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

Факультет программной инженерии и компьютерной техники

**Отчет**

По лабораторной работа № 2 по дисциплине “Программирование”

Вариант № 21111

Студент:

Мирзаитов Тимур

Группа P3112

Преподаватель:

Харитонова Анастасия Евгеньевна

Санкт-Петербург 2023

Оглавление

[Текст Задания 2](#_Toc147666755)

[Диаграмма классов реализованной объектной модели 3](#_Toc147666756)

[Исходный код программы 3](#_Toc147666757)

[Результат работы программы 12](#_Toc147666758)

[Вывод 13](#_Toc147666759)

# 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.

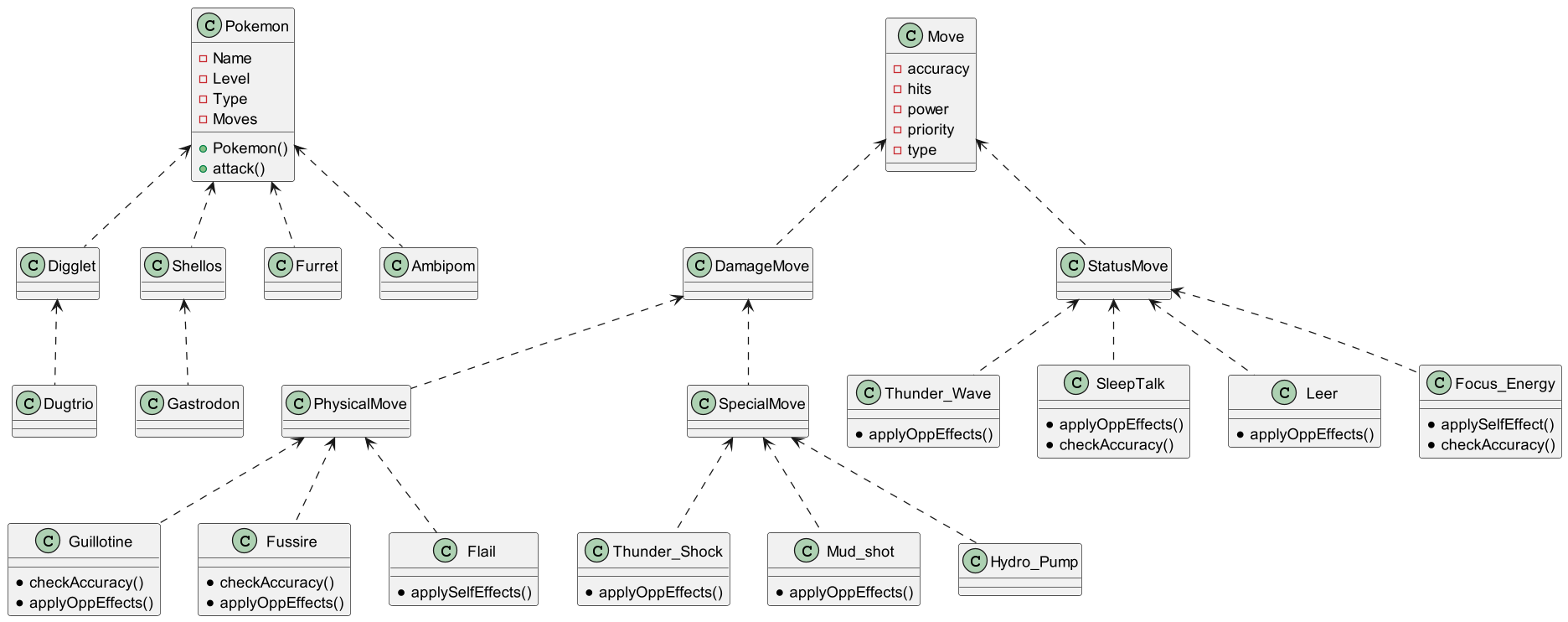
Базовые классы и симулятор сражения находятся в jar-архиве (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - здесь.

