**Федеральное государственное автономное образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
  
Факультет программной инженерии и компьютерной техники  
  
Направление подготовки 09.03.04 «Программная инженерия» – Системное и прикладное программное обеспечение**

**Отчёт  
По лабораторной работе №2  
По дисциплине «Программирование»  
Вар: 2626**

**Выполнил:  
Студент 1 курса  
Васильев Артём Евгеньевич  
Группа: 3119  
Преподаватель:  
Бойко Владислав Алексеевич**

Санкт-Петербург, 2023 г.

# Оглавление:

[Задание: 3](#_Toc6297)

[Диаграмма классов реализованной объектной модели: 4](#_Toc15854)

[Исходный код программы: 4](#_Toc354)

[Ссылка на гитхаб с исходным кодом: 12](#_Toc31135)

[Результат работы программы: 13](#_Toc29262)

[Вывод: 16](#_Toc10365)

[Список источников: 17](#_Toc6121)

# 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.



Рис. 1

# Диаграмма классов реализованной объектной модели:

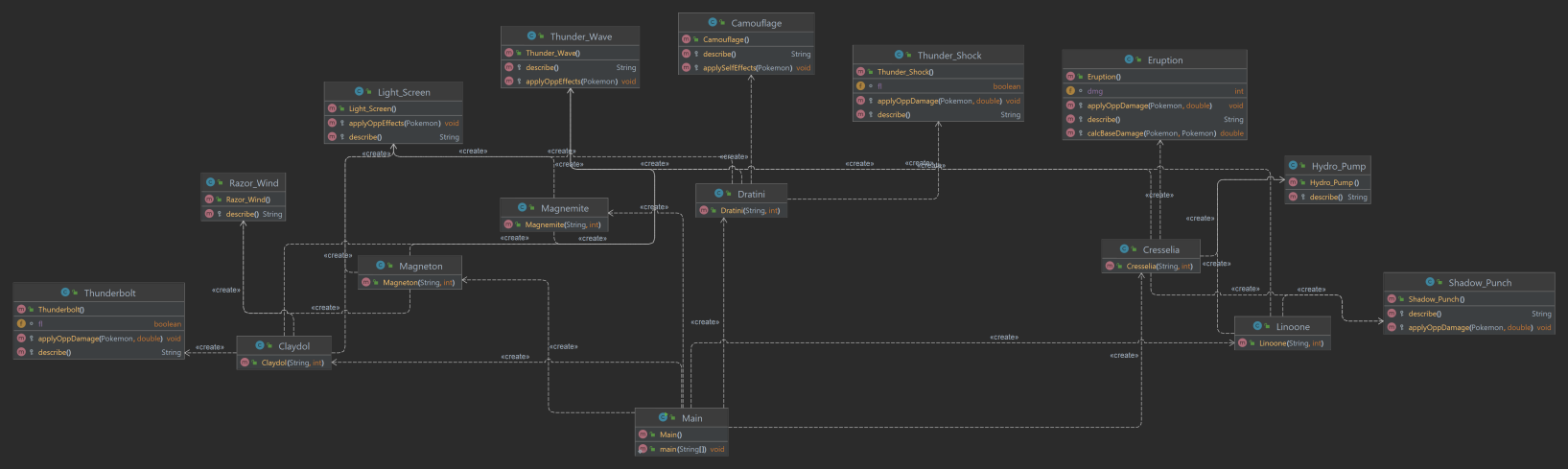


Рис. 2

# Исходный код программы:

// Main

import ru.ifmo.se.pokemon.\*;  
import My\_Pokemons.\*;  
  
public class Main  
{  
 public static void main(String[] args)  
 {  
 Battle b = new Battle();  
 Dratini p11 = new Dratini("(Dratini)", 10);  
 Linoone p12 = new Linoone("(Linoone)", 10);  
 Cresselia p13 = new Cresselia("(Cresselia)", 10);  
  
 Magnemite p21 = new Magnemite("(Magnemite)", 1);  
 Magneton p22 = new Magneton("(Magneton)", 1);  
 Claydol p23 = new Claydol("(Claydol)", 1);  
  
  
 b.addAlly(p11);  
 b.addAlly(p12);  
 b.addAlly(p13);  
 b.addFoe(p23);  
 b.addFoe(p22);  
 b.addFoe(p21);  
  
 b.go();  
 }  
}

Package My\_Pokemons:

// Claydol

package My\_Pokemons;  
import ru.ifmo.se.pokemon.\*;  
import Attacks.\*;  
  
  
public class Claydol extends Pokemon  
{  
 public Claydol(String name, int lvl)  
 {  
 super(name, lvl);  
 setStats(50, 60, 95, 120, 70, 70);  
 setType(Type.*ELECTRIC*, Type.*STEEL*);  
 setMove(new Light\_Screen(), new Thunder\_Wave(), new Razor\_Wind(), new Thunderbolt());  
 }  
}

// Cresselia

package My\_Pokemons;  
import ru.ifmo.se.pokemon.\*;  
import Attacks.\*;  
  
  
public class Cresselia extends Pokemon  
{  
 public Cresselia(String name, int lvl)  
 {  
 super(name, lvl);  
 setStats(120, 70, 110, 75, 120, 85);  
 setType(Type.*PSYCHIC*);  
 setMove(new Thunder\_Wave(), new Shadow\_Punch(), new Hydro\_Pump(), new Eruption());  
 }  
}

// Dratini

package My\_Pokemons;  
import ru.ifmo.se.pokemon.\*;  
import Attacks.\*;  
  
  
public class Dratini extends Pokemon  
{  
 public Dratini(String name, int lvl)  
 {  
 super(name, lvl);  
 setStats(41, 64, 45, 50, 50, 50);  
 setType(Type.*DRAGON*);  
 setMove(new Light\_Screen(), new Thunder\_Shock(), new Thunder\_Wave(), new Camouflage());  
 }  
}

// Linoone

package My\_Pokemons;  
import ru.ifmo.se.pokemon.\*;  
import Attacks.\*;  
  
  
public class Linoone extends Pokemon  
{  
 public Linoone(String name, int lvl)  
 {  
 super(name, lvl);  
 setStats(78, 70, 61, 50, 61, 100);  
 setType(Type.*NORMAL*);  
 setMove(new Thunder\_Wave(), new Shadow\_Punch(), new Hydro\_Pump());  
 }  
}

// Magnemite

package My\_Pokemons;  
import ru.ifmo.se.pokemon.\*;  
import Attacks.\*;  
  
  
public class Magnemite extends Pokemon  
{  
 public Magnemite(String name, int lvl)  
 {  
 super(name, lvl);  
 setStats(25, 35, 70, 95, 55, 45);  
 setType(Type.*ELECTRIC*, Type.*STEEL*);  
 setMove(new Light\_Screen(), new Thunder\_Wave());  
 }  
}

// Magneton

package My\_Pokemons;  
import ru.ifmo.se.pokemon.\*;  
import Attacks.\*;  
  
  
public class Magneton extends Pokemon  
{  
 public Magneton(String name, int lvl)  
 {  
 super(name, lvl);  
 setStats(50, 60, 95, 120, 70, 70);  
 setType(Type.*ELECTRIC*, Type.*STEEL*);  
 setMove(new Light\_Screen(), new Thunder\_Wave(), new Razor\_Wind());  
 }  
}

// Thunderbolt

package My\_Pokemons;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Thunderbolt extends SpecialMove  
{  
 public Thunderbolt()  
 {  
 super(Type.*ELECTRIC*, 90, 100);  
 }  
 boolean fl = false;  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage)  
 {  
 int ch = (int)(Math.*random*() \* 101);  
 if (ch <= 10)  
 {  
 Effect.*paralyze*(def);  
 fl = true;  
 }  
  
 }  
  
 @Override  
 protected String describe()  
 {  
 if (fl)  
 return "Использует Thunderbolt: наносит урон и парализует противника";  
  
 return "Использует Thunderbolt: наносит урон";  
 }  
}

Package Attacks:

// Camouflage

package Attacks;  
import ru.ifmo.se.pokemon.\*;  
  
public class Camouflage extends StatusMove  
{  
 public Camouflage()  
 {  
 super(Type.*NORMAL*, 0, 100); //new Thunder\_Shock(), new Thunder\_Wave()  
 }  
  
