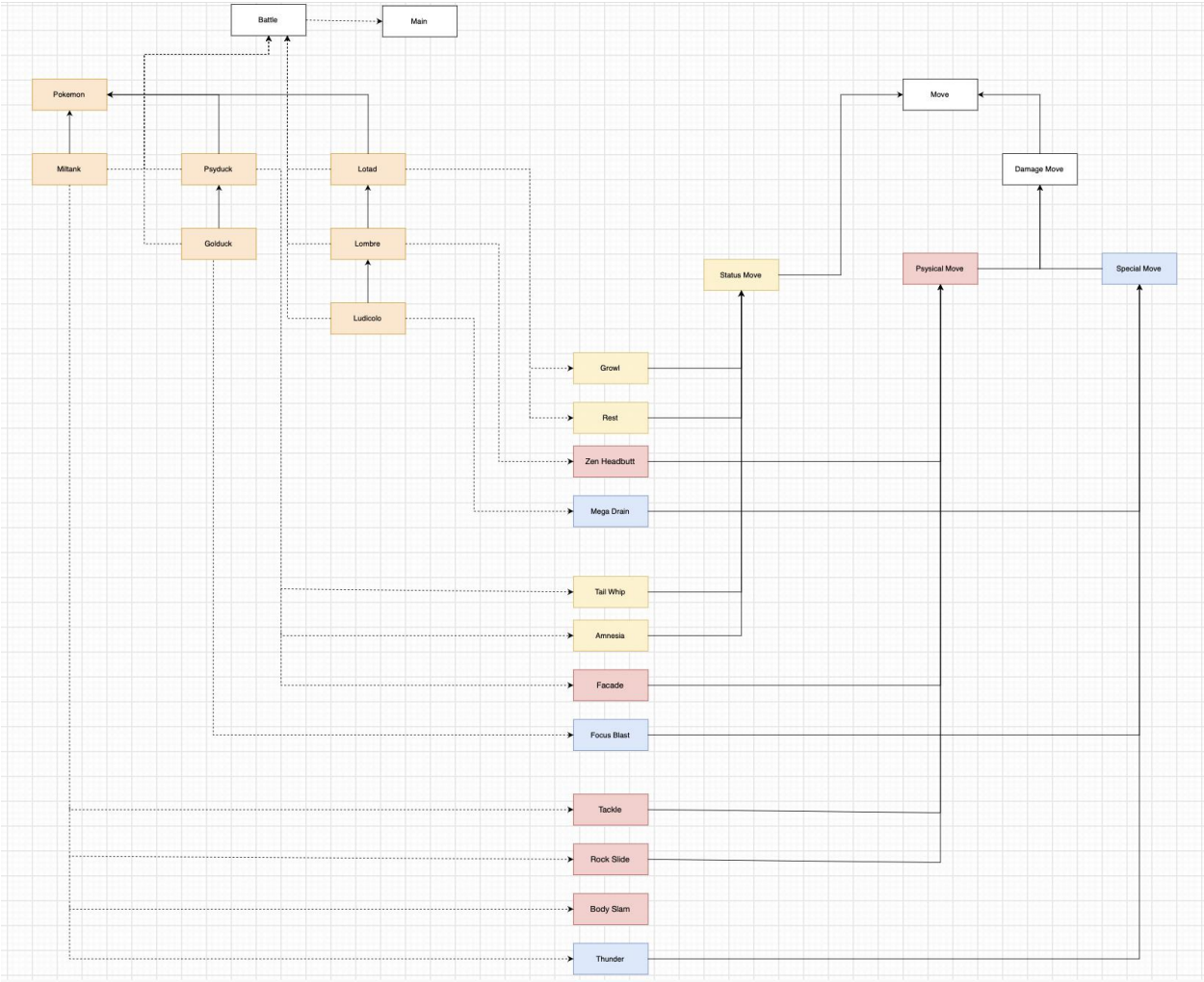
Class Rest

```
package attacks.statusAttacks;

import ru.ifmo.se.pokemon.*;

public class Rest extends StatusMove
{
    public Rest(){
        super(Type.NORMAL,0,0);

    }

    @Override
    protected void applySelfEffects(Pokemon pokemon){
        Effect e = new Effect().condition(Status.SLEEP).turns(2);
        pokemon.restore();
        pokemon.addEffect(e);

        super.applySelfEffects(pokemon);
    }

    @Override
    protected String describe()
    {
        return "Применяет Rest";
    }
}
```

Class TailWhip

```
package attacks.statusAttacks;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.StatusMove;
import ru.ifmo.se.pokemon.Type;

public class TailWhip extends StatusMove
{
    public TailWhip ()
    {
        super(Type.NORMAL, 0, 100);
    }

    @Override
    protected void applyOppEffects(Pokemon pokemon)
    {
        pokemon.setMod(Stat.DEFENSE, -2);

        super.applyOppEffects(pokemon);
    }

    @Override
    protected String describe()
    {
        return "Применяет Tail Whip";
    }
}
```

Диаграмма классов



UML-формат в репозитории.

