Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

**Лабораторная работа по программированию №2**

вариант: 839

Преподаватель: Харитонова Анастасия Евгеньевна

Выполнил: Дагаев Даниил Сергеевич

Группа: Р3108

Санкт-Петербург, 2021г

Оглавление

[Задание 2](#_Toc84076276)

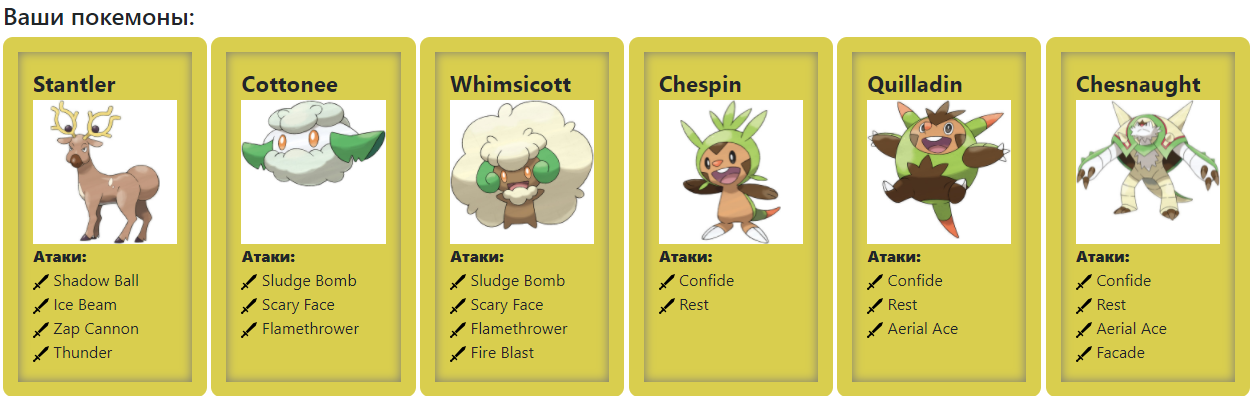
[Исходный код программы 2](#_Toc84076277)

[Результат работы программы 3](#_Toc84076278)

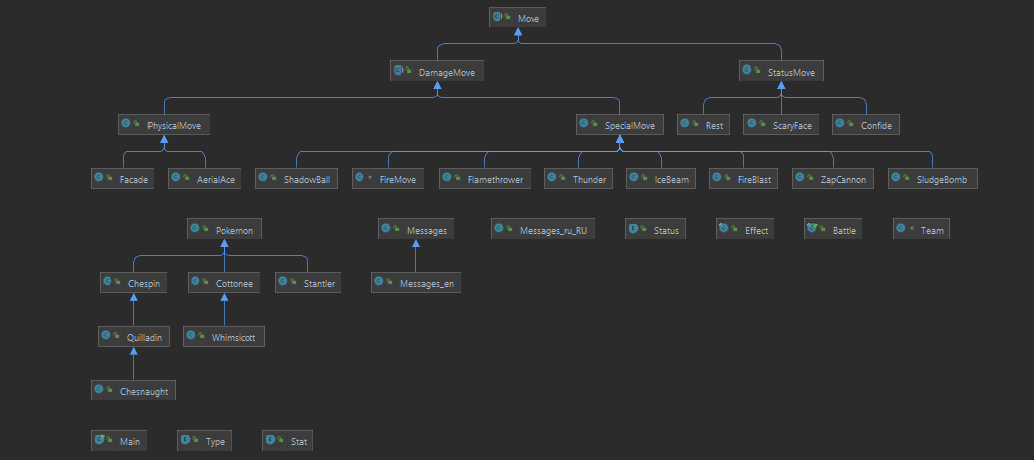
[Вывод 3](#_Toc84076279)

# Задание

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.
4. Battle b = new Battle();
5. Pokemon p1 = new Pokemon("Чужой", 1);
6. Pokemon p2 = new Pokemon("Хищник", 1);
7. b.addAlly(p1);
8. b.addFoe(p2);
9. b.go();
10. 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.
11. 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.
12. 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.
13. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.



# Диаграмма



# Исходный код программы

import info.moves.Confide;  
import info.moves.Rest;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Chespin extends Pokemon {  
 public Chespin(String name,int level){  
 super(name,level);  
 setStats(56,61,65,48,45,38);  
 setMove(new Confide(),new Rest());  
 setType(Type.*GRASS*);  
 }  
}

import info.moves.AerialAce;  
import info.moves.Confide;  
import info.moves.Rest;  
import ru.ifmo.se.pokemon.Effect;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Quilladin extends Chespin {  
 public Quilladin(String name,int level){  
 super(name,level);  
 setStats(61,78,95,56,58,57);  
 setMove(new Confide(),new Rest(),new AerialAce());  
 setType(Type.*GRASS*);  
 }  
}

