Санкт-Петербургский Национальный Исследовательский Университет

Информационных Технологий, Механики и Оптики

ФКТиУ, кафедра Вычислительной техники

Лабораторная работа №2

по дисциплине

«Программирование»

Вариант - 1231

Выполнил:

Анищенко Анатолий Алексеевич

Группа: З112

Санкт-Петербург

2018 г.

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](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/ce164749-d074-40ed-bc13-15e09d62600a). Документация в формате javadoc - в [zip-архиве](https://se.ifmo.ru/documents/10180/660917/doc.zip/244252e6-e4dd-4b52-8de1-b497d4402c7a).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах [http://poke-universe.ru](http://poke-universe.ru/), [http://pokemondb.net](http://pokemondb.net/),<http://veekun.com/dex/pokemon>

Вариант

|  |
| --- |
| +Pokemon |
| #name: String  -types: Type[]  -moves:Move[]  -preparedMove: Move  -stage: Efffect  -condition: Effect  -effects: List<Effect>  -confusion: int  #level: int  -base: double[] |
| +Pokemon(String)  +setStats(double, double, double,  double, double, double): void {final}  +getStat(Stat): double {final}  +hasType(Type): Boolean {final}  +addEffect(Effect): void {final}  ~setCondition(Effect): void {final}  +confuse(): void  +restore(): void {final}  +getHP(): double  +setMod(Stat, int): void {final}  +getTypes(): Type[] {final}  +getLevel(): int {final}  -getAttackChance(): double  +prepareMove(): void {final}  +isAlive(): boolean {final}  +attack(Pokemon): Boolean  +turn(): void {final}  +win(Pokemon): void  #setType(Type…): void{final}  #setMove(Move…): void {final}  #getPreparedMove(): Move {final}  +toString(): String |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| +Togedemaru | |  | |  | | |  | | |  | | |
|  | |  | |  | | |  | | |  | | |
| +Togedemaru(String, int) | |  | |  | | |  | | |  | | |
|  |  | |  | |  | | |  | | | |
| +Clauncher |  | | +Clawitzer | | |  | | |  | | | | |
|  |  | |  | | |  | | |  | | | | |
| +Clauncher(String, int) |  | | +Clawitzer(String, int) | | |  | | |  | | | | |
|  |  | |  | | |  | | |  | | | | |
| +Cyndaquil |  | | +Quilava | | |  | | | | |  | | | |
|  |  | |  | | |  | | | | |  | | | |
| + Cyndaquil(String, int) |  | | +Quilava(String, int) | | |  | | | | |  | | | |
|  |  | |  | | |  | | | | |  | | | |
|  |  | | +Typhlosion | | |  | | | | |  | | | |
|  |  | |  | | |  | | | | |  | | | |
|  |  | | +Typhlosion(String, int) | | |  | | | | |  | | | |

|  |
| --- |
| +Battleground |
|  |
| +main(String[]):void {static} |

Battle

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  | | |  |  | Move | |  | | |  | |  |
|  | |  | | |  |  | |  | | |  |  | |  |
|  | | DamageMove | | | | | |  | | |  | StatusMove | | |
|  | |  | | |  |  | |  | | |  |  | |  |
| PhysicalMove | | | | |  | SpecialMove | | | | |  |  | |  |
|  | |  | | |  |  | |  | | |  |  | |  |
| ~Pound |  | | | | | ~AncientPower | | | |  | | | ~FeatherDance | |
|  |  | | | | |  | | | |  | | |  | |
| +Pound() |  | | | | | +AncientPower()  #applySelfEfects(Pokemon): void | | | |  | | | +FeatherDance()  #applyOppEffects(Pokemon): void | |
|  | | | | |  |  | | | | |  |  | | |
| ~DoubleHit | | |  | | | ~FireBlast | | | | |  | ~Rest | | |
|  | | |  | | |  | | | | |  |  | | |
| +DoubleHit() | | |  | | | +FireBlast()  #applyOppEfects(Pokemon): void | | | | |  | +Rest()  #checkAccuracy(Pokemon, Pokemon): boolean  #applySelfEfects(Pokemon): void | | |
|  | | | | |  |  | | | | |  |  | | |
| ~RockTomb | | | |  | |  | | | | |  |  | | |
|  | | | |  | |  | | | | |  |  | | |
| +RockTomb()  #applyOppEffects(Pokemon): void | | | |  | |  | | | | |  |  | | |
|  | | | | |  |  | | | | |  |  | | |
| ~PoisonJab | | | |  | | ~VitalThrow | | | |  | |  | | |
|  | | | |  | |  | | | |  | |  | | |
| +PoisonJab()  #applyOppEffects(Pokemon): void | | | |  | | +VitalThrow()  #checkAccuracy(Pokemon, Pokemon): boolean | | | |  | |  | | |

