

Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики

ФКТиУ, кафедра Вычислительной техники

Лабораторная работа №2
по дисциплине
«Программирование»

Вариант - 18065

Выполнил: Чан Куанг Лок
Группа: Р3100

Санкт-Петербург 2018 г.

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.

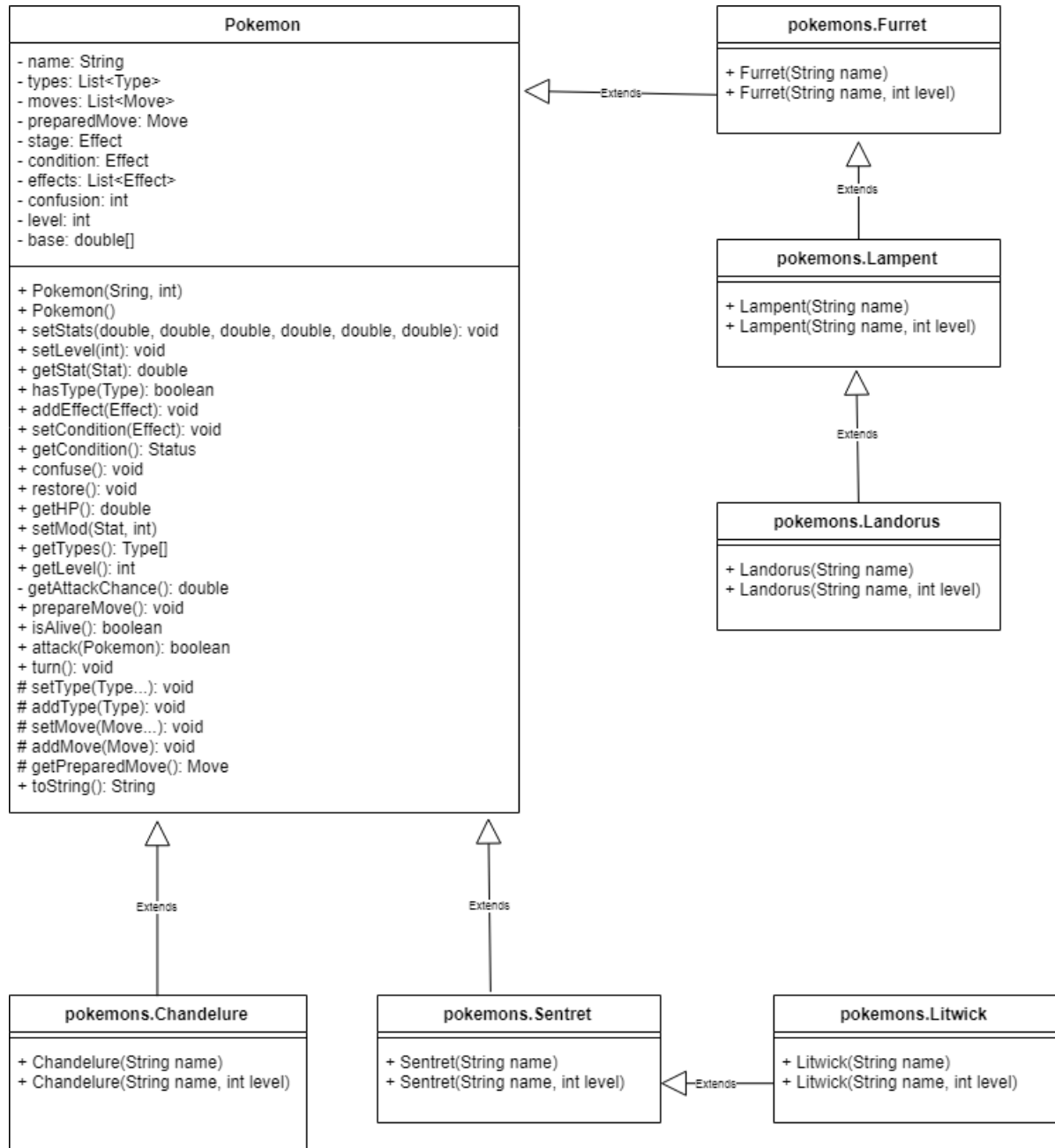
Базовые классы и симулятор сражения находятся в jar-архиве. Документация в формате javadoc - в zip-архиве.

