



Факультет Систем управления и робототехники
“Программирование”

Лабораторная работа №2
Вариант 31682

Преподаватель: Сорокин Роман Борисович
Выполнил: Федяков Григорий Дмитриевич
Группа: R3142

Санкт-Петербург

2020

Задание:

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.

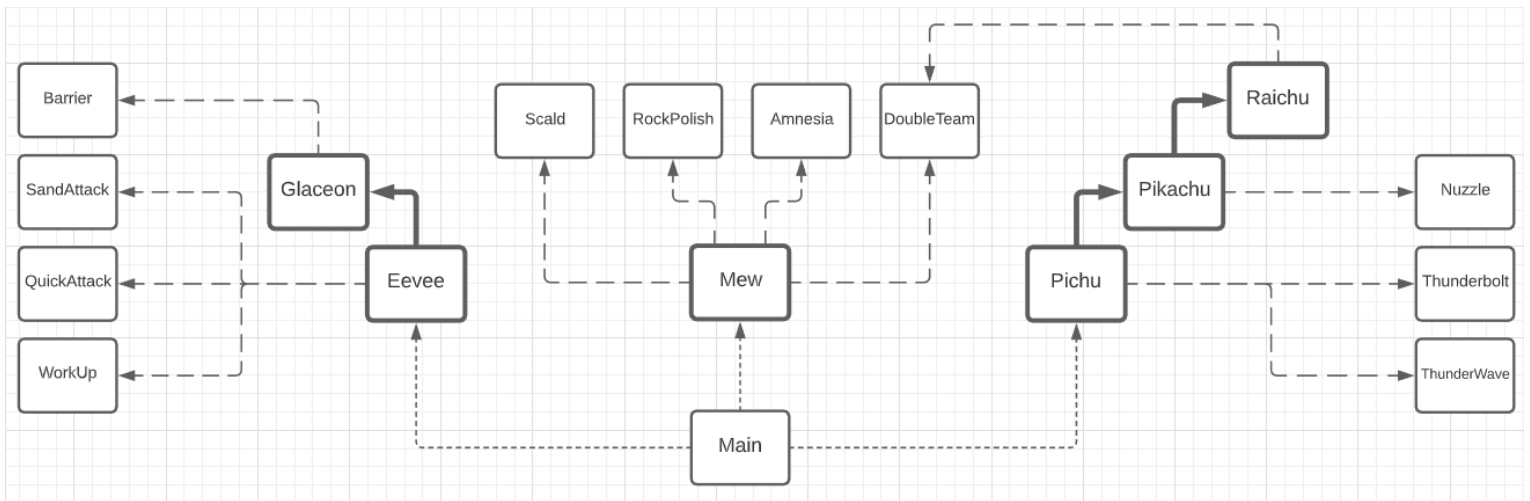
Базовые классы и симулятор сражения находятся в [jar-архиве](#) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](#).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах <http://poke-universe.ru>, <http://pokemondb.net>, <http://veekun.com/dex/pokemon>

Покемоны

<div>Mew</div> <div></div> <div>Атаки:<ul style="list-style-type: none">ScaldDouble TeamAmnesiaRock Polish</div>	<div>Eevee</div> <div></div> <div>Атаки:<ul style="list-style-type: none">Work UpSand AttackQuick Attack</div>	<div>Glaceon</div> <div></div> <div>Атаки:<ul style="list-style-type: none">Work UpSand AttackQuick AttackBarrier</div>	<div>Pichu</div> <div></div> <div>Атаки:<ul style="list-style-type: none">Thunder WaveThunderbolt</div>	<div>Pikachu</div> <div></div> <div>Атаки:<ul style="list-style-type: none">Thunder WaveThunderboltNuzzle</div>	<div>Raichu</div> <div></div> <div>Атаки:<ul style="list-style-type: none">Thunder WaveThunderboltNuzzleDouble Team</div>
--	--	--	--	---	--

Диаграмма классов реализованной объектной модели



Исходный код:

Main.java

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

public class Main {
    public static void main(String[] args) {
        Battle b = new Battle();
        Pokemon misha = new Mew ("Misha",270);
        Pokemon igor = new Eevee ("Igor",65);
        Pokemon maxim = new Glaceon("Maxim",184);
        Pokemon sanya = new Pichu ("Sanya",41);
        Pokemon sasha = new Pikachu ("Sasha",112);
        Pokemon alexandr = new Raichu ("Alexandr",218);

        b.addAlly(misha);
        b.addFoe(sanya);
        b.addAlly(igor);
        b.addFoe(sasha);
        b.addAlly(maxim);
        b.addFoe(alexandr);
        b.go();
    }
}
```

Mew.java

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

public class Mew extends Pokemon {
    Mew(String name, int level){
        super(name,level);
        setType(Type.ELECTRIC);
        setStats(100,100,100,100,100,100);
        addMove(new Scald());
        addMove(new DoubleTeam());
        addMove(new Amnesia());
        addMove(new RockPolish());
    }
}
```