Результаты работы программы

Miltank Счастливая корова из команды полосатых вступает в бой!

Psyduck Утка-кря из команды зеленых вступает в бой!

Miltank Счастливая корова применяет Body Slam.

Psyduck Утка-кря теряет 26 здоровья.

Psyduck Утка-кря парализован

Psyduck Утка-кря применяет Facade.

Miltank Счастливая корова теряет 8 здоровья.

Miltank Счастливая корова применяет Tackle.

Psyduck Утка-кря теряет 14 здоровья.

Psyduck Утка-кря борется с соперником.

Miltank Счастливая корова теряет 7 здоровья.

Psyduck Утка-кря теряет 2 здоровья.

Miltank Счастливая корова применяет Thunder.

Psyduck Утка-кря теряет 18 здоровья.

Psyduck Утка-кря Применяет Amnesia.

Psyduck Утка-кря увеличивает специальную защиту.

Miltank Счастливая корова применяет Thunder.

Psyduck Утка-кря теряет 16 здоровья.

Psyduck Утка-кря теряет сознание.

Lotad Кувшинка из команды зеленых вступает в бой!

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка теряет 9 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Thunder.

Критический удар!

Lotad Кувшинка теряет 20 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Thunder.

Lotad Кувшинка теряет 6 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка восстанавливает 1 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка восстанавливает 1 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка восстанавливает 1 здоровья.

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка восстанавливает 1 здоровья.

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка восстанавливает 1 здоровья.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка восстанавливает 1 здоровья.

Lotad Кувшинка борется с соперником.

Miltank Счастливая корова теряет 8 здоровья.

Lotad Кувшинка теряет 2 здоровья.

Miltank Счастливая корова применяет Body Slam.

Lotad Кувшинка восстанавливает 1 здоровья.

Lotad Кувшинка парализован

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Thunder.

Lotad Кувшинка теряет 9 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка восстанавливает 1 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 1 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Thunder.

Lotad Кувшинка теряет 10 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 2 здоровья.

Lotad Кувшинка борется с соперником.

Miltank Счастливая корова теряет 12 здоровья.

Lotad Кувшинка теряет 3 здоровья.

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка теряет 2 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Body Slam.

Критический удар!

Lotad Кувшинка теряет 4 здоровья.

Lotad Кувшинка борется с соперником.

Miltank Счастливая корова теряет 12 здоровья.

Lotad Кувшинка теряет 3 здоровья.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 2 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 3 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Rock Slide.

Критический удар!

Lotad Кувшинка теряет 2 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка теряет 1 здоровья.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 4 здоровья.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 3 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Body Slam.

Lotad Кувшинка теряет 2 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Body Slam.

Lotad Кувшинка теряет 2 здоровья.

Lotad Кувшинка промахивается

Miltank Счастливая корова применяет Body Slam.

Lotad Кувшинка теряет 1 здоровья.

Lotad Кувшинка борется с соперником.

Miltank Счастливая корова теряет 9 здоровья.

Lotad Кувшинка теряет 2 здоровья.

Miltank Счастливая корова применяет Rock Slide.

Lotad Кувшинка теряет 1 здоровья.

Lotad Кувшинка Применяет Growl.

Miltank Счастливая корова уменьшает атаку.

Miltank Счастливая корова применяет Rock Slide.

Критический удар!

Lotad Кувшинка теряет 3 здоровья.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 4 здоровья.

Miltank Счастливая корова применяет Tackle.

Lotad Кувшинка теряет 4 здоровья.

Lotad Кувшинка теряет сознание.

Ludicolo Яйцешинка из команды зеленых вступает в бой!

Miltank Счастливая корова применяет Body Slam.

Ludicolo Яйцешинка теряет 2 здоровья.

Ludicolo Яйцешинка применяет Mega Drain.

Miltank Счастливая корова теряет 17 здоровья.

Ludicolo Яйцешинка восстанавливает 8 здоровья.

Miltank Счастливая корова теряет сознание.

Golduck Монстрик из команды полосатых вступает в бой!

Ludicolo Яйцешинка применяет Zen Headbutt.

Golduck Монстрик теряет 19 здоровья.