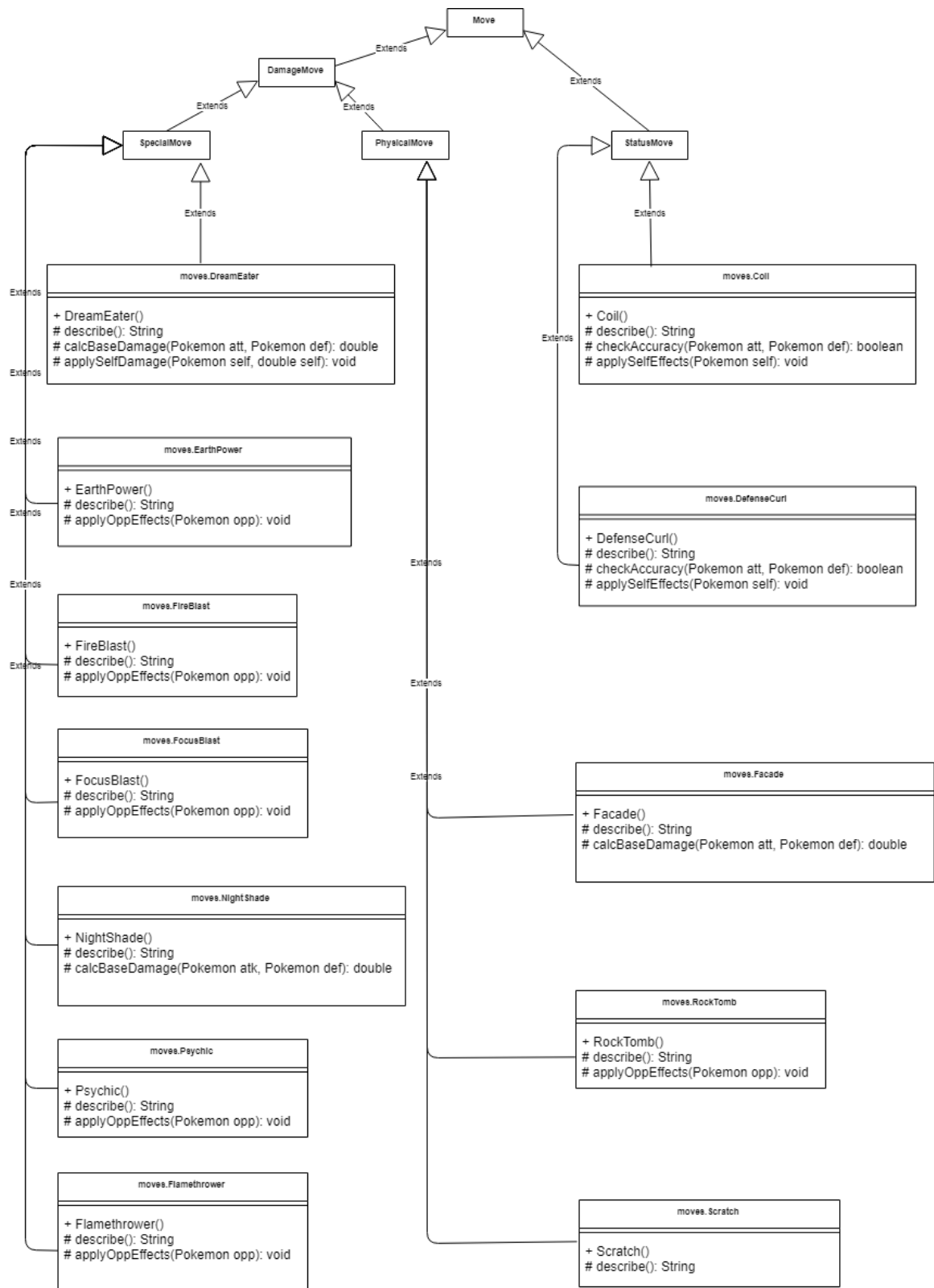
Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах <http://poke-universe.ru>, <http://pokemondb.net>, <http://veekun.com/dex/pokemon>

