

Университет ИТМО

Программирование

Отчет по лабораторной работе №2

ФИО студента: Готовко Алексей Владимирович

Номер варианта: 311958

Направление подготовки: 09.03.04 (СППО)

Учебная группа