Eevee.java

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

public class Eevee extends Pokemon {
    Eevee(String name, int level){
        super(name,level);
        setType(Type.NORMAL);
        setStats(55,55,50,45,65,55);
        addMove(new WorkUp());
        addMove(new SandAttack());
        addMove(new QuickAttack());
    }
}
```

Glaceon.java

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

public class Glaceon extends Eevee {
    Glaceon(String name, int level){
        super(name,level);
        setType(Type.ICE);
        setStats(65,60,110,130,95,65);
        addMove(new WorkUp());
        addMove(new SandAttack());
        addMove(new QuickAttack());
        addMove(new Barrier());
    }
}
```

Pichu.java

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

public class Pichu extends Pokemon {
    Pichu(String name, int level){
        super(name,level);
        setType(Type.ELECTRIC);
        setStats(20,40,15,35,35,60);
        addMove(new ThunderWave());
        addMove(new Thunderbolt());
    }
}
```

Pikachu.java

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

public class Pikachu extends Pichu {
    Pikachu(String name, int level){
        super(name,level);
        setStats(35,55,40,50,50,90);
        addMove(new Nuzzle());
    }
}
```

Raichu.java

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

public class Raichu extends Pikachu {
    Raichu(String name, int level){
        super(name,level);
        setStats(50,90,55,90,80,110);
        addMove(new DoubleTeam());
    }
}
```

ThunderWave.java

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

public class ThunderWave extends StatusMove{
    ThunderWave () {
        super(Type.ELECTRIC,0,90);
    }
    protected String describe() {
        return "paralyzed";
    }
    private final Effect bolt = new Effect().chance(1).condition(Status.PARALYZE);
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        p.addEffect(bolt);
    }
}
```

Thunderbolt.java

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

public class Thunderbolt extends SpecialMove{
    Thunderbolt () {
        super(Type.ELECTRIC,90,100);
    }
    protected String describe() {
        return "strongly paralyzed";
    }
    private final Effect bolt = new Effect().chance(0.1).condition(Status.PARALYZE);
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        p.addEffect(bolt);
    }
}
```

Nuzzle.java

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

public class Nuzzle extends PhysicalMove{
    Nuzzle () {
        super(Type.ELECTRIC,20,100);
    }
    protected String describe() {
        return "nuzzled";
    }
    private final Effect bolt = new Effect().chance(1).condition(Status.PARALYZE);
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        p.addEffect(bolt);
    }
}
```

DoubleTeam.java

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

public class DoubleTeam extends StatusMove{
    DoubleTeam () {
        super(Type.NORMAL,0,0);
    }
    protected String describe() {
        return "moved";
    }

    protected void applySelfEffects(Pokemon p) {
        p.setMod(Stat.EVASION, +1);
    }
}
```

Amnesia.java

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

public class Amnesia extends StatusMove{
    Amnesia () {
        super(Type.PSYCHIC,0,0);
    }
    protected String describe() {
        return "remembered";
    }
    protected void applySelfEffects(Pokemon p) {
        p.setMod(Stat.DEFENSE, +2);
    }
}
```

RockPolish.java

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

public class RockPolish extends StatusMove{
    RockPolish () {
        super(Type.ROCK,0,0);
    }
    protected String describe() {
        return "speeded up";
    }
    protected void applySelfEffects(Pokemon p) {
        p.setMod(Stat.SPEED, +2);
    }
}
```

Scald.java

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

public class Scald extends SpecialMove{
    Scald () {
        super(Type.WATER,80,100);
    }
    protected String describe() {
        return "scalded";
    }
    private final Effect burn = new Effect().chance(0.3).condition(Status.BURN);
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        p.addEffect(burn);
    }
}
```

WorkUp.java

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

public class WorkUp extends StatusMove{
    WorkUp () {
        super(Type.NORMAL,80,100);
    }
    protected String describe() {
        return "worked up";
    }
    protected void applySelfEffects(Pokemon p) {
        p.setMod(Stat.SPECIAL_ATTACK, +1);
        p.setMod(Stat.ATTACK, +1);
    }
}
```

QuickAttack.java

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

public class QuickAttack extends PhysicalMove{
    QuickAttack () {
        super(Type.NORMAL,40,100,1,0);
    }
    protected String describe() {
        return "quickly attacked";
    }
}
```

SandAttack.java

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

public class SandAttack extends StatusMove{
    SandAttack () {
        super(Type.GROUND,0,100);
    }
    protected String describe() {
        return "blinded his enemy by the lights";
    }
    protected void applyOppEffects(Pokemon p) {
        p.setMod(Stat.ACCURACY, -1);
    }
}
```

Barrier.java

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

public class Barrier extends StatusMove{
    Barrier () {
        super(Type.PSYCHIC,0,0);
    }
    protected String describe() {
        return "defended";
    }
    protected void applySelfEffects(Pokemon p) {
        p.setMod(Stat.DEFENSE, 2);
    }
}
```

Результаты работы:

Mew Misha из команды синих вступает в бой!