Вариант



UML Диаграмма





Исходный код программы

```
// ru/ifmo/se/s267880/lab2/moves/Coil.java
package ru.ifmo.se.s267880.lab2.moves;

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

public class Coil extends StatusMove {
    public Coil() {
        super(Type.POISON, 0, 1);
    }

    @Override
    protected String describe() {
        // yep, strong one
        return "Coil: Raises the user's Attack, Defense, and accuracy by one stage each";
    }

    @Override
    protected boolean checkAccuracy(Pokemon att, Pokemon def) {
        return true;
    }

    @Override
    protected void applySelfEffects(Pokemon self) {
        self.setMod(Stat.ATTACK, 1);
        self.setMod(Stat.DEFENSE, 1);
        self.setMod(Stat.ACCURACY, 1);
    }
}

// ru/ifmo/se/s267880/lab2/moves/DefenseCurl.java
package ru.ifmo.se.s267880.lab2.moves;

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

public class DefenseCurl extends StatusMove {
    public DefenseCurl() {
        super(Type.NORMAL, 0, 1);
    }

    @Override
    protected String describe() {
        return "DefenseCurl: Raises user's Defense by one stage";
    }

    @Override
    protected boolean checkAccuracy(Pokemon att, Pokemon def) {
        return true;
    }

    @Override
```

```

        protected void applySelfEffects(Pokemon self) {
            self.setMod(Stat.DEFENSE, 1);
        }
    }

// ru/ifmo/se/s267880/lab2/moves/DreamEater.java
package ru.ifmo.se.s267880.lab2.moves;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Status;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.Type;

public class DreamEater extends SpecialMove {
    public DreamEater() {
        super(Type.PSYCHIC, 100, 1);
    }

    @Override
    protected String describe() {
        return "Dream Eater: Dream Eater only works if the target is asleep. Dream Eater inflicts damage and 50% of the damage dealt (rounded up) is restored to the user as HP";
    }

    @Override
    protected double calcBaseDamage(Pokemon att, Pokemon def) {
        if (def.getCondition() == Status.SLEEP) {
            return super.calcBaseDamage(att, def);
        }
        return 0;
    }

    @Override
    protected void applySelfDamage(Pokemon self, double dam) {
        self.setMod(Stat.HP, (int)Math.ceil(dam / 2));
    }
}

// ru/ifmo/se/s267880/lab2/moves/EarthPower.java
package ru.ifmo.se.s267880.lab2.moves;

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

public class EarthPower extends SpecialMove {
    public EarthPower() {
        super(Type.GROUND, 90, 1);
    }

    @Override
    protected String describe() {
        return "Earth Power: Has a 10% chance to lower the target's Special Defense by one stage";
    }
}

```

```

    @Override
    protected void applyOppEffects(Pokemon opp) {
        if (Math.random() < 0.1) opp.setMod(Stat.SPECIAL_DEFENSE, -1);
    }
}

// ru/ifmo/se/s267880/lab2/moves/Facade.java
package ru.ifmo.se.s267880.lab2.moves;
import ru.ifmo.se.pokemon.PhysicalMove;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.Status;
import ru.ifmo.se.pokemon.Type;

public class Facade extends PhysicalMove {
    public Facade() {
        super(Type.NORMAL, 70, 1);
    }

    @Override
    protected String describe() {
        return "Facade: Power doubles if user is burned, paralyzed, or
poisoned";
    }

    @Override
    protected double calcBaseDamage(Pokemon att, Pokemon def) {
        double dam = super.calcBaseDamage(att, def);
        Status defSt = def.getCondition();
        if (defSt == Status.BURN || defSt == Status.PARALYZE || defSt ==
Status.POISON) {
            dam *= 2;
        }
        return dam;
    }
}

// ru/ifmo/se/s267880/lab2/moves/FireBlast.java
package ru.ifmo.se.s267880.lab2.moves;

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() {
        super(Type.FIRE, 110, 0.85);
    }

    @Override
    protected String describe() {
        return "Fire Blast: Has a 10% chance to burn the target";
    }

    @Override
    protected void applyOppEffects(Pokemon opp) {
        if (Math.random() < 0.1) Effect.burn(opp);
    }
}

