

Министерство образования и науки Российской Федерации
САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ,
МЕХАНИКИ И ОПТИКИ

Факультет программной инженерии и компьютерной техники
Дисциплина: «Программирование»

ОТЧЁТ
по лабораторной работе №2

Вариант №34003

Выполнила:
Студентка группы Р3121
Егорова Варвара
Александровна
Преподаватель:
Данилов Павел Юрьевич

Оглавление

Текст задания.....	3
Покемоны и их атаки варианта:	4
Исходный код:.....	5
Main.....	5
Pokemons.....	6
Groudon.....	6
LiLeep	6
Cradily.....	7
Nidoran-M	7
Nidorino.....	7
Nidoking.....	8
Moves	9
Bulldoze	9
Dragon Claw.....	9
Focus Blast	10
Lava Plume.....	10
Double_Team.....	11
Giga Drain	11
Rest.....	12
Rock Polish.....	12
Thunderbolt	13
Ice Beam	14
Flatter	15
UML	16
Пример выходных данных программы:	17
Вывод.....	24

Текст 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 <http://poke-universe.ru>, <http://pokemondb.net>, <http://veekun.com/dex/pokemon>

Комментарии

Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Что надо сделать (краткое описание)

1. Ознакомиться с [документацией](#), обращая особое внимание на классы **Pokemon** и **Move**. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
2. Скачать файл **Pokemon.jar**. Его необходимо будет использовать как для компиляции, так и для запуска программы. Распаковывать его не надо! Нужно научиться подключать внешние jar-файлы к своей программе.
3. Написать минимально работающую программу и посмотреть как она работает.

```
Battle b = new Battle();
Pokemon p1 = new Pokemon("Чужой", 1);
Pokemon p2 = new Pokemon("Хищник", 1);
b.addAlly(p1);
```

```
b.addFoe(p2);  
b.go();
```

4. Создать один из классов покемонов для своего варианта. Класс должен наследоваться от базового класса **Pokemon**. В конструкторе нужно будет задать типы покемона и его базовые характеристики. После этого попробуйте добавить покемона в сражение.
5. Создать один из классов атак для своего варианта (лучше всего начать с физической или специальной атаки). Класс должен наследоваться от класса **PhysicalMove** или **SpecialMove**. В конструкторе нужно будет задать тип атаки, ее силу и точность. После этого добавить атаку покемону и проверить ее действие в сражении. Не забудьте переопределить метод **describe**, чтобы выводилось нужное сообщение.
6. Если действие атаки отличается от стандартного, например, покемон не промахивается, либо атакующий покемон также получает повреждение, то в классе атаки нужно дополнительно переопределить соответствующие методы (см. документацию). При реализации атак, которые меняют статус покемона (наследники **StatusMove**), скорее всего придется разобраться с классом **Effect**. Он позволяет на один или несколько ходов изменить состояние покемона или модификатор его базовых характеристик.
7. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

Покемоны и их атаки варианта:

Groudon  Атаки: <ul style="list-style-type: none">✓ Lava Plume✓ Focus Blast✓ Thunder✓ Dragon Claw	Lileep  Атаки: <ul style="list-style-type: none">✓ Giga Drain✓ Rock Polish✓ Rest	Cradily  Атаки: <ul style="list-style-type: none">✓ Giga Drain✓ Rock Polish✓ Rest✓ Bulldoze	Nidoran-M  Атаки: <ul style="list-style-type: none">✓ Rest✓ Ice Beam	Nidorino  Атаки: <ul style="list-style-type: none">✓ Rest✓ Ice Beam✓ Flatter	Nidoking  Атаки: <ul style="list-style-type: none">✓ Rest✓ Ice Beam✓ Flatter✓ Thunderbolt
---	---	---	---	---	---

Исходный код:

Main

```
package ru.itmo.lab2;

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

public class Main{
    public static void main(String[] args){
        Battle b = new Battle();
        Cradily cradily = new Cradily("Андрюха", 50);
        Groudon groudon = new Groudon("Работяга", 50);
        Lileep lileep = new Lileep("Никита", 50);
        NidoranM nidoranM = new NidoranM("Лешка", 50);
        Nidorino nidorino = new Nidorino("Леха", 50);
        Nidoking nidoking = new Nidoking("Алексей", 50);

        b.addAlly(nidorino);
        b.addAlly(lileep);
        b.addAlly(groudon);
        b.addFoe(nidoranM);
        b.addFoe(cradily);
        b.addFoe(nidoking);
        b.go();
    }
}
```

