

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

**Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»**

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 2

**по дисциплине
‘ПРОГРАММИРОВАНИЕ’**

Вариант №134117

**Выполнил:
Студент группы Р3134
Лазарев Марк Олегович
Принял:
Кулинич Я. В.**



**Санкт-Петербург
2022**

ОГЛАВЛЕНИЕ:

• Текст задания.....	3
• Исходный код.....	4
• Результат работы.....	5
• Вывод.....	5

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](#).

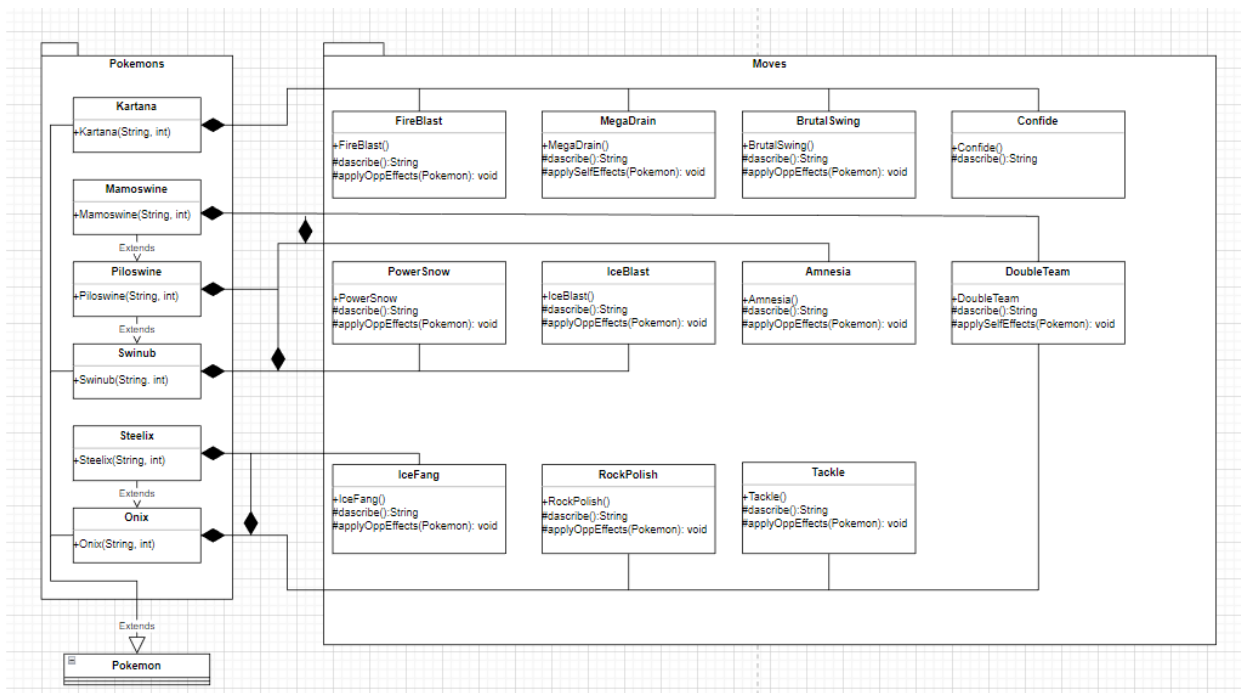
Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах <http://poke-universe.ru>, <http://pokemondb.net>, <http://veekun.com/dex/pokemon>

Покемоны:

Ваши покемоны:

<p>Kartana</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Fire Blast✓ Mega Drain✓ Brutal Swing✓ Confide	<p>Onix</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Tackle✓ Rock Polish✓ Double Team	<p>Steelix</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Tackle✓ Rock Polish✓ Double Team✓ Ice Fang	<p>Swinub</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Powder Snow✓ Ice Beam	<p>Piloswine</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Powder Snow✓ Ice Beam✓ Amnesia	<p>Mamoswine</p>  <p>Атаки:</p> <ul style="list-style-type: none">✓ Powder Snow✓ Ice Beam✓ Amnesia✓ Double Team
---	--	--	---	---	---

Диаграмма классов:



Исходный код:

Proga.java "Main"

```

package me.mark.lab2;
import me.mark.lab2.pokemons.*;
import ru.ifmo.se.pokemon.Battle;

public class Proga {
    // https://pokedex.net/pokedex/kartana
    // https://pokedex.net/pokedex/onix
    // https://pokedex.net/pokedex/steelix
    // https://pokedex.net/pokedex/swinub
    // https://pokedex.net/pokedex/piloswine
    // https://pokedex.net/pokedex/mamoswine
    public static void main(String[] args) {
        Battle b = new Battle();

        Onix onix = new Onix("", 27);
        Kartana kartana = new Kartana("", 2000);
        Mamoswine mamoswine = new Mamoswine("", 23);

        Piloswine piloswine = new Piloswine("", 19);
        Steelix steelix = new Steelix("", 35);
        Swinub swinub = new Swinub("", 24);

        b.addAlly(onix);
        b.addAlly(kartana);
        b.addAlly(mamoswine);
    }
}

```

```

        b.addFoe(piloswine);
        b.addFoe(swinub);
        b.addFoe(steelix);

        b.go();
    }

    public static boolean chance (double r) {
        return r > Math.random();
    }
}

```

Moves->PhysicalMove

BrutalSwing

```

package me.mark.lab2.pokemons;

import me.mark.lab2.moves.PhysicalMove.BrutalSwing;
import me.mark.lab2.moves.SpecialMove.FireBlast;
import me.mark.lab2.moves.SpecialMove.MegaDrain;
import me.mark.lab2.moves.StatusMove.Confide;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Kartana extends Pokemon {

    public Kartana(String name, int level) {
        super(name, level);

        super.setType(Type.GRASS, Type.STEEL);
        super.setStats(59, 181, 131, 59, 31, 109);

        FireBlast fireBlast = new FireBlast(110, 85);
        MegaDrain megaDrain = new MegaDrain(40, 100);
        BrutalSwing brutalSwing = new BrutalSwing(60, 100);
        Confide confide = new Confide(0, 0);

        super.setMove(fireBlast, megaDrain, brutalSwing, confide);
    }
}

```

IceFang

```

package me.mark.lab2.moves.PhysicalMove;

import me.mark.lab2.Proga;
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 IceFang extends PhysicalMove {

    public IceFang(double pow, double acc) {
        super(Type.ICE, pow, acc);
    }
}

```

```

    }
    @Override
    protected void applyOppEffects(Pokemon e) {
        //вызываем родительский порядок эвектов заложенных в данную
        способность
        super.applyOppEffects(e);

        //затем проверяем 10% шанс (мафс рандом проверяет от 0 до 1) и в
        случае попадания в 0.1 прога вернёт True
        if(Proga.chance(0.1)) {
            Effect.flinch(e);
            Effect.freeze(e);
        }
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length - 1];
    }
}

```

Tackle

```

package me.mark.lab2.moves.PhysicalMove;

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

public class Tackle extends PhysicalMove {

    public Tackle(double power, double acc) {
        super(Type.NORMAL, power, acc);
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length-1];
    }
}

```

Moves->SpecialMove

FireBlast

```

package me.mark.lab2.moves.SpecialMove;

import me.mark.lab2.Proga;
import ru.ifmo.se.pokemon.Effect;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Type;

```

```

public class FireBlast extends SpecialMove {
    public FireBlast(double power, double acc) {
        super(Type.FIRE, power, acc);
    }

    @Override
    //метод applyOppEffects вызывается когда мы применяем эффекты на соперника
    protected void applyOppEffects(Pokemon e){
        //вызываем родительский порядок эвектов заложенных в данную
        способность
        super.applyOppEffects(e);

        //затем проверяем 10% шанс (мафс рандом проверяет от 0 до 1) и в
        случае попадания в 0.1 прога вернёт True
        if(Proga.chance(0.1)) {
            Effect.burn(e);
        }
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length - 1];
    }
}

```

IceBeam

```

package me.mark.lab2.moves.SpecialMove;

import me.mark.lab2.Proga;
import ru.ifmo.se.pokemon.Effect;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Type;

public class IceBeam extends SpecialMove {

    public IceBeam(double pow, double acc) {
        super(Type.ICE, pow, acc);
    }

    @Override
    protected void applyOppEffects(Pokemon e){
        super.applyOppEffects(e);

        if(Proga.chance(0.1))
            Effect.freeze(e);
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length-1];
    }
}

```

MegaDrain

```
package me.mark.lab2.moves.SpecialMove;

import ru.ifmo.se.pokemon.*;

public class MegaDrain extends SpecialMove {

    public MegaDrain(double power, double acc) {
        super(Type.GRASS, power, acc);
    }

    @Override
    protected void applySelfDamage(Pokemon p, double damage) {
        p.setMod(Stat.HP, -(int) (Math.round(damage)) / 2);
    }

    @Override
    protected String describe() {
        return "deals damage and the user will recover 50% of the HP drained.";
    }

}
```