Вариант работы (рис. 1):



Рисунок 1.1 Вариант№21111 ЛР№2

# Диаграмма классов реализованной объектной модели



# Исходный код программы

Main.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.Pokemon**;  **import** **ru.ifmo.se.pokemon.Battle**;  **public** **class** **Main** {  **public** **static** **void** **main**(String[] args) {  Battle b = **new** Battle();  Pokemon p1 = **new** Diglett("Diglett", **1**);  Pokemon p2 = **new** Furret("Furret", **1**);  Pokemon p3 = **new** Dugtrio("Dugtrio", **1**);  Pokemon p4 = **new** Ambipom("Ambipom", **1**);  Pokemon p5 = **new** Gastrodon("Gastrodon", **1**);  Pokemon p6 = **new** Shellos("Shellos", **1**);  b.addAlly(p1);  b.addAlly(p2);  b.addAlly(p3);  b.addFoe(p4);  b.addFoe(p5);  b.addFoe(p6);  b.go();  }  } |

Ambipom.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.Pokemon**;  **import** **ru.ifmo.se.pokemon.Type**;  **public** **class** **Ambipom** **extends** Pokemon {  **public** **Ambipom**(String \_name, **int** \_lvl)  {  **super**(\_name, \_lvl);  **this**.setType(Type.NORMAL);  **this**.setStats(**75**, **100**, **66**, **60**, **66**, **115**);  **this**.setMove(**new** Thunder\_Shock(), **new** Hydro\_Pump(), **new** Mud\_Shot(), **new** Guillotine());  }  } |

Diglett.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.Pokemon**;  **import** **ru.ifmo.se.pokemon.Type**;  **public** **class** **Diglett** **extends** Pokemon {  **public** **Diglett**(String \_name, **int** \_lvl){  **super**(\_name, \_lvl);  **this**.setStats(**10**, **55**, **25**, **35**, **45**, **95**);  **this**.setType(Type.GROUND);  **this**.setMove(**new** Focus\_Energy(), **new** Thunder\_Wave(), **new** Thunder\_Shock());  }  } |

Dugtrio.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | **package** ru.Timur.Pokemon;  **public** **class** **Dugtrio** **extends** Diglett{  **public** **Dugtrio**(String \_name, **int** \_lvl)  {  **super**(\_name, \_lvl);  **this**.setStats(**35**, **100**, **50**, **50**, **70**, **120**);  **this**.addMove(**new** Fissure());  }  } |

Furret.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.Pokemon**;  **import** **ru.ifmo.se.pokemon.Type**;  **public** **class** **Furret** **extends** Pokemon {  **double** maxHP;  **public** **Furret**(String \_name, **int** \_lvl)  {  **super**(\_name, \_lvl);  **this**.setType(Type.NORMAL);  **this**.setStats(**85**, **76**, **64**, **45**, **55**, **90**);  maxHP = **85**;  setMove(**new** Flail(**this**), **new** Leer(), **new** SleepTalk());  }  **public** **double** **getMaxHP**()  {  **return** maxHP;  }  **public** **void** **setMaxHP**(**double** \_maxHP)  {  maxHP = \_maxHP;  }  } |

Gastrodon.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.Type**;  **public** **class** **Gastrodon** **extends** Shellos{  **double** \_HP = **111**;  **double** \_ATT = **83**;  **double** \_DEF = **68**;  **double** \_SpATT = **92**;  **double** \_SpDEF = **82**;  **double** \_SPEED = **39**;  **public** **Gastrodon**(String \_name, **int** \_lvl){  **super**(\_name, \_lvl);  **this**.setStats(\_HP, \_ATT, \_DEF, \_SpATT, \_SpDEF, \_SPEED);  **this**.addType(Type.GROUND);  **this**.addMove(**new** Mud\_Shot());  }  } |

Shellos.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.Pokemon**;  **import** **ru.ifmo.se.pokemon.Type**;  **public** **class** **Shellos** **extends** Pokemon {  **double** \_HP = **76**;  **double** \_ATT = **48**;  **double** \_DEF = **48**;  **double** \_SpATT = **57**;  **double** \_SpDEF = **62**;  **double** \_SPEED = **34**;  **public** **Shellos**(String \_name, **int** \_lvl)  {  **super**(\_name, \_lvl);  **this**.setType(Type.WATER);  **this**.setStats(\_HP, \_ATT, \_DEF, \_SpATT, \_SpDEF, \_SPEED);  **this**.setMove(**new** Thunder\_Shock(), **new** Hydro\_Pump());  }  } |

Fissure

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Fissure** **extends** PhysicalMove {  **public** **Fissure**()  {  **super**(Type.GROUND, **0**, **30**);  }  **@Override**  **protected** **boolean** **checkAccuracy**(Pokemon att, Pokemon def)  {  **return** **30** + att.getLevel() - def.getLevel() > Math.random();  }  **@Override**  **protected** **void** **applyOppEffects**(Pokemon opp){  Effect eff = **new** Effect().stat(Stat.HP, **0**);  opp.addEffect(eff);  }  } |

Flail

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Flail** **extends** PhysicalMove {  **final** **double** maxHP;  **public** **Flail**(Pokemon self)  {  **super**(Type.NORMAL, **20**, **100**);  maxHP = self.getHP();  }  **@Override**  **protected** **void** **applySelfEffects**(Pokemon self) {  **double** N = **48**\*self.getHP()/maxHP;  **if**(N >= **0** && N < **2**)  {  power = **200**;  } **else** **if** (N >= **2** && N < **4**) {  power = **150**;  } **else** **if** (N >= **4** && N < **9**) {  power = **100**;  } **else** **if** (N >= **9** && N < **16**) {  power = **80**;  } **else** **if** (N >= **16** && N < **32**) {  power = **40**;  } **else** **if** (N >= **32** && N < **48**) {  power = **20**;  }  }  } |

Focus\_Energy

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Focus\_Energy** **extends** StatusMove {  **public** **Focus\_Energy**()  {  **super**(Type.NORMAL, **0**, **0**);  }  **@Override**  **protected** **void** **applySelfEffects**(Pokemon self){  Effect eff = **new** Effect().stat(Stat.SPEED, +**10**);  self.addEffect(eff);  }  **@Override**  **protected** **boolean** **checkAccuracy**(Pokemon att, Pokemon def) {  **return** **true**;  }  } |

Guillotine

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **package** ru.Timur.Pokemon;  **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Guillotine** **extends** PhysicalMove {  **public** **Guillotine**(){  **super**(Type.NORMAL, **0**, **30**);  }  **@Override**  **protected** **boolean** **checkAccuracy**(Pokemon att, Pokemon def)  {  **return** **30** + att.getLevel() - def.getLevel() > Math.random();  }  **@Override**  **protected** **void** **applyOppEffects**(Pokemon opp){  Effect eff = **new** Effect().stat(Stat.HP, **0**);  opp.addEffect(eff);  }  } |

Hydro\_Pump

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | **package** ru.Timur.Move;  **import** **ru.ifmo.se.pokemon.SpecialMove**;  **import** **ru.ifmo.se.pokemon.Type**;  **public** **class** **Hydro\_Pump** **extends** SpecialMove {  **public** **Hydro\_Pump**()  {  **super**(Type.WATER, **110**, **80**);  }  007D |

Leer

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | **package** ru.Timur.Move;  **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Leer** **extends** StatusMove {  **public** **Leer**()  {  **super**(Type.NORMAL, **0**, **100**);  }  **@Override**  **protected** **void** **applyOppEffects**(Pokemon p)  {  Effect eff = **new** Effect().stat(Stat.DEFENSE, -**1**).turns(-**1**);  p.addEffect(eff);  }  } |

Mud\_Shot

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | **package** ru.Timur.Move;  **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Mud\_Shot** **extends** SpecialMove {  **public** **Mud\_Shot**()  {  **super**(Type.GROUND, **55**, **95**);  }  **@Override**  **protected** **void** **applyOppEffects**(Pokemon opp)  {  opp.setMod(Stat.SPEED, -**1**);  }  } |

SleepTalk

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32 | **package** ru.Timur.Move;  **import** **ru.ifmo.se.pokemon.Pokemon**;  **import** **ru.ifmo.se.pokemon.Status**;  **import** **ru.ifmo.se.pokemon.StatusMove**;  **import** **ru.ifmo.se.pokemon.Type**;  **import** **java.security.cert.PolicyNode**;  **import** **java.util.ArrayList**;  **public** **class** **SleepTalk** **extends** StatusMove {  **public** **SleepTalk**()  {  **super**(Type.NORMAL, **0**, **100**);  }  **@Override**  **protected** **void** **applySelfEffects**(Pokemon self)  {  **if**(self.getCondition() == Status.SLEEP)  {  self.prepareMove();  }  }  **@Override**  **protected** **boolean** **checkAccuracy**(Pokemon att, Pokemon def){  **return** **true**;  }  } |

Thunder\_Shock

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | **package** ru.Timur.Move;  **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Thunder\_Shock** **extends** SpecialMove {  **public** **Thunder\_Shock**()  {  **super**(Type.ELECTRIC, **40**, **100**);  }  **@Override**  **protected** **void** **applyOppEffects**(Pokemon opp)  {  Effect eff = **new** Effect().chance(**0.1**).condition(Status.PARALYZE);  **if**(!opp.hasType(Type.ELECTRIC))  {  opp.setCondition(eff);  }  }  } |

Thunder\_Wave

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | **package** ru.Timur.Move;  **import** **ru.ifmo.se.pokemon.\***;  **import** **java.lang.reflect.Parameter**;  **public** **class** **Thunder\_Wave** **extends** StatusMove {  **public** **Thunder\_Wave**() {  **super**(Type.ELECTRIC, **0**, **90**);  }  **@Override**  **protected** **void** **applyOppEffects**(Pokemon opp)  {  Effect eff = **new** Effect().condition(Status.PARALYZE);  **if**(!opp.hasType(Type.ELECTRIC))  {  opp.setCondition(eff);  }  }  } |

# Результат работы программы

Diglett Diglett из команды красных вступает в бой!

Ambipom Ambipom из команды полосатых вступает в бой!

Ambipom Ambipom атакует.

Diglett Diglett теряет 13 здоровья.

Diglett Diglett теряет сознание.

Furret Furret из команды красных вступает в бой!

Ambipom Ambipom атакует.

Furret Furret теряет 5 здоровья.

Furret Furret уменьшает скорость.

Furret Furret атакует.

Furret Furret атакует.

Ambipom Ambipom теряет 7 здоровья.

Ambipom Ambipom атакует.

Furret Furret теряет 5 здоровья.

Furret Furret атакует.

Ambipom Ambipom теряет 4 здоровья.

Ambipom Ambipom атакует.

Критический удар!

Furret Furret теряет 6 здоровья.

Furret Furret уменьшает скорость.

Furret Furret теряет сознание.

Dugtrio Dugtrio из команды красных вступает в бой!

Dugtrio Dugtrio атакует.

Ambipom Ambipom атакует.

Dugtrio Dugtrio теряет 1 здоровья.

Dugtrio Dugtrio не замечает воздействие типа ELECTRIC

Dugtrio Dugtrio атакует.

Ambipom Ambipom теряет 4 здоровья.

Ambipom Ambipom теряет сознание.

Gastrodon Gastrodon из команды полосатых вступает в бой!

Dugtrio Dugtrio атакует.

Gastrodon Gastrodon не замечает воздействие типа ELECTRIC

Gastrodon Gastrodon атакует.

Dugtrio Dugtrio теряет 13 здоровья.

Dugtrio Dugtrio теряет сознание.

В команде красных не осталось покемонов.

Команда полосатых побеждает в этом бою!

# Вывод

В процессе выполнения лабораторной работы я получил навыки использования объектно-ориентированного подхода программирования при использовании языка Java. Научился работать с классами, конструкторами, полями и модификаторами доступа.

В процессе выполнения лабораторной работы я получил навыки использования ООП используя язык Java. Научился наследовать классы, переопределять методы, создавать пакеты и подключать внешние библиотеки.