**Togedemaru.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  30  31  32  33  34  35  36  37  38  39  40  41  42 | **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Togedemaru** **extends** Pokemon {  **public** **Togedemaru**(String name, **int** lvl) {  **super**(name, lvl);  setStats(**65**, **98**, **63**, **40**, **73**, **96**);  setType(Type.ELECTRIC, Type.STEEL);  setMove(**new** Pound(), **new** FeatherDance(), **new** DoubleHit(), **new** RockTomb());  }  }  **class** **Pound** **extends** PhysicalMove {  **public** **Pound**() {  **super**(Type.NORMAL, **40**, **1.0**);  }  }  **class** **FeatherDance** **extends** StatusMove {  **public** **FeatherDance**() {  **super**(Type.FLYING, **0**, **1.0**);  }    **protected** **void** **applyOppEffects**(Pokemon p) {  p.setMod(Stat.ATTACK, -**2**);  }  }  **class** **DoubleHit** **extends** PhysicalMove {  **public** **DoubleHit**() {  **super**(Type.NORMAL, **35**, **0.9**, **0**, **2**);  }  }  **class** **RockTomb** **extends** PhysicalMove {  **public** **RockTomb**() {  **super**(Type.ROCK, **60**, **0.95**);  }    **protected** **void** **applyOppEffects**(Pokemon p) {  p.setMod(Stat.SPEED, -**1**);  }  } |

**Clauncher.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  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54 | **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Clauncher** **extends** Pokemon {  **public** **Clauncher**(String name, **int** lvl) {  **super**(name, lvl);  setStats(**50**, **53**, **62**, **58**, **63**, **44**);  setType(Type.WATER);  setMove(**new** Scald(), **new** DoubleTeam(), **new** Rest());  }  }  **class** **Scald** **extends** SpecialMove {  **public** **Scald**() {  **super**(Type.WATER, **80**, **1.0**);  }  **protected** **void** **applyOppEffects**(Pokemon p) {  **if** (Math.random() < **0.3**) {  Effect.burn(p);  }  }  }  **class** **DoubleTeam** **extends** StatusMove {  **public** **DoubleTeam**() {  **super**(Type.NORMAL, **0**, **1.0**);  }  **@Override**  **protected** **boolean** **checkAccuracy**(Pokemon att, Pokemon def) {  **return** **true**;  }    **protected** **void** **applySelfEffects**(Pokemon p) {  p.setMod(Stat.EVASION, **1**);  }  }  **class** **Rest** **extends** StatusMove {  **public** **Rest**() {  **super**(Type.PSYCHIC, **0**, **1.0**);  }  **@Override**  **protected** **boolean** **checkAccuracy**(Pokemon att, Pokemon def) {  **return** **true**;  }    **protected** **void** **applySelfEffects**(Pokemon p) {  p.setMod(Stat.HP, (**int**) (p.getHP() - p.getStat(Stat.HP)));  Effect e = **new** Effect().turns(**2**).condition(Status.SLEEP);  p.setCondition(e);  }  } |