PowerSnow

```
package me.mark.lab2.moves.SpecialMove;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Type;

public class PowerSnow extends SpecialMove {

    public PowerSnow(double pow, double acc) {
        super(Type.ICE, pow, acc);
    }

    @Override
    protected void applyOppEffects(Pokemon e) {
        super.applyOppEffects(e);
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        //последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length-1];
    }

}
```

Moves->StatusMove

Amnesia

```
package me.mark.lab2.moves.StatusMove;
import ru.ifmo.se.pokemon.*;
```



```

public class Amnesia extends StatusMove {

    public Amnesia(double pow, double acc){
        super(Type.PSYCHIC, pow, acc);
    }
    @Override
    protected void applyOppEffects(Pokemon e){
        super.applyOppEffects(e);

        Effect s = new Effect().stat(Stat.SPECIAL_DEFENSE, 2);
        e.addEffect(s);
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        //последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length-1];
    }
}

```

Confide

```

package me.mark.lab2.moves.StatusMove;

import ru.ifmo.se.pokemon.*;

public class Confide extends StatusMove {

    public Confide(double power, double acc) {
        super(Type.NORMAL, power, acc);
    }
    @Override
    protected void applyOppEffects(Pokemon e){
        super.applyOppEffects(e);

        Effect p = new Effect().stat(Stat.ATTACK, -2);
        e.addEffect(p);
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        //последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length-1];
    }
}

```

DoubleTeam

```

package me.mark.lab2.moves.StatusMove;

import ru.ifmo.se.pokemon.*;

```

```

public class DoubleTeam extends StatusMove {

    public DoubleTeam(double power, double acc) {
        super(Type.NORMAL, power, acc);
    }

    @Override
    protected void applyOppEffects(Pokemon e) {
        super.applyOppEffects(e);

        Effect p = new Effect().stat(Stat.EVASION, 1);
        e.addEffect(p);

    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        //последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length - 1];
    }

}

```

RockPolish

```

package me.mark.lab2.moves.StatusMove;

import ru.ifmo.se.pokemon.*;

public class RockPolish extends StatusMove {

    public RockPolish(double power, double acc) {
        super(Type.ROCK, power, acc);
    }

    @Override
    protected void applyOppEffects(Pokemon e){
        super.applyOppEffects(e);
        Effect p = new Effect().stat(Stat.SPEED,6);

        e.addEffect(p);
    }

    @Override
    protected String describe() {
        //название класса, стриговое, делим на отдельные кусочки и отбираем
        //последний кусочек с названием атаки
        String[] opokemone = this.getClass().toString().split("\\.");
        return "does" + " " + opokemone[opokemone.length-1];
    }

}

```

Pokemons

Kartana

```

package me.mark.lab2.pokemons;

import me.mark.lab2.moves.PhysicalMove.BrutalSwing;
import me.mark.lab2.moves.SpecialMove.FireBlast;
import me.mark.lab2.moves.SpecialMove.MegaDrain;
import me.mark.lab2.moves.StatusMove.Confide;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Kartana extends Pokemon {

    public Kartana(String name, int level) {
        super(name, level);

        super.setType(Type.GRASS, Type.STEEL);
        super.setStats(59, 181, 131, 59, 31, 109);

        FireBlast fireBlast = new FireBlast(110, 85);
        MegaDrain megaDrain = new MegaDrain(40,100);
        BrutalSwing brutalSwing = new BrutalSwing(60,100);
        Confide confide = new Confide(0,0);

        super.setMove(fireBlast, megaDrain, brutalSwing, confide);

    }
}

```

Mamoswine

```

package me.mark.lab2.pokemons;

import me.mark.lab2.moves.SpecialMove.IceBeam;
import me.mark.lab2.moves.SpecialMove.PowerSnow;
import me.mark.lab2.moves.StatusMove.Amnesia;
import me.mark.lab2.moves.StatusMove.DoubleTeam;
import ru.ifmo.se.pokemon.Type;

public class Mamoswine extends Swinub {

    public Mamoswine(String name, int level) {
        super(name, level);

        super.setType(Type.ICE, Type.GROUND);
        super.setStats(110, 130, 80, 70, 60, 80);

        PowerSnow powerSnow = new PowerSnow(40, 100);
        IceBeam iceBeam = new IceBeam(90,100);
        Amnesia amnesia = new Amnesia(0,0);
        DoubleTeam doubleTeam = new DoubleTeam(0,0);

        super.setMove(powerSnow, iceBeam, amnesia, doubleTeam);

    }
}

