Университет ИТМО

Факультет программной инженерии и компьютерной техники



**Лабораторная работа №2**

Вариант 311961

Выполнил: Исупов Денис Васильевич

Группа: P3119

Преподаватель: Письмак Алексей Евгеньевич

г. Санкт-Петербург

2021 год

Оглавление

[1. Текст задания 3](#_Toc86832898)

[2. Диаграмма моделей 3](#_Toc86832899)

[3. Исходный код 3](#_Toc86832900)

[Aipom.java 4](#_Toc86832901)

[Ambipom.java 4](#_Toc86832902)

[Attacks.java 4](#_Toc86832903)

[Guzzlord.java 7](#_Toc86832904)

[Main.java 7](#_Toc86832905)

[Nuzleaf.java 7](#_Toc86832906)

[Seedot.java 7](#_Toc86832907)

[Shiftry.java 8](#_Toc86832908)

[4. Результат работы программы 8](#_Toc86832909)

[5. Выводы по работе 8](#_Toc86832910)

1. **Текст задания**

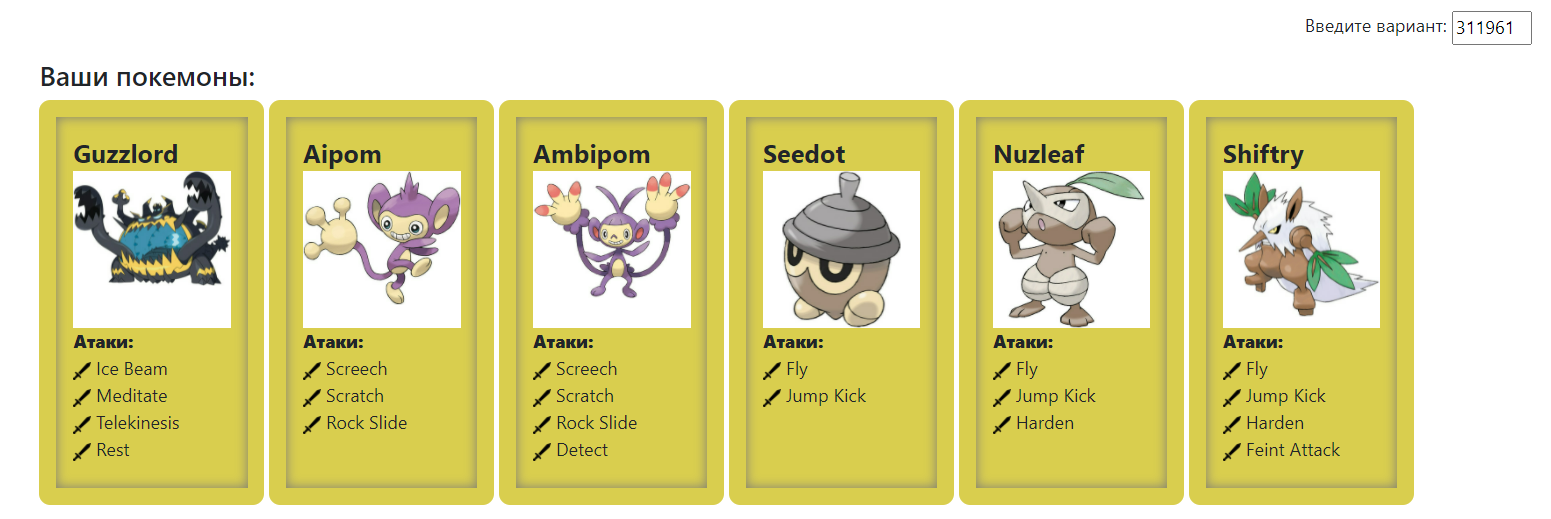
На основе базового класса Pokemon написать свои классы для заданных видов покемонов. Каждый вид покемона должен иметь один или два типа и стандартные базовые характеристики:

* 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.