```

```

}

// ru/ifmo/se/s267880/lab2/moves/Flamethrower.java
package ru.ifmo.se.s267880.lab2.moves;

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

public class Flamethrower extends SpecialMove {
    public Flamethrower() {
        super(Type.FIRE, 90, 1);
    }

    @Override
    protected String describe() {
        return "Flamethrower: Has a 10% chance to burn the target";
    }

    @Override
    protected void applyOppEffects(Pokemon opp) {
        if (Math.random() < 0.1) Effect.burn(opp);
    }
}

// ru/ifmo/se/s267880/lab2/moves/FocusBlast.java
package ru.ifmo.se.s267880.lab2.moves;

import ru.ifmo.se.pokemon.Effect;
import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Stat;
import ru.ifmo.se.pokemon.Type;

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

    @Override
    protected String describe() {
        return "Focus Blast: Has a 10% chance to lower the target's Special Defense by one stage";
    }

    @Override
    protected void applyOppEffects(Pokemon opp) {
        if (Math.random() < 0.1) opp.setMod(Stat.SPECIAL_DEFENSE, -1);
    }
}

// ru/ifmo/se/s267880/lab2/moves/NightShade.java
package ru.ifmo.se.s267880.lab2.moves;

import ru.ifmo.se.pokemon.SpecialMove;
import ru.ifmo.se.pokemon.Type;
import ru.ifmo.se.pokemon.Pokemon;

```



```

public class NightShade extends SpecialMove {
    public NightShade() {
        super(Type.GHOST, -1, 1);
    }

    @Override
    protected String describe() {
        return "Night Shade: Inflicts damage equal to the user's level";
    }

    @Override
    protected double calcBaseDamage(Pokemon atk, Pokemon def) {
        return atk.getLevel();
    }
}

// ru/ifmo/se/s267880/lab2/moves/Psychic.java
package ru.ifmo.se.s267880.lab2.moves;

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

public class Psychic extends SpecialMove {
    public Psychic() {
        super(Type.PSYCHIC, 90, 1);
    }

    @Override
    protected String describe() {
        return "Psychic: Has a 10% chance to lower the target's Special Defense by one stage";
    }

    @Override
    protected void applyOppEffects(Pokemon opp) {
        if (Math.random() < 0.1) opp.setMod(Stat.SPECIAL_DEFENSE, -1);
    }
}

// ru/ifmo/se/s267880/lab2/moves/RockTomb.java
package ru.ifmo.se.s267880.lab2.moves;

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

public class RockTomb extends PhysicalMove {
    public RockTomb() {
        super(Type.ROCK, 60, 0.95);
    }

    @Override
    protected String describe() {
        return "Rock Tomb: Has a 100% chance to lower the target's Speed by one stage";
    }
}

```

```

    @Override
    protected void applyOppEffects(Pokemon opp) {
        opp.setMod(Stat.SPEED, -1);
    }
}

// ru/ifmo/se/s267880/lab2/moves/Scratch.java
package ru.ifmo.se.s267880.lab2.moves;

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

public class Scratch extends PhysicalMove {
    public Scratch() {
        super(Type.NORMAL, 40, 1);
    }

    @Override
    protected String describe() {
        return "Scratch: Inflicts regular damage with no additional effect";
    }
}

// ru/ifmo/se/s267880/lab2/pokemons/Chandelure.java
package ru.ifmo.se.s267880.lab2.pokemons;

import ru.ifmo.se.s267880.lab2.pokemons.Lampent;
import ru.ifmo.se.s267880.lab2.moves.Psychic;

public class Chandelure extends Lampent {
    public Chandelure(String name) {
        this(name, 13);
    }

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

        setStats(60, 55, 90, 145, 90, 80);

        // move learnt by using machine
        addMove(new Psychic());
    }
}

// ru/ifmo/se/s267880/lab2/pokemons/Furret.java
package ru.ifmo.se.s267880.lab2.pokemons;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.s267880.lab2.pokemons.Sentret;
import ru.ifmo.se.s267880.lab2.moves.Coil;

public class Furret extends Sentret {
    public Furret(String name) {
        this(name, 4);
    }

    public Furret(String name, int level) {
        super(name, level);
    }
}