Golduck Монстрик Применяет Amnesia.

Golduck Монстрик увеличивает специальную защиту.

Ludicolo Яйцешинка применяет Mega Drain.

Golduck Монстрик теряет 32 здоровья.

Ludicolo Яйцешинка восстанавливает 16 здоровья.

Golduck Монстрик теряет сознание.

Lombre Кувшимби из команды полосатых вступает в бой!

Lombre Кувшимби Применяет Growl.

Ludicolo Яйцешинка уменьшает атаку.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби борется с соперником.

Критический удар!

Ludicolo Яйцешинка теряет 29 здоровья.

Lombre Кувшимби теряет 7 здоровья.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби восстанавливает 1 здоровья.

Lombre Кувшимби Применяет Growl.

Ludicolo Яйцешинка уменьшает атаку.

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби промахивается

Ludicolo Яйцешинка применяет Mega Drain.

Lombre Кувшимби теряет 10 здоровья.

Ludicolo Яйцешинка восстанавливает 5 здоровья.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби Применяет Growl.

Ludicolo Яйцешинка уменьшает атаку.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби промахивается

Ludicolo Яйцешинка применяет Zen Headbutt.

Критический удар!

Lombre Кувшимби теряет 3 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 6 здоровья.

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби Применяет Growl.

Ludicolo Яйцешинка уменьшает атаку.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби промахивается

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка применяет Mega Drain.

Lombre Кувшимби теряет 7 здоровья.

Ludicolo Яйцешинка восстанавливает 3 здоровья.

Lombre Кувшимби Применяет Growl.

Ludicolo Яйцешинка уменьшает атаку.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 1 здоровья.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби Применяет Growl.

Ludicolo Яйцешинка уменьшает атаку.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби борется с соперником.

Критический удар!

Ludicolo Яйцешинка теряет 2 здоровья.

Lombre Кувшимби теряет 1 здоровья.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Критический удар!

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка применяет Mega Drain.

Lombre Кувшимби теряет 11 здоровья.

Ludicolo Яйцешинка восстанавливает 5 здоровья.

Lombre Кувшимби промахивается

Ludicolo Яйцешинка применяет Mega Drain.

Lombre Кувшимби теряет 7 здоровья.

Ludicolo Яйцешинка восстанавливает 3 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Критический удар!

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка применяет Mega Drain.

Lombre Кувшимби теряет 12 здоровья.

Ludicolo Яйцешинка восстанавливает 6 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 1 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка применяет Mega Drain.

Lombre Кувшимби теряет 10 здоровья.

Ludicolo Яйцешинка восстанавливает 5 здоровья.

Lombre Кувшимби промахивается

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка применяет Mega Drain.

Lombre Кувшимби теряет 11 здоровья.

Ludicolo Яйцешинка восстанавливает 5 здоровья.

Lombre Кувшимби Применяет Growl.

Ludicolo Яйцешинка уменьшает атаку.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби борется с соперником.

Критический удар!

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби промахивается

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби промахивается

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби промахивается

Ludicolo Яйцешинка Применяет Growl.

Lombre Кувшимби уменьшает атаку.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби промахивается

Ludicolo Яйцешинка промахивается

Lombre Кувшимби промахивается

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка применяет Zen Headbutt.

Lombre Кувшимби теряет 2 здоровья.

Lombre Кувшимби применяет Zen Headbutt.

Ludicolo Яйцешинка восстанавливает 1 здоровья.

Ludicolo Яйцешинка промахивается

Lombre Кувшимби борется с соперником.

Ludicolo Яйцешинка теряет 1 здоровья.

Ludicolo Яйцешинка применяет Mega Drain.

Критический удар!

Lombre Кувшимби теряет 21 здоровья.

Ludicolo Яйцешинка восстанавливает 10 здоровья.

Lombre Кувшимби теряет сознание.

В команде полосатых не осталось покемонов.

Команда зеленых побеждает в этом бою!

Дополнительное задание

Условие

1. Реализовать отбор покемонов по уровню: если у покемона уровень, который не входит в промежуток 1-100, то он исключается из боя. В случае, если ни один покемон не подходит, то бой не состоится.
2. Попробовать сделать равное количество переносов строк после каждой хода в бое.