**Clawitzer.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 | **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Clawitzer** **extends** Clauncher {  **public** **Clawitzer**(String name, **int** lvl) {  **super**(name, lvl);  setStats(**71**, **73**, **88**, **120**, **89**, **59**);  setType(Type.WATER);  setMove(**new** Scald(), **new** DoubleTeam(), **new** Rest(), **new** AncientPower());  }  }  **class** **AncientPower** **extends** SpecialMove {  **public** **AncientPower**() {  **super**(Type.ROCK, **60**, **1.0**);  }  **protected** **void** **applySelfEffects**(Pokemon p) {  **if** (Math.random() < **0.1**) {  p.setMod(Stat.ATTACK, **1**);  p.setMod(Stat.DEFENSE, **1**);  p.setMod(Stat.SPECIAL\_ATTACK, **1**);  p.setMod(Stat.SPECIAL\_DEFENSE, **1**);  p.setMod(Stat.SPEED, **1**);  }  }  } |

**Cyndaquil.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  30  31  32  33  34 | **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Cyndaquil** **extends** Pokemon {  **public** **Cyndaquil**(String name, **int** lvl) {  **super**(name, lvl);  setStats(**39**, **52**, **43**, **60**, **50**, **65**);  setType(Type.FIRE);  setMove(**new** FireBlast(), **new** PoisonJab());  }  }  **class** **FireBlast** **extends** SpecialMove {  **public** **FireBlast**() {  **super**(Type.FIRE, **110**, **0.85**);  }  **protected** **void** **applyOppEffects**(Pokemon p) {  **if** (Math.random() < **0.1**) {  Effect.burn(p);  }  }  }  **class** **PoisonJab** **extends** PhysicalMove {  **public** **PoisonJab**() {  **super**(Type.POISON, **80**, **1.0**);  }    **protected** **void** **applyOppEffects**(Pokemon p) {  **if** (Math.random() < **0.3**) {  Effect.poison(p);  }  }  } |

**Quilava.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Quilava** **extends** Cyndaquil {  **public** **Quilava**(String name, **int** lvl) {  **super**(name, lvl);  setStats(**58**, **64**, **58**, **80**, **65**, **80**);  setType(Type.FIRE);  setMove(**new** FireBlast(), **new** PoisonJab(), **new** VitalThrow());  }  }  **class** **VitalThrow** **extends** PhysicalMove {  **public** **VitalThrow**() {  **super**(Type.FIGHTING, **70**, **1.0**, -**1**, **1**);  }    **@Override**  **protected** **boolean** **checkAccuracy**(Pokemon att, Pokemon def) {  **return** **true**;  }  } |

**Typhlosion.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | **import** **ru.ifmo.se.pokemon.\***;  **public** **class** **Typhlosion** **extends** Quilava {  **public** **Typhlosion**(String name, **int** lvl) {  **super**(name, lvl);  setStats(**78**, **84**, **78**, **109**, **85**, **100**);  setType(Type.FIRE);  setMove(**new** FireBlast(), **new** PoisonJab(), **new** VitalThrow(), **new** DoubleTeam());  }  } |

**Battleground.java**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | **import** **ru.ifmo.se.pokemon.\***;  **import** **static** java.lang.Math.\*;  **public** **class** **Battleground** {  **public** **static** **void** **main**(String[] args) {  Battle battle = **new** Battle();  battle.addAlly(**new** Togedemaru("Гурин", **13**));  battle.addAlly(**new** Clauncher("Шайхотаров", **17**));  battle.addAlly(**new** Clawitzer("Анищенко", **15**));  battle.addFoe(**new** Cyndaquil("Письмак", **15**));  battle.addFoe(**new** Quilava("Гаврилов", **17**));  battle.addFoe(**new** Typhlosion("Перминов", **13**));  battle.go();  }  } |

**Output (все совпадения абсолютно случайны)**

Togedemaru Гурин из команды белых вступает в бой!