```

Onix

```

package me.mark.lab2.pokemons;

import me.mark.lab2.moves.PhysicalMove.Tackle;
import me.mark.lab2.moves.StatusMove.DoubleTeam;
import me.mark.lab2.moves.StatusMove.RockPolish;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Onix extends Pokemon {

    public Onix(String name, int level) {
        super(name, level);

        super.setType(Type.ROCK, Type.GROUND);
        super.setStats(35, 45, 160, 30, 45, 70);

        Tackle tackle = new Tackle(40, 100);
        RockPolish rockPolish = new RockPolish(0, 0);
        DoubleTeam doubleTeam = new DoubleTeam(0, 0);

        super.setMove(tackle, rockPolish, doubleTeam);
    }
}

```

Poliswine

```

package me.mark.lab2.pokemons;

import me.mark.lab2.moves.SpecialMove.IceBeam;
import me.mark.lab2.moves.SpecialMove.PowerSnow;
import me.mark.lab2.moves.StatusMove.Amnesia;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Type;

public class Piloswine extends Swinub {

    public Piloswine(String name, int level) {
        super(name, level);

        super.setType(Type.ICE, Type.GROUND);
        super.setStats(100, 100, 80, 60, 50, 80);

        PowerSnow powerSnow = new PowerSnow(40, 100);
        IceBeam iceBeam = new IceBeam(90, 100);
        Amnesia amnesia = new Amnesia(0, 0);

        super.setMove(powerSnow, iceBeam, amnesia);
    }
}

```

Результаты работы:

Onix из команды зеленых вступает в бой!

Piloswine из команды синих вступает в бой!

Onix does Tackle.

Piloswine теряет 7 здоровья.

Piloswine промахивается

Onix промахивается

Piloswine does PowerSnow.

Onix теряет 20 здоровья.

Onix промахивается

Piloswine does IceBeam.

Onix теряет 32 здоровья.

Onix промахивается

Piloswine does PowerSnow.

Критический удар!

Onix теряет 38 здоровья.

Onix теряет сознание.

Kartana из команды зеленых вступает в бой!

Kartana does FireBlast.

Piloswine теряет 38 здоровья.

Piloswine does PowerSnow.

Kartana теряет 13 здоровья.

Kartana does BrutalSwing.

Piloswine теряет 15 здоровья.

Piloswine промахивается

Kartana промахивается

Piloswine does PowerSnow.

Kartana теряет 12 здоровья.

Kartana does FireBlast.

Piloswine теряет 23 здоровья.

Piloswine теряет сознание.

Swinub из команды синих вступает в бой!

Kartana does FireBlast.

Swinub теряет 28 здоровья.

Swinub does IceBeam.

Kartana теряет 29 здоровья.

Kartana does BrutalSwing.

Swinub теряет 25 здоровья.

Swinub does IceBeam.

Kartana теряет 27 здоровья.

Kartana теряет сознание.

Mamoswine из команды зеленых вступает в бой!

Mamoswine промахивается

Swinub does IceBeam.

Mamoswine теряет 17 здоровья.

Mamoswine промахивается

Swinub does PowerSnow.

Mamoswine теряет 8 здоровья.

Mamoswine промахивается

Swinub does PowerSnow.

Mamoswine теряет 9 здоровья.

Mamoswine does PowerSnow.

Swinub теряет 12 здоровья.

Swinub теряет сознание.

Steelix из команды синих вступает в бой!

Mamoswine does PowerSnow.

Steelix теряет 8 здоровья.

Steelix does Tackle.

Mamoswine теряет 16 здоровья.

Mamoswine does IceBeam.

Steelix теряет 18 здоровья.

Steelix does IceFang.

Mamoswine теряет 21 здоровья.

Mamoswine промахивается

Steelix промахивается

Mamoswine does IceBeam.

Steelix теряет 14 здоровья.

Steelix does Tackle.

Mamoswine теряет 11 здоровья.

Mamoswine does PowerSnow.

Steelix теряет 9 здоровья.

Steelix does IceFang.

Mamoswine теряет 21 здоровья.

Mamoswine теряет сознание.

В команде зеленых не осталось покемонов.

Команда синих побеждает в этом бою!

Process finished with exit code 0

Вывод:

В процессе выполнения лабораторной работы были получены навыки использования объектно-ориентированного подхода программирования при использовании языка Java.