Pokemons

Groudon

```
package ru.itmo.lab2.pokemons;

import ru.itmo.lab2.moves.Groudon.*;
import ru.ifmo.se.pokemon.*;

public class Groudon extends Pokemon {
    public Groudon(String name, int level) {
        super(name, level);
        super.setType(Type.GROUND);
        super.setStats(100, 150, 140, 100, 90, 90);
        super.setMove(new LavaPlume(80, 100), new
FocusBlast(120, 70), new Thunder(110, 70), new DragonClaw(80,
100));
    }
}
```

Lileep

```
package ru.itmo.lab2.pokemons;

import ru.itmo.lab2.moves.Lileep.*;
import ru.ifmo.se.pokemon.*;

public class Lileep extends Pokemon {
    public Lileep(String name, int level) {
        super(name, level);
        super.setType(Type.ROCK, Type.GRASS);
        super.setStats(66, 41, 77, 61, 87, 23);
        super.setMove(new GigaDrain(75, 100), new RockPolish(0,
100), new Rest(0, 100));
    }
}
```

Cradily

```
package ru.itmo.lab2.pokemons;

import ru.itmo.lab2.moves.Lileep.*;
import ru.itmo.lab2.moves.Cradily.*;
import ru.ifmo.se.pokemon.*;

public class Cradily extends Lileep {
    public Cradily(String name, int level) {
        super(name, level);
        super.setType(Type.ROCK, Type.GRASS);
        super.setStats(86, 81, 97, 81, 107, 43);
    }
}
```

```

        super.setMove(new GigaDrain(75, 100), new RockPolish(0,
100), new Rest(0, 100), new Bulldoze(60, 100));
    }
}

```

Nidoran-M

```

package ru.itmo.lab2.pokemons;

import ru.itmo.lab2.moves.NidoranM.*;
import ru.ifmo.se.pokemon.*;

public class NidoranM extends Pokemon {
    public NidoranM(String name, int level) {
        super(name, level);
        super.setType(Type.POISON);
        super.setStats(46, 57, 40, 40, 40, 50);
        super.setMove(new Rest(0, 100), new IceBeam(90, 100));
    }
}

```

Nidorino

```

package ru.itmo.lab2.pokemons;

import ru.itmo.lab2.moves.NidoranM.*;
import ru.itmo.lab2.moves.Nidorino.*;
import ru.ifmo.se.pokemon.*;

public class Nidorino extends NidoranM {
    public Nidorino(String name, int level) {
        super(name, level);
        super.setType(Type.POISON);
        super.setStats(61, 72, 57, 55, 55, 65);
        super.setMove(new Rest(0, 100), new IceBeam(90, 100),
new Flatter(0, 100));
    }
}

```

Nidoking

```

package ru.itmo.lab2.pokemons;

import ru.itmo.lab2.moves.NidoranM.*;
import ru.itmo.lab2.moves.Nidorino.*;
import ru.itmo.lab2.moves.Nidoking.*;
import ru.ifmo.se.pokemon.*;

public class Nidoking extends Nidorino {
    public Nidoking(String name, int level) {
        super(name, level);
        super.setType(Type.POISON, Type.GROUND);
        super.setStats(81, 102, 77, 85, 75, 85);
        super.setMove(new IceBeam(90, 100), new Flatter(0, 100),
new Thunderbolt(90, 100), new Rest(0, 100));
    }
}

```

```
}  
}
```


Moves

Bulldoze

```
package ru.itmo.lab2.moves.Cradily;

import ru.ifmo.se.pokemon.*;

public class Bulldoze extends PhysicalMove {
    public Bulldoze(double pow, double acc) {
        super(Type.GROUND, pow, acc);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        Effect EffectObject = new Effect().stat(Stat.SPEED, -1);
        p.addEffect(EffectObject);
    }

    @Override
    protected String describe() {
        return "бьет по всему вокруг, утрамбовывая землю";
    }
}
```

Dragon Claw

```
package ru.itmo.lab2.moves.Groudon;

import ru.ifmo.se.pokemon.*;

public class DragonClaw extends PhysicalMove{
    public DragonClaw(double pow, double acc){
        super(Type.DRAGON, pow, acc);
    }
    @Override
    protected String describe(){
        return "наносит удар мощными драконьими когтями";
    }
}
```