```

```

        setStats(85, 76, 64, 45, 55, 90);

        // no level required
        addMove(new Coil());
    }
}

// ru/ifmo/se/s267880/lab2/pokemons/Lampent.java
package ru.ifmo.se.s267880.lab2.pokemons;

import ru.ifmo.se.s267880.lab2.pokemons.Litwick;
import ru.ifmo.se.s267880.lab2.moves.NightShade;

public class Lampent extends Litwick {
    public Lampent(String name) {
        this(name, 13);
    }

    public Lampent(String name, int level) {
        super(name, level);
        setStats(60, 40, 60, 95, 60, 55);
        if (level >= 13) addMove(new NightShade());
    }
}

// ru/ifmo/se/s267880/lab2/pokemons/Landorus.java
package ru.ifmo.se.s267880.lab2.pokemons;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.s267880.lab2.moves.Facade;
import ru.ifmo.se.s267880.lab2.moves.FocusBlast;
import ru.ifmo.se.s267880.lab2.moves.RockTomb;
import ru.ifmo.se.s267880.lab2.moves.EarthPower;

public class Landorus extends Pokemon {
    public Landorus(String name) {
        this(name, 43); // required level
    }

    public Landorus(String name, int level) {
        super(name, level);
        setStats(89, 125, 90, 115, 80, 101);

        // Moves learnt by machine
        addMove(new Facade());
        addMove(new FocusBlast());

        // this move can be learnt using machine or pokemon is at the level
        addMove(new RockTomb());

        if (level >= 43) {
            // actually this move can be taught by an NPC anytime :v
            addMove(new EarthPower());
        }
    }
}

```

```
// ru/ifmo/se/s267880/lab2/pokemons/Litwick.java
package ru.ifmo.se.s267880.lab2.pokemons;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.s267880.lab2.moves.DreamEater;
import ru.ifmo.se.s267880.lab2.moves.FireBlast;

public class Litwick extends Pokemon {
    public Litwick(String name) {
        this(name, 1); // required level
    }

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

        setStats(50, 30, 55, 65, 55, 20);

        // moves learnt by using machines
        addMove(new DreamEater());
        addMove(new FireBlast());
    }
}
```

```
// ru/ifmo/se/s267880/lab2/pokemons/Sentret.java
package ru.ifmo.se.s267880.lab2.pokemons;

import ru.ifmo.se.pokemon.Pokemon;
import ru.ifmo.se.s267880.lab2.moves.Flamethrower;
import ru.ifmo.se.s267880.lab2.moves.DefenseCurl;
import ru.ifmo.se.s267880.lab2.moves.Scratch;

public class Sentret extends Pokemon {
    public Sentret(String name) {
        this(name, 4);
    }

    public Sentret(String name, int level) {
        super(name, level);
        setStats(35, 46, 34, 35, 45, 20);

        // moves learnt by using machine
        addMove(new Flamethrower());

        // no level require
        addMove(new Scratch());

        if (level >= 4) addMove(new DefenseCurl());
    }
}
```

```
// ru/ifmo/se/s267880/lab2/IWannaBeTheVeryBest.java
package ru.ifmo.se.s267880.lab2;

import ru.ifmo.se.pokemon.Battle;
import ru.ifmo.se.s267880.lab2.pokemons.*;

public class IWannaBeTheVeryBest {
```

```

public static void main(String[] args) {
    String names[] = {
        "Pankratiy Andrei",
        "Maryana Uliana",
        "Dorofei Anisim",
        "Sveta Igor",
        "Diana Gavriil",
        "Yuliy Vitaliya",
    };
    Battle bt = new Battle();
    bt.addAlly(new Sentret(names[2]));
    bt.addAlly(new Lampent(names[0]));
    bt.addAlly(new Landorus(names[5]));
    bt.addFoe(new Furret(names[3]));
    bt.addFoe(new Litwick(names[4]));
    bt.addFoe(new Chandelure(names[1]));
    bt.go();
}
}

```

Результат работы программы.