Исходный код

Class main

```
import ru.ifmo.se.pokemon.*;
import pokemons.*;

public class main{
    public static void main(String[] args)
    {

        Miltank miltank = new Miltank("Счастливая корова", 3434);
        Psyduck psyduck = new Psyduck("Утка-кря", 101);
        Golduck golduck = new Golduck("Монстрик", -1);
        Lotad lotad = new Lotad("Кувшинка", 101);
        Lombre lombre = new Lombre("Кувшимби", 101);
        Ludicolo ludicolo = new Ludicolo("Яйцешинка", 3434);

        System.out.println();

        New_Pokemon[] Foe = {miltank, psyduck, golduck};
        New_Pokemon[] Ally = {lotad, lombre, ludicolo};

        New_Battle battle = new New_Battle(Foe, Ally);
        battle.go();
    }
}
```

Class New_Pokemon (все покемоны наследуются от этого класса)

```
package pokemons;

import ru.ifmo.se.pokemon.Pokemon;

public class New_Pokemon extends Pokemon
{
    private boolean correct_lvl = true;

    public New_Pokemon(String name, int lvl)
    {
        super(name, lvl);
        if (lvl > 100 || lvl < 1)
        {
            correct_lvl = false;
            System.out.printf("%s был исключен из боя, потому что lvl
```

```
должен лежать в промежутке 1-100\n", name);
        }
    }

    public boolean get_correct_lvl (){
        return correct_lvl;
    }
}
```

Class New_Battle

```
import pokemons.New_Pokemon;
import ru.ifmo.se.pokemon.Battle;

public class New_Battle
{
    Battle b = new Battle();

    private int Count_Ally;
    private int Count_Foe;

    public New_Battle (New_Pokemon[] Ally, New_Pokemon[] Foe)
    {
        for (New_Pokemon poke : Ally)
        {
            if (poke.get_correct_lvl())
            {
                b.addAlly(poke);
                Count_Ally++;
            }
        }
        for (New_Pokemon poke : Foe)
        {
            if (poke.get_correct_lvl())
            {
                b.addFoe(poke);
                Count_Foe++;
            }
        }
    }

    public void go()
    {
        if (Count_Ally == 0 && Count_Foe == 0)
        {
            System.out.println("Невозможно начать бой. В каждой команде 0
покемонов :(");
        }
        else if (Count_Foe == 0 || Count_Ally == 0)
        {
            System.out.println("Невозможно начать бой. В одной из команд
нет покемонов :(");
```

```
            }
            else
            {
                System.out.println("Бой возможен, все условия выполнены
:)\n");
                b.go();
            }
        }

}
```