Focus Blast

```
package ru.itmo.lab2.moves.Groudon;

import ru.ifmo.se.pokemon.*;

public class FocusBlast extends SpecialMove {
    public FocusBlast(double pow, double acc) {
        super(Type.FIGHTING, pow, acc);
    }
}
```

```

    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        Effect EffectObject = new Effect().chance(0.1);
        EffectObject.stat(Stat.SPECIAL_DEFENSE, -1);
    }

    @Override
    protected String describe() {
        return "фокусирует боевую энергию и запускает шар";
    }
}

```

Lava Plume

```

package ru.itmo.lab2.moves.Groudon;

import ru.ifmo.se.pokemon.*;

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

    @Override
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        Effect e = new Effect().chance(0.3);
        e.burn(p);
    }

    @Override
    protected String describe() {
        return "выплескивает поток лавы";
    }
}

```

Thunder

```

package ru.itmo.lab2.moves.Groudon;

import ru.ifmo.se.pokemon.*;

public class Thunder extends SpecialMove {
    public Thunder(double pow, double acc) {
        super(Type.ELECTRIC, pow, acc);
    }
}

```

```

@Override
protected void applyOppEffects(Pokemon p) {
    super.applyOppEffects(p);
    Effect e = new Effect().chance(0.3);
    e.paralyze(p);
}

@Override
protected String describe() {
    return "атакует мощнейшим ударом молнии";
}
}

```

Giga Drain

```

package ru.itmo.lab2.moves.Lileep;

import ru.ifmo.se.pokemon.*;

public class GigaDrain extends SpecialMove{
    public GigaDrain(double pow, double acc){
        super(Type.GRASS, pow, acc);
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        super.applySelfEffects(p);
        Effect EffectObject = new Effect().stat(Stat.HP, -2);
        p.addEffect(EffectObject);
    }
    @Override
    protected String describe(){
        return "высосал питательные вещества";
    }
}

```

Rest

```

package ru.itmo.lab2.moves.Lileep;

import ru.ifmo.se.pokemon.*;

public class Rest extends StatusMove{
    public Rest(double pow, double acc){
        super(Type.PSYCHIC, pow, acc);
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        super.applySelfEffects(p);
        Effect.sleep(p);
        Effect EffectObject = new

```

```

Effect().turns(1).stat(Stat.HP, -30);
    p.addEffect(EffectObject);
}
@Override
protected String describe(){
    return "отдыхает";
}
}

```

Rock Polish

```

package ru.itmo.lab2.moves.Lileep;

import ru.ifmo.se.pokemon.*;

public class RockPolish extends StatusMove{
    public RockPolish(double pow, double acc){
        super(Type.ROCK, pow, acc);
    }
    @Override
    protected void applySelfEffects(Pokemon p){
        super.applySelfEffects(p);
        Effect EffectObject = new
Effect().stat(Stat.SPEED, 2);
        p.addEffect(EffectObject);
    }
    @Override
    protected String describe(){
        return "полирует своё тело";
    }
}

```

Thunderbolt

```

package ru.itmo.lab2.moves.Nidoking;

import ru.ifmo.se.pokemon.*;

public class Thunderbolt extends SpecialMove {
    public Thunderbolt(double pow, double acc) {
        super(Type.ELECTRIC, pow, acc);
    }

    @Override
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        Effect e = new Effect().chance(0.1);
        e.paralyze(p);
    }
}

```

```

@Override
protected String describe() {
    return "атакует точным ударом молнии";
}
}

```

Ice Beam

```

package ru.itmo.lab2.moves.NidoranM;

import ru.ifmo.se.pokemon.*;

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

    @Override
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        Effect e = new Effect().chance(0.1);
        e.freeze(p);
    }

    @Override
    protected String describe() {
        return "выпускает леденящий луч";
    }
}