Pichu Sanya из команды черных вступает в бой!

Mew Misha промахивается

Pichu Sanya борется с соперником.

Mew Misha теряет 5 здоровья.

Pichu Sanya теряет 1 здоровья.

Mew Misha scalded.

Критический удар!

Pichu Sanya теряет 300 здоровья.

Pichu Sanya теряет сознание.

Pikachu Sasha из команды черных вступает в бой!

Mew Misha промахивается

Pikachu Sasha paralyzed.

Mew Misha парализован

Mew Misha scalded.

Pikachu Sasha теряет 71 здоровья.

Pikachu Sasha борется с соперником.

Критический удар!

Mew Misha теряет 23 здоровья.

Pikachu Sasha теряет 6 здоровья.

Mew Misha промахивается

Pikachu Sasha strongly paralyzed.

Mew Misha теряет 16 здоровья.

Mew Misha промахивается

Pikachu Sasha strongly paralyzed.

Mew Misha теряет 12 здоровья.

Mew Misha промахивается

Pikachu Sasha paralyzed.

Mew Misha промахивается

Pikachu Sasha paralyzed.

Mew Misha промахивается

Pikachu Sasha борется с соперником.

Mew Misha теряет 20 здоровья.

Pikachu Sasha теряет 5 здоровья.

Mew Misha scalded.

Критический удар!

Pikachu Sasha теряет 86 здоровья.

Pikachu Sasha воспламеняется

Pikachu Sasha strongly paralyzed.

Критический удар!

Mew Misha теряет 32 здоровья.

Mew Misha промахивается

Pikachu Sasha nuzzled.

Критический удар!

Mew Misha теряет 11 здоровья.

Mew Misha scalded.

Pikachu Sasha теряет 46 здоровья.

Pikachu Sasha теряет сознание.

Raichu Alexandr из команды черных вступает в бой!

Raichu Alexandr nuzzled.

Mew Misha теряет 7 здоровья.

Mew Misha scalded.

Raichu Alexandr теряет 35 здоровья.

Raichu Alexandr nuzzled.

Критический удар!

Mew Misha теряет 16 здоровья.

Mew Misha промахивается

Raichu Alexandr strongly paralyzed.

Критический удар!

Mew Misha теряет 44 здоровья.

Mew Misha промахивается

Raichu Alexandr nuzzled.

Критический удар!

Mew Misha теряет 16 здоровья.

Mew Misha scalded.

Raichu Alexandr теряет 32 здоровья.

Raichu Alexandr промахивается

Mew Misha промахивается

Raichu Alexandr paralyzed.

Mew Misha промахивается

Raichu Alexandr nuzzled.

Mew Misha теряет 8 здоровья.

Mew Misha промахивается

Raichu Alexandr промахивается

Mew Misha промахивается

Raichu Alexandr paralyzed.

Mew Misha промахивается

Raichu Alexandr промахивается

Mew Misha промахивается

Raichu Alexandr промахивается

Mew Misha промахивается

Raichu Alexandr strongly paralyzed.

Mew Misha теряет 28 здоровья.

Mew Misha промахивается

Raichu Alexandr strongly paralyzed.

Mew Misha теряет 21 здоровья.

Mew Misha scalded.

Критический удар!

Raichu Alexandr теряет 89 здоровья.

Raichu Alexandr paralyzed.

Mew Misha промахивается

Raichu Alexandr paralyzed.

Mew Misha scalded.

Raichu Alexandr теряет 39 здоровья.

Raichu Alexandr воспламеняется

Raichu Alexandr промахивается

Mew Misha scalded.

Raichu Alexandr теряет 58 здоровья.

Raichu Alexandr теряет сознание.

В команде черных не осталось покемонов.

Команда синих побеждает в этом бою!

Вывод:

Во время выполнения лабораторной работы я изучил объектно-ориентированный подход программирования на языке Java и научился использовать сторонние библиотеки в написании новых программ. Эти фундаментальные знания пригодятся мне в будущем для разработки сложных проектов.