Cyndaquil Письмак из команды фиолетовых вступает в бой!

Togedemaru Гурин атакует.

Cyndaquil Письмак уменьшает атаку.

Cyndaquil Письмак атакует.

Togedemaru Гурин теряет 40 здоровья.

Togedemaru Гурин атакует.

Cyndaquil Письмак теряет 8 здоровья.

Togedemaru Гурин промахивается

Cyndaquil Письмак атакует.

Togedemaru Гурин не замечает воздействие типа POISON

Togedemaru Гурин атакует.

Cyndaquil Письмак теряет 14 здоровья.

Cyndaquil Письмак уменьшает скорость.

Cyndaquil Письмак атакует.

Критический удар!

Togedemaru Гурин теряет 50 здоровья.

Togedemaru Гурин теряет сознание.

Clauncher Шайхотаров из команды белых вступает в бой!

Cyndaquil Письмак атакует.

Clauncher Шайхотаров теряет 6 здоровья.

Clauncher Шайхотаров атакует.

Cyndaquil Письмак теряет 28 здоровья.

Cyndaquil Письмак теряет сознание.

Quilava Гаврилов из команды фиолетовых вступает в бой!

Quilava Гаврилов атакует.

Clauncher Шайхотаров теряет 13 здоровья.

Clauncher Шайхотаров атакует.

Clauncher Шайхотаров восстанавливает 19 здоровья.

Clauncher Шайхотаров засыпает

Clauncher Шайхотаров атакует.

Quilava Гаврилов теряет 20 здоровья.

Quilava Гаврилов атакует.

Clauncher Шайхотаров теряет 9 здоровья.

Clauncher Шайхотаров атакует.

Clauncher Шайхотаров увеличивает уклоняемость.

Quilava Гаврилов атакует.

Clauncher Шайхотаров теряет 10 здоровья.

Quilava Гаврилов промахивается

Clauncher Шайхотаров атакует.

Clauncher Шайхотаров увеличивает уклоняемость.

Quilava Гаврилов атакует.

Clauncher Шайхотаров теряет 10 здоровья.

Clauncher Шайхотаров атакует.

Clauncher Шайхотаров увеличивает уклоняемость.

Quilava Гаврилов атакует.

Clauncher Шайхотаров теряет 8 здоровья.

Clauncher Шайхотаров атакует.

Clauncher Шайхотаров восстанавливает 37 здоровья.

Clauncher Шайхотаров засыпает

Clauncher Шайхотаров атакует.

Quilava Гаврилов теряет 23 здоровья.

Quilava Гаврилов атакует.

Clauncher Шайхотаров теряет 12 здоровья.

Clauncher Шайхотаров атакует.

Quilava Гаврилов теряет 26 здоровья.

Quilava Гаврилов теряет сознание.

Typhlosion Перминов из команды фиолетовых вступает в бой!

Typhlosion Перминов атакует.

Критический удар!

Clauncher Шайхотаров теряет 15 здоровья.

Clauncher Шайхотаров атакует.

Clauncher Шайхотаров восстанавливает 27 здоровья.

Clauncher Шайхотаров засыпает

Clauncher Шайхотаров атакует.

Typhlosion Перминов теряет 37 здоровья.

Typhlosion Перминов атакует.

Clauncher Шайхотаров теряет 5 здоровья.

Typhlosion Перминов атакует.

Clauncher Шайхотаров теряет 7 здоровья.

Clauncher Шайхотаров атакует.

Typhlosion Перминов теряет 37 здоровья.

Typhlosion Перминов теряет сознание.

В команде фиолетовых не осталось покемонов.

Команда белых побеждает в этом бою!

**Выводы:** в процессе выполнения лабораторной работы были получены навыки использования объектно-ориентированного подхода программирования. Также я “почти понял” смысл игры Pokemon