```

Flatter

```

package ru.itmo.lab2.moves.Nidorino;

import ru.ifmo.se.pokemon.*;

public class Flatter extends SpecialMove {
    public Flatter(double pow, double acc) {
        super(Type.DARK, pow, acc);
    }

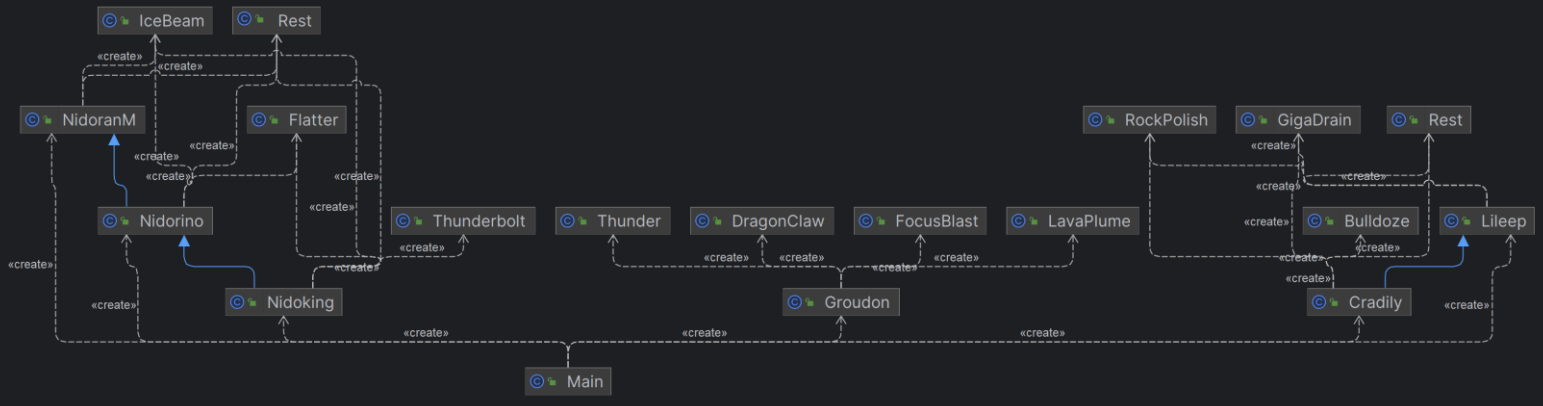
    @Override
    protected void applyOppEffects(Pokemon p) {
        super.applyOppEffects(p);
        Effect EffectObject = new
Effect().stat(Stat.SPECIAL_ATTACK, 1);
        p.addEffect(EffectObject);
    }
}

```

```
        Effect.confuse(p);
    }

    @Override
    protected String describe() {
        return "говорит лестные слова";
    }
}
```

UML



Пример выходных данных программы:

Nidorino Леха из команды черных вступает в бой!

NidoranM Лешка из команды синих вступает в бой!

Nidorino Леха выпускает леденящий луч.

NidoranM Лешка теряет 30 здоровья.

NidoranM Лешка отдыхает.

NidoranM Лешка засыпает

NidoranM Лешка восстанавливает 28 здоровья.

Nidorino Леха говорит лестные слова.

NidoranM Лешка теряет 2 здоровья.

Nidorino Леха отдыхает.

Nidorino Леха засыпает

Nidorino Леха восстанавливает 28 здоровья.

NidoranM Лешка выпускает леденящий луч.

Nidorino Леха теряет 17 здоровья.

Nidorino Леха выпускает леденящий луч.

NidoranM Лешка теряет 29 здоровья.

NidoranM Лешка растерянно попадает по себе.

NidoranM Лешка теряет 15 здоровья.

Nidorino Леха выпускает леденящий луч.

NidoranM Лешка теряет 23 здоровья.

NidoranM Лешка выпускает леденящий луч.

Nidorino Леха теряет 22 здоровья.

Nidorino Леха говорит лестные слова.

NidoranM Лешка теряет 3 здоровья.

NidoranM Лешка отдыхает.

NidoranM Лешка засыпает

NidoranM Лешка восстанавливает 28 здоровья.

Nidorino Леха говорит лестные слова.

Критический удар!

NidoranM Лешка теряет 5 здоровья.

Nidorino Леха отдыхает.

Nidorino Леха засыпает

NidoranM Лешка выпускает леденящий луч.

Nidorino Леха теряет 45 здоровья.

Nidorino Леха восстанавливает 28 здоровья.

Nidorino Леха говорит лестные слова.

NidoranM Лешка теряет 3 здоровья.

NidoranM Лешка отдыхает.

NidoranM Лешка засыпает

NidoranM Лешка восстанавливает 28 здоровья.

Nidorino Леха выпускает леденящий луч.

NidoranM Лешка теряет 20 здоровья.

Nidorino Леха выпускает леденящий луч.