 @Override  
 protected void applySelfEffects(Pokemon p)  
 {  
 ;  
 }  
  
 @Override  
 protected String describe()  
 {  
 return "Использует Camouflage. Тип не изменился, поскольку местность отсутствует";  
 }  
}

// Eruption

package Attacks;  
  
import ru.ifmo.se.pokemon.\*;  
*/\*\*точно готово\*\*/*public class Eruption extends SpecialMove  
{  
 public Eruption()  
 {  
 super(Type.*FIRE*, 150, 100);  
 }  
 int dmg;  
 @Override  
 protected double calcBaseDamage(Pokemon att, Pokemon def)  
 {  
 //System.out.println((150 \* att.getHP() / att.getStat(Stat.HP)) + " " + att.getStat(Stat.HP));  
 dmg = (int) Math.*round*(150 \* att.getHP() / att.getStat(Stat.*HP*));  
 return (int) Math.*round*(150 \* att.getHP() / att.getStat(Stat.*HP*));  
 }  
  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage)  
 {  
 def.setMod(Stat.*HP*, dmg);  
 }  
  
 @Override  
 protected String describe()  
 {  
 return "покемон применяет Eruption и наносит урон";  
 }  
}

// Hydro\_Pump

package Attacks;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Hydro\_Pump extends SpecialMove  
{  
 public Hydro\_Pump()  
 {  
 super(Type.*WATER*, 110, 80);  
 }  
  
  
 @Override  
 protected String describe()  
 {  
 return "покемон применяет Hydro Pump и наносит урон";  
 }  
}

// Light\_Screen

package Attacks;  
import ru.ifmo.se.pokemon.\*;  
  
  
*/\*\*Light Screen reduces damage from Special attacks by 50%, for 5 turns. Its effects apply to all Pokémon on the user's side of the field.\*\*/*public class Light\_Screen extends StatusMove  
{  
 public Light\_Screen()  
 {  
 super(Type.*PSYCHIC*, 0, 100);  
  
 }  
  
 @Override  
 protected void applyOppEffects(Pokemon p)  
 {  
 Effect ls\_eff = new Effect().turns(5).stat(Stat.*SPECIAL\_ATTACK*, -2);  
 p.addEffect(ls\_eff);  
 }  
  
 @Override  
 protected String describe()  
 {  
 return "использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов";  
 }  
}

// Razor\_Wind

package Attacks;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Razor\_Wind extends SpecialMove  
{  
 public Razor\_Wind()  
 {  
 super(Type.*NORMAL*, 80, 100);  
 }  
 //public static int damag;  
  
 /\*@Override  
 protected void applyOppDamage(Pokemon def, double damage)  
 {  
 damag = (int) damage;  
 }\*/  
  
 @Override  
 protected String describe()  
 {  
 return "покемон применяет Razor\_Wind и создаёт вихрь и наносит урон";  
 }  
}

// Shadow\_Punch

package Attacks;  
  
import ru.ifmo.se.pokemon.\*;  
  
public class Shadow\_Punch extends PhysicalMove  
{  
 public Shadow\_Punch()  
 {  
 super(Type.*GHOST*, 60, 100);  
 }  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage)  
 {  
 // System.out.println(damage + " " + def.getHP());  
 def.setMod(Stat.*HP*, (int)Math.*round*(damage));  
 // System.out.println(damage + " " + def.getHP());  
 }  
  
 @Override  
 protected String describe()  
 {  
 return "покемон применяет Shadow Punch и наносит урон";  
 }  
}

//Thunder\_Shock

package Attacks;  
import ru.ifmo.se.pokemon.\*;  
  
*/\*\*Удар грома наносит урон и с вероятностью 10% парализует цель.\*\*/*public class Thunder\_Shock extends SpecialMove  
{  
 public Thunder\_Shock()  
 {  
 super(Type.*ELECTRIC*, 40, 100);  
 }  
 boolean fl = false;  
 @Override  
 protected void applyOppDamage(Pokemon def, double damage)  
 {  
 int ch = (int)(Math.*random*() \* 101);  
 if (ch <= 10)  
 {  
 Effect.*paralyze*(def);  
 fl = true;  
 }  
  