1. Диаграмма моделей

<https://raw.githubusercontent.com/IsupDen/programming-lab/main/lab2/UML.svg?token=AWJRW7YHNWVE3LUPWK47R73BRN2N6>

1. Исходный код

Aipom.java

**import** **ru.ifmo.se.pokemon.\***;

**public** **class** **Aipom** **extends** Pokemon {

**public** Aipom(String name, int lvl) {

**super**(name, lvl);

setType(Type.NORMAL);

setStats(55, 70, 55, 40, 55, 85);

setMove(**new** Screech(), **new** Scratch(), **new** Rock\_Slide());

}

}

Ambipom.java

**import** **ru.ifmo.se.pokemon.\***;

**public** **class** **Ambipom** **extends** Aipom {

**public** Ambipom(String name, int lvl) {

**super**(name, lvl);

setType(Type.NORMAL);

setStats(75, 100, 66, 60, 66, 115);

addMove(**new** Detect());

}

}

Attacks.java

**import** **ru.ifmo.se.pokemon.\***;

**class** **Scratch** **extends** PhysicalMove{

**protected** Scratch() {**super**(Type.NORMAL, 40, 120);}

@Override

**protected** String describe() {

**return** "use Scratch (It deals damage with no additional effects)";

}

}

**class** **Rock\_Slide** **extends** PhysicalMove {

**protected** Rock\_Slide() {

**super**(Type.ROCK, 75, 90);

}

@Override

**protected** void applyOppEffects(Pokemon def) {

**if** (Math.random() < 0.3) {

Effect.flinch(def);

}

}

@Override

**protected** String describe() {

**return** "use Rock Slide (Has a 30% chance of causing the target to flinch)";

}

}

**class** **Jump\_Kick** **extends** PhysicalMove{

**protected** Jump\_Kick() {**super**(Type.FIGHTING, 100, 95);}

@Override

**protected** void applySelfDamage(Pokemon att, double v) {

**if** (Math.random() <0.5) {

att.setMod(Stat.HP, (int)Math.round(v));

}

}

@Override

**protected** String describe() {

**return** "use Jump Kick (If the user misses, it takes half the damage it would have inflicted in recoil)";

}

}

**class** **Harden** **extends** StatusMove{

**protected** Harden() {**super**(Type.NORMAL, 0, 0);}

@Override

**protected** void applySelfEffects(Pokemon att) {

Effect defense\_up = **new** Effect().turns(0).stat(Stat.DEFENSE, +1);

att.addEffect(defense\_up);

}

@Override

**protected** String describe() {

**return** "use Harden (Raises the user's Defense by one stage)";

}

}

**class** **Ice\_Beam** **extends** SpecialMove{

**protected** Ice\_Beam() {**super**(Type.ICE, 90, 100);}

@Override

**protected** void applyOppEffects(Pokemon def) {

**if** (Math.random() <= 0.1) {

Effect.freeze(def);

}

}

@Override

**protected** String describe() {

**return** "use Ice Beam (Has a 10% chance to freeze the target)";

}

}

**class** **Meditate** **extends** StatusMove{

**protected** Meditate() {**super**(Type.PSYCHIC, 0, 0);}

@Override

**protected** void applySelfEffects(Pokemon att) {

Effect attack\_up = **new** Effect().turns(0).stat(Stat.ATTACK, +1);

att.addEffect(attack\_up);

}

@Override

**protected** String describe() {

**return** "use Meditate (Raises the user's Attack by one stage)";

}

}

**class** **Telekinesis** **extends** StatusMove{

**protected** Telekinesis() {**super**(Type.PSYCHIC, 0, 0);}

@Override

**protected** String describe() {

**return** "use Telekinesis (Moves have 100% accuracy against the target for three turns)";

}

}

**class** **Rest** **extends** StatusMove{

**protected** Rest() {**super**(Type.PSYCHIC, 0, 0);}

@Override

**protected** void applySelfEffects(Pokemon att) {

Effect e1 = **new** Effect().turns(0).stat(Stat.HP, -99999);

Effect e2 = **new** Effect().turns(2).condition(Status.SLEEP);

att.addEffect(e1);

att.setCondition(e2);

}

@Override

**protected** String describe() {

**return** "use Rest (User sleeps for two turns, completely healing itself)";

}

}

**class** **Screech** **extends** StatusMove{

**protected** Screech() {**super**(Type.NORMAL, 0, 85);}

@Override

**protected** void applyOppEffects(Pokemon def) {

Effect defense\_down = **new** Effect().turns(0).stat(Stat.DEFENSE, -2);

def.addEffect(defense\_down);