NidoranM Лешка теряет 36 здоровья.

NidoranM Лешка отдыхает.

NidoranM Лешка засыпает

NidoranM Лешка восстанавливает 28 здоровья.

Nidorino Леха отдыхает.

Nidorino Леха засыпает

Nidorino Леха восстанавливает 28 здоровья.

NidoranM Лешка отдыхает.

NidoranM Лешка засыпает

NidoranM Лешка восстанавливает 28 здоровья.

Nidorino Леха выпускает леденящий луч.

NidoranM Лешка теряет 21 здоровья.

Nidorino Леха выпускает леденящий луч.

Критический удар!

NidoranM Лешка теряет 70 здоровья.

NidoranM Лешка теряет сознание.

Cradily Андрюха из команды синих вступает в бой!

Nidorino Леха отдыхает.

Nidorino Леха засыпает

Cradily Андрюха полирует своё тело.

Nidorino Леха восстанавливает 28 здоровья.

Cradily Андрюха бьет по всему вокруг, утрамбовывая землю.

Nidorino Леха теряет 45 здоровья.

Nidorino Леха говорит лестные слова.

Критический удар!

Cradily Андрюха теряет 7 здоровья.

Nidorino Леха говорит лестные слова.

Cradily Андрюха теряет 3 здоровья.

Cradily Андрюха растерянно попадает по себе.

Cradily Андрюха теряет 18 здоровья.

Nidorino Леха выпускает леденящий луч.

Cradily Андрюха теряет 34 здоровья.

Cradily Андрюха отдыхает.

Cradily Андрюха засыпает

Cradily Андрюха восстанавливает 30 здоровья.

Nidorino Леха выпускает леденящий луч.

Cradily Андрюха теряет 32 здоровья.

Nidorino Леха отдыхает.

Nidorino Леха засыпает

Cradily Андрюха полирует своё тело.

Nidorino Леха восстанавливает 28 здоровья.

Cradily Андрюха растерянно попадает по себе.

Cradily Андрюха теряет 16 здоровья.

Nidorino Леха говорит лестные слова.

Критический удар!

Cradily Андрюха теряет 5 здоровья.

Cradily Андрюха отдыхает.

Cradily Андрюха засыпает

Cradily Андрюха восстанавливает 30 здоровья.

Nidorino Леха отдыхает.

Nidorino Леха засыпает

Nidorino Леха восстанавливает 28 здоровья.

Nidorino Леха выпускает леденящий луч.

Критический удар!

Cradily Андрюха теряет 56 здоровья.

Cradily Андрюха бьет по всему вокруг, утрамбовывая землю.

Nidorino Леха теряет 35 здоровья.

Nidorino Леха выпускает леденящий луч.

Критический удар!

Cradily Андрюха теряет 52 здоровья.

Cradily Андрюха теряет сознание.

Nidoking Алексей из команды синих вступает в бой!

Nidorino Леха отдыхает.

Nidorino Леха засыпает

Nidoking Алексей атакует точным ударом молнии.

Критический удар!

Nidorino Леха теряет 53 здоровья.

Nidorino Леха восстанавливает 28 здоровья.

Nidoking Алексей выпускает леденящий луч.

Nidorino Леха теряет 24 здоровья.

Nidorino Леха замерзает

Nidorino Леха оттаивает

Nidorino Леха отдыхает.

Nidorino Леха засыпает

Nidoking Алексей выпускает леденящий луч.

Nidorino Леха теряет 39 здоровья.

Nidorino Леха восстанавливает 28 здоровья.

Nidorino Леха говорит лестные слова.

Критический удар!

Nidoking Алексей теряет 5 здоровья.

Nidoking Алексей говорит лестные слова.

Nidorino Леха теряет 3 здоровья.

Nidorino Леха выпускает леденящий луч.

Критический удар!

Nidoking Алексей теряет 75 здоровья.

Nidoking Алексей говорит лестные слова.

Nidorino Леха теряет 3 здоровья.

Nidorino Леха выпускает леденящий луч.

Критический удар!

Nidoking Алексей теряет 117 здоровья.

Nidoking Алексей теряет сознание.

В команде синих не осталось покемонов.

Команда черных побеждает в этом бою!

Вывод

Для выполнения этой лабораторной работы нужно было ознакомиться с основными принципами ООП, понятиями объектов, классов, наследования, инкапсуляции, научиться переопределять методы.