 }  
  
 @Override  
 protected String describe()  
 {  
 if (fl)  
 return "наносит урон и парализует противника";  
  
 return "наносит урон";  
 }  
}

// Thunder\_Wave

package Attacks;  
import ru.ifmo.se.pokemon.\*;  
  
import java.util.logging.Level;  
  
public class Thunder\_Wave extends StatusMove {  
 public Thunder\_Wave()  
 {  
 super(Type.*ELECTRIC*, 0, 90);  
 }  
 @Override  
 protected void applyOppEffects(Pokemon p)  
 {  
 Effect.*paralyze*(p);  
 }  
  
 @Override  
 protected String describe()  
 {  
 return "покемон применяет Thunder Wave: парализует противника";  
 }  
}

# Ссылка на гитхаб с исходным кодом:

<https://github.com/frizyyu/lab2_prog>

# Результат работы программы:

Dratini (Dratini) из команды красных вступает в бой!

Claydol (Claydol) из команды белых вступает в бой!

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 4 здоровья.

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 4 здоровья.

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 5 здоровья.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) восстанавливает 1 здоровья.

Dratini (Dratini) наносит урон.

Claydol (Claydol) покемон применяет Thunder Wave: парализует противника.

Dratini (Dratini) парализован

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) покемон применяет Thunder Wave: парализует противника.

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) покемон применяет Thunder Wave: парализует противника.

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) покемон применяет Thunder Wave: парализует противника.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) восстанавливает 1 здоровья.

Dratini (Dratini) покемон применяет Thunder Wave: парализует противника.

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) покемон применяет Thunder Wave: парализует противника.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) восстанавливает 1 здоровья.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) покемон применяет Thunder Wave: парализует противника.

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 4 здоровья.

Dratini (Dratini) покемон применяет Thunder Wave: парализует противника.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 3 здоровья.

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 3 здоровья.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) восстанавливает 1 здоровья.

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 4 здоровья.

Dratini (Dratini) покемон применяет Thunder Wave: парализует противника.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 5 здоровья.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) наносит урон.

Критический удар!

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Dratini (Dratini) Использует Camouflage. Тип не изменился, поскольку местность отсутствует.

Claydol (Claydol) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) покемон применяет Thunder Wave: парализует противника.

Dratini (Dratini) наносит урон и парализует противника.

Критический удар!

Claydol (Claydol) Использует Thunderbolt: наносит урон.

Dratini (Dratini) использует Light\_Screen и уменьшает урон от специальных атак на 50% на 5 ходов.

Claydol (Claydol) покемон применяет Razor\_Wind и создаёт вихрь и наносит урон.

Dratini (Dratini) теряет 4 здоровья.

Dratini (Dratini) теряет сознание.

Linoone (Linoone) из команды красных вступает в бой!

Linoone (Linoone) покемон применяет Hydro Pump и наносит урон.

Claydol (Claydol) теряет 14 здоровья.

Claydol (Claydol) теряет сознание.

Magneton (Magneton) из команды белых вступает в бой!

Linoone (Linoone) покемон применяет Shadow Punch и наносит урон.

Magneton (Magneton) теряет 13 здоровья.

Magneton (Magneton) теряет сознание.

Magnemite (Magnemite) из команды белых вступает в бой!

Linoone (Linoone) покемон применяет Shadow Punch и наносит урон.

Magnemite (Magnemite) теряет 13 здоровья.

Magnemite (Magnemite) теряет сознание.

В команде белых не осталось покемонов.

Команда красных побеждает в этом бою!

# Вывод:

В ходе выполнения данной лабораторной работы я получил классы, наследующиеся от базового класса Pokemon. Я заметил, что методы, имеющие модификатор доступа private, доступны только внутри класса, в котором они определены, переопределить такой метод нельзя, как и метод с модификатором final, что означает завершённый метод. Методы с модификатором доступа public можно переопределить, а также вызвать при обращении к объекту класса. Методы с модификатором static переопределять нельзя, но можно вызывать, обращаясь к самому классу, а не к его объекту. Также при наследовании класса, дочернему классу будут доступны для вызова все методы родительского класса, также будет возможность переопределить методы, имеющие модификатор public.

# Список источников:

**Pockemon Документация** <https://se.ifmo.ru/~tony/doc/>

**Pokemon Database** <https://pokemondb.net>

**Pokemon Database** <https://bulbagarden.net>