Sentret Dorofei Anisim from the team Greren enters the battle!

Furret Sveta Igor from the team Purple enters the battle!

Furret Sveta Igor Scratch: Inflicts regular damage with no additional effect.

Sentret Dorofei Anisim loses 6 hit points.

Sentret Dorofei Anisim Flamethrower: Has a 10% chance to burn the target.

Furret Sveta Igor loses 6 hit points.

Furret Sveta Igor Coil: Raises the user's Attack, Defense, and accuracy by one stage each.

Furret Sveta Igor increases attack.

Furret Sveta Igor increases defense.

Furret Sveta Igor increases accuracy.

Sentret Dorofei Anisim Scratch: Inflicts regular damage with no additional effect.

Furret Sveta Igor loses 5 hit points.

Furret Sveta Igor Flamethrower: Has a 10% chance to burn the target.

Sentret Dorofei Anisim loses 6 hit points.

Sentret Dorofei Anisim DefenseCurl: Raises user's Defense by one stage.

Sentret Dorofei Anisim increases defense.

Furret Sveta Igor Coil: Raises the user's Attack, Defense, and accuracy by one stage each.

Furret Sveta Igor increases attack.

Furret Sveta Igor increases defense.

Furret Sveta Igor increases accuracy.

Sentret Dorofei Anisim Flamethrower: Has a 10% chance to burn the target.

Furret Sveta Igor loses 7 hit points.

Furret Sveta Igor Scratch: Inflicts regular damage with no additional effect.

Sentret Dorofei Anisim loses 4 hit points.

Sentret Dorofei Anisim Flamethrower: Has a 10% chance to burn the target.

Furret Sveta Igor loses 4 hit points.

Furret Sveta Igor faints.

Litwick Diana Gavriil from the team Purple enters the battle!

Sentret Dorofei Anisim DefenseCurl: Raises user's Defense by one stage.

Sentret Dorofei Anisim increases defense.

Litwick Diana Gavriil struggles.
Sentret Dorofei Anisim loses 3 hit points.
Litwick Diana Gavriil loses 1 hit points.
Sentret Dorofei Anisim faints.
Lampent Pankratiy Andrei from the team Greren enters the battle!
Lampent Pankratiy Andrei struggles.
Litwick Diana Gavriil loses 12 hit points.
Lampent Pankratiy Andrei loses 3 hit points.
Litwick Diana Gavriil faints.
Chandelure Maryana Uliana from the team Purple enters the battle!
Chandelure Maryana Uliana Psychic: Has a 10% chance to lower the target's Special Defense by one stage.
Lampent Pankratiy Andrei loses 15 hit points.

Lampent Pankratiy Andrei struggles.
Chandelure Maryana Uliana loses 6 hit points.
Lampent Pankratiy Andrei loses 2 hit points.

Chandelure Maryana Uliana Dream Eater: Dream Eater only works if the target is asleep.
Dream Eater inflicts damage and 50% of the damage dealt (rounded up) is restored to the user as HP.
Lampent Pankratiy Andrei loses 3 hit points.
Chandelure Maryana Uliana loses 2 hit points.

Lampent Pankratiy Andrei struggles.
Chandelure Maryana Uliana loses 5 hit points.
Lampent Pankratiy Andrei loses 1 hit points.

Chandelure Maryana Uliana Night Shade: Inflicts damage equal to the user's level.
Critical hit!
Lampent Pankratiy Andrei loses 86 hit points.
Lampent Pankratiy Andrei faints.
Landorus Yuliy Vitaliya from the team Greren enters the battle!
Landorus Yuliy Vitaliya Rock Tomb: Has a 100% chance to lower the target's Speed by one stage.
Critical hit!
Chandelure Maryana Uliana loses 106 hit points.
Chandelure Maryana Uliana decreases speed.
Chandelure Maryana Uliana faints.
Team Purple loses its last Pokemon.
The team Greren wins the battle!

Выводы: в процессе выполнения лабораторной работы были получены навыки использования объектно-ориентированного подхода программирования.