import info.moves.AerialAce;  
import info.moves.Confide;  
import info.moves.Facade;  
import info.moves.Rest;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Chesnaught extends Quilladin {  
 public Chesnaught(String name, int level){  
 super(name, level);  
 setStats(88,107,122,74,75,64);  
 setMove(new Confide(),new Rest(),new AerialAce(),new Facade());  
 setType(Type.*GRASS*,Type.*FIGHTING*);  
 }  
}

import info.moves.Flamethrower;  
import info.moves.ScaryFace;  
import info.moves.SludgeBomb;  
import ru.ifmo.se.pokemon.Pokemon;  
import ru.ifmo.se.pokemon.Type;  
  
public class Cottonee extends Pokemon {  
 public Cottonee(String name,int level){  
 super(name,level);  
 setStats(40,27,60,37,50,66);  
 setMove(new SludgeBomb(), new ScaryFace(), new Flamethrower());  
 setType(Type.*GRASS*,Type.*FAIRY*);  
 }  
}

import info.moves.FireBlast;  
import info.moves.Flamethrower;  
import info.moves.ScaryFace;  
import info.moves.SludgeBomb;  
import ru.ifmo.se.pokemon.Type;  
  
public class Whimsicott extends Cottonee {  
 public Whimsicott(String name,int level){  
 super(name,level);  
 setStats(60,67,85,77,75,116);  
 setMove(new SludgeBomb(),new ScaryFace(),new Flamethrower(),new FireBlast());  
 setType(Type.*GRASS*,Type.*FAIRY*);  
 }  
}

import ru.ifmo.se.pokemon.Pokemon;  
import info.moves.\*;  
import ru.ifmo.se.pokemon.\*;  
  
public class Stantler extends Pokemon {  
 public Stantler(String name, int level){  
 super(name,level);  
 setStats(73, 95,62,85,65,85);  
 setMove(new ShadowBall(), new IceBeam(), new ZapCannon(),new Thunder());  
 setType(Type.*NORMAL*);  
 }  
}

package info.moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class AerialAce extends PhysicalMove {  
 public AerialAce(){  
 super(Type.*FLYING*,60,100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.setMod(Stat.*EVASION*,0);  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Aerial Ace";  
 }  
}

package info.moves;  
  
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 Confide extends StatusMove {  
 public Confide(){  
 super(Type.*NORMAL*,0,0);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 p.setMod(Stat.*SPECIAL\_ATTACK*,-1);  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Confide";  
 }  
}

package info.moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Facade extends PhysicalMove {  
 public Facade(){  
 super(Type.*NORMAL*,70,100);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon p, double damage) {  
 Status PokCon = p.getCondition();  
 if (PokCon.equals(Status.*BURN*)|| PokCon.equals(Status.*POISON*)|| PokCon.equals(Status.*PARALYZE*)){  
 super.applyOppDamage(p,damage \* 2);  
 }  
 else{  
 super.applyOppDamage(p,damage);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Facade";  
 }  
}

package info.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*,110,85); }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.*random*() <= 0.1) {  
 Effect.*burn*(p);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "Fire Blast";  
 }  
}

package info.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 Flamethrower extends SpecialMove {  
 public Flamethrower(){  
 super(Type.*FIRE*,90,100);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 Effect.*burn*(p);  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Flamethrower";  
 }  
}

package info.moves;  
  
import ru.ifmo.se.pokemon.SpecialMove;  
import ru.ifmo.se.pokemon.\*;  
public class IceBeam extends SpecialMove {  
 public IceBeam(){  
 super(Type.*ICE*,90,100);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p){  
 if (Math.*random*() <= 0.1){  
 Effect.*freeze*(p);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Ice Beam";  
 }  
}

package info.moves;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Rest extends StatusMove {  
 public Rest(){  
 super(Type.*PSYCHIC*,0,0);  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p) {  
 p.restore();  
 Effect eff = new Effect();  
 eff = eff.condition(Status.*SLEEP*);  
 eff = eff.turns(2);  
 p.addEffect(eff);  
 }  
  
 @Override  
 protected String describe() {  
 return "кастует Rest";  
 }  
}

package info.moves;  
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 ScaryFace extends StatusMove {  
 public ScaryFace(){  
 super(Type.*NORMAL*,0,100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 p.setMod(Stat.*SPEED*,-1);  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Scary Face";  
 }  
}

package info.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 ShadowBall extends SpecialMove {  
 public ShadowBall(){  
 super(Type.*GHOST*,80,100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.*random*() <= 0.20) {  
 p.setMod(Stat.*DEFENSE*, -1);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Shadow Ball";  
 }  
}

package info.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 SludgeBomb extends SpecialMove {  
 public SludgeBomb(){  
 super(Type.*POISON*,90,100);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.*random*() <= 0.3){  
 Effect.*poison*(p);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "Кастует Sludge Bomb";  
 }  
}

package info.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 Thunder extends SpecialMove {  
 public Thunder(){  
 super(Type.*ELECTRIC*,110,70);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 if (Math.*random*() <= 0.3){  
 Effect.*paralyze*(p);  
 }  
 }  
  