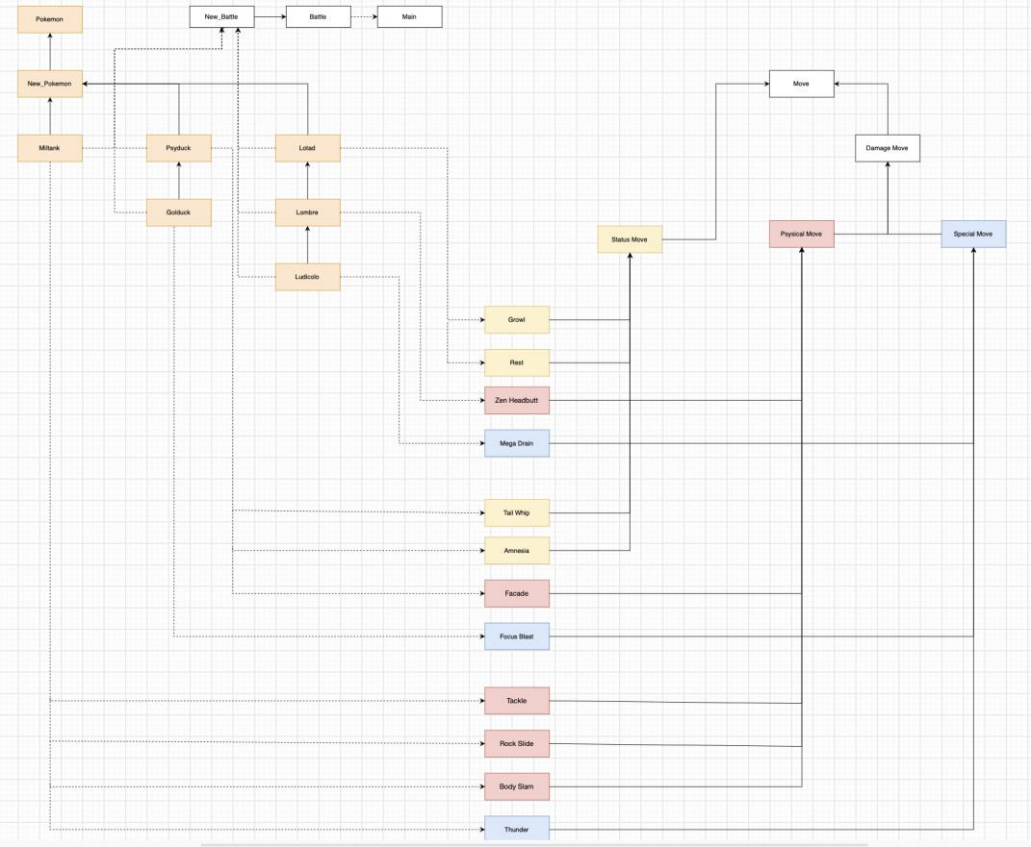
Объяснение

Для реализации 1 задание было написано 2 дополнительных класса: New_Pokemon, New_Battle, и незначительно переписаны существующие классы: main, классы всех существующих покемонов.

При создании нового покемона его уровень проходит проверку в классе New_Pokemon, результат проверки сохраняется в переменную boolean correct_lvl. Так же в этом классе реализован геттер, с помощью которого можно узнать, является ли уровень покемона корректным для участия в бое.

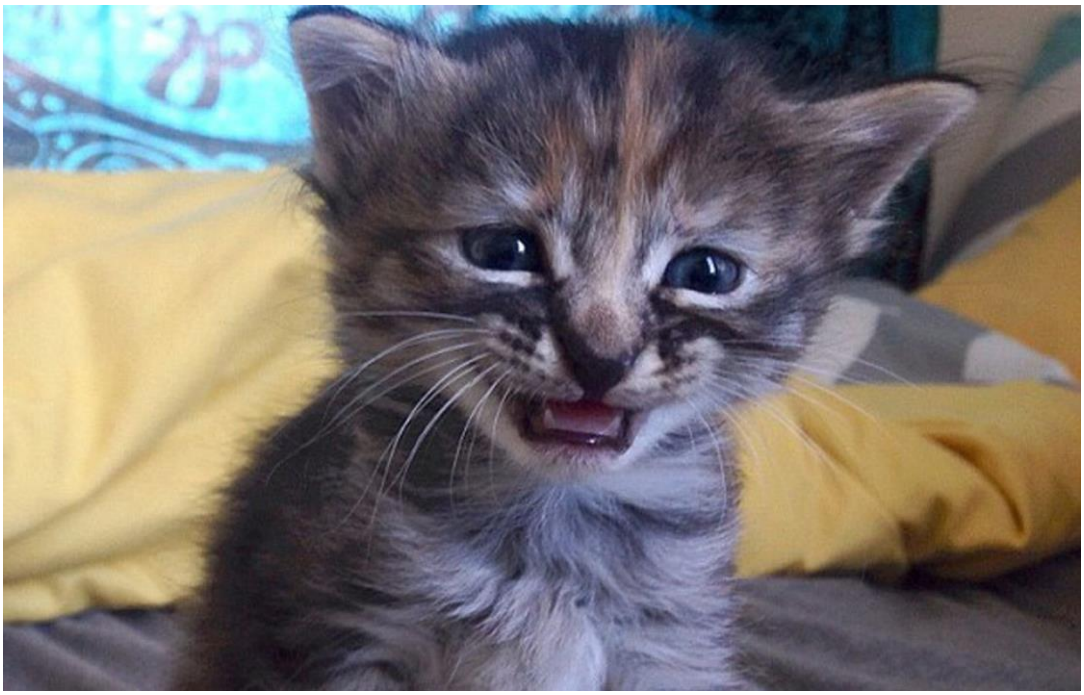
Затем для правильного распределения по командам был прописан новый класс New_Battle. Внутри него создается объект боя, реализованный в уже прописанном классе Battle. После чего происходит распределение покемонов по командам: цикл проходится по двум командам и для каждого покемона проходит проверка по уровню (для этого используется информация из get_correct_lvl). После чего исключаются варианты начала боя, если: в двух командах нет ни одного покемона, в одной из команд нет ни одного покемона. После чего запускается бой.

Диаграмма классов после изменений



Объяснение #2

Чтобы добиться одинакового количества переносов строк между атаками нужно было бы переписать класс Pokemon, потому что переносы строк реализованы в самой системе боя. К тому же метод attack, в котором выводятся переносы строк, является final, что делает невозможным реализовать поставленную задачу.



Вывод

Во время выполнения данной лабораторной работы я изучил основы ООП. После чего применил полученные знания на практике.

В процессе выполнения работы я множество раз прочитал документацию о работе внешней библиотеки, которую подключил к собственному проекту.

Научился собирать jar-файл полноценного проекта со внешней библиотекой.

Полученные знания будут использованы мной в дальнейшем изучении языка программирования Java.