}

@Override

**protected** String describe() {

**return** "use Screech (Lowers the target's Defense by two stages)";

}

}

**class** **Detect** **extends** StatusMove{

**protected** Detect() {**super**(Type.PSYCHIC, 0, 0);}

@Override

**protected** String describe() {

**return** "use Detect (Prevents any moves from hitting the user this turn)";

}

}

**class** **Fly** **extends** PhysicalMove{

**protected** Fly() {**super**(Type.FLYING, 90, 95);}

@Override

**protected** String describe() {

**return** "use Fly (User flies high into the air, dodging all attacks, and hits next turn)";

}

}

**class** **Feint\_Attack** **extends** PhysicalMove{

**protected** Feint\_Attack() {**super**(Type.DARK, 60, 10000000);}

@Override

**protected** String describe() {

**return** "use Feint Attack (Never misses)";

}

}

Guzzlord.java

**import** **ru.ifmo.se.pokemon.\***;

**public** **class** **Guzzlord** **extends** Pokemon {

**public** Guzzlord(String name, int lvl) {

**super**(name, lvl);

setType(Type.DARK, Type.DRAGON);

setStats(223, 101, 53, 97, 53, 43);

setMove(**new** Meditate(), **new** Telekinesis(), **new** Ice\_Beam(), **new** Rest());

}

}

Main.java

**import** **ru.ifmo.se.pokemon.\***;

**public** **class** **Main** {

**public** **static** void main(String[] args) {

Battle field = **new** Battle();

field.addAlly(**new** Aipom("Oleg", 1));

field.addAlly(**new** Ambipom("Roma", 2));

field.addAlly(**new** Guzzlord("Sasha", 1));

field.addFoe(**new** Seedot("Pasha", 4));

field.addFoe(**new** Nuzleaf("Dima", 3));

field.addFoe(**new** Shiftry("Kirill", 2));

field.go();

}

}

Nuzleaf.java

**import** **ru.ifmo.se.pokemon.\***;

**public** **class** **Nuzleaf** **extends** Seedot {

**public** Nuzleaf(String name, int lvl) {

**super**(name, lvl);

setType(Type.GRASS, Type.DARK);

setStats(70, 70, 40, 60, 40, 60);

addMove(**new** Harden());

}

}

Seedot.java

**import** **ru.ifmo.se.pokemon.\***;

**public** **class** **Seedot** **extends** Pokemon {

**public** Seedot(String name, int lvl) {

**super**(name, lvl);

setType(Type.GRASS);

setStats(40, 40, 50, 30, 30, 30);

setMove(**new** Fly(), **new** Jump\_Kick());

}

}

Shiftry.java

**import** **ru.ifmo.se.pokemon.\***;

**public** **class** **Shiftry** **extends** Nuzleaf {

**public** Shiftry(String name, int lvl) {

**super**(name, lvl);

setType(Type.GRASS, Type.DARK);

setStats(90, 100, 60, 90, 60, 80);

addMove(**new** Feint\_Attack());

}

}

1. Результат работы программы

Aipom Oleg из команды зеленых вступает в бой!

Seedot Pasha из команды синих вступает в бой!

Seedot Pasha use Jump Kick (If the user misses, it takes half the damage it would have inflicted in recoil).

Aipom Oleg теряет 15 здоровья.

Seedot Pasha теряет 15 здоровья.

Aipom Oleg теряет сознание.

Ambipom Roma из команды зеленых вступает в бой!

Ambipom Roma use Screech (Lowers the target's Defense by two stages).

Seedot Pasha use Fly (User flies high into the air, dodging all attacks, and hits next turn).

Ambipom Roma теряет 6 здоровья.

Ambipom Roma use Screech (Lowers the target's Defense by two stages).

Seedot Pasha use Jump Kick (If the user misses, it takes half the damage it would have inflicted in recoil).

Ambipom Roma теряет 14 здоровья.

Ambipom Roma теряет сознание.

Guzzlord Sasha из команды зеленых вступает в бой!

Seedot Pasha use Fly (User flies high into the air, dodging all attacks, and hits next turn).

Guzzlord Sasha теряет 5 здоровья.

Guzzlord Sasha промахивается

Seedot Pasha use Fly (User flies high into the air, dodging all attacks, and hits next turn).

Guzzlord Sasha теряет 10 здоровья.

Guzzlord Sasha промахивается

Seedot Pasha use Jump Kick (If the user misses, it takes half the damage it would have inflicted in recoil).

Guzzlord Sasha теряет 15 здоровья.

Guzzlord Sasha теряет сознание.

В команде зеленых не осталось покемонов.

Команда синих побеждает в этом бою!

1. Выводы по работе

Во время выполнения данной лабораторной работы я изучил основные принципы ООП, научился работать с методами, классами, модификаторами доступами и сторонними библиотеками.