 @Override  
 protected String describe() {  
 return "кастует Thunder";  
 }  
}

package info.moves;  
import ru.ifmo.se.pokemon.\*;  
import java.security.spec.ECField;  
public class ZapCannon extends SpecialMove {  
 public ZapCannon() {  
 super(Type.*ELECTRIC*,120,50);  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p) {  
 Effect.*paralyze*(p);  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon p, double damage) {  
 if (Math.*random*() <= 0.25){  
 super.applyOppDamage(p, damage \* 0);  
 double s = p.getStat(Stat.*SPEED*);  
 p.setMod(Stat.*SPEED*, (int) Math.*round*(s \* 0.5));  
 }  
 else{  
 super.applyOppDamage(p,damage);  
 }  
 }  
 @Override  
 protected String describe() {  
 return "Кастует ZapCannon";  
 }  
}

import ru.ifmo.se.pokemon.\*;  
public class Main {  
 public static void main(String[] args){  
 Battle b = new Battle();  
 b.addAlly(new Chesnaught("Соляра", 1));  
 b.addAlly(new Chespin("Карасик",1));  
 b.addAlly(new Cottonee("FP AM",1));  
 b.addFoe(new Quilladin("Solo Mid",1));  
 b.addFoe(new Stantler("Дед Внутри", 1));  
 b.addFoe(new Whimsicott("ZXCursed",1));  
 b.go();  
 }  
}

# Результат работы программы

"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.2\lib\idea\_rt.jar=51041:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2021.1.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\imia\AppData\Local\JetBrains\IdeaIC2021.1\compile-server\icebeam\_java\_8fa47ce3\scratches\out;C:\Users\imia\IdeaProjects\Laba2\out\production\Laba2 Main

Chesnaught Соляра из команды зеленых вступает в бой!

Quilladin Solo Mid из команды синих вступает в бой!

Chesnaught Соляра промахивается

Quilladin Solo Mid промахивается

Chesnaught Соляра промахивается

Quilladin Solo Mid промахивается

Chesnaught Соляра промахивается

Quilladin Solo Mid Кастует Aerial Ace.

Chesnaught Соляра теряет 18 здоровья.

Chesnaught Соляра теряет сознание.

Chespin Карасик из команды зеленых вступает в бой!

Quilladin Solo Mid Кастует Aerial Ace.

Chespin Карасик теряет 7 здоровья.

Chespin Карасик промахивается

Quilladin Solo Mid Кастует Aerial Ace.

Chespin Карасик теряет 10 здоровья.

Chespin Карасик теряет сознание.

Cottonee FP AM из команды зеленых вступает в бой!

Cottonee FP AM Кастует Flamethrower.

Quilladin Solo Mid теряет 8 здоровья.

Quilladin Solo Mid воспламеняется

Quilladin Solo Mid Кастует Aerial Ace.

Cottonee FP AM восстанавливает 1 здоровья.

Cottonee FP AM Кастует Scary Face.

Quilladin Solo Mid уменьшает скорость.

Quilladin Solo Mid Кастует Aerial Ace.

Cottonee FP AM восстанавливает 1 здоровья.

Quilladin Solo Mid промахивается

Cottonee FP AM Кастует Flamethrower.

Quilladin Solo Mid теряет 10 здоровья.

Quilladin Solo Mid теряет сознание.

Stantler Дед Внутри из команды синих вступает в бой!

Stantler Дед Внутри Кастует ZapCannon.

Cottonee FP AM теряет 3 здоровья.

Cottonee FP AM парализован

Cottonee FP AM Кастует Scary Face.

Stantler Дед Внутри уменьшает скорость.

Cottonee FP AM Кастует Flamethrower.

Критический удар!

Stantler Дед Внутри теряет 11 здоровья.

Stantler Дед Внутри воспламеняется

Stantler Дед Внутри Кастует Shadow Ball.

Cottonee FP AM теряет 5 здоровья.

Stantler Дед Внутри кастует Thunder.

Cottonee FP AM теряет 3 здоровья.

Stantler Дед Внутри Кастует Ice Beam.

Cottonee FP AM теряет 8 здоровья.

Cottonee FP AM теряет сознание.

В команде зеленых не осталось покемонов.

Команда синих побеждает в этом бою!

Process finished with exit code 0

# Вывод

Во время выполнения лабораторной работы я познакомился с ООП, классами, объектами, наследуемостью, инкапсуляцией и т.д. То есть были получены навыки использования объектно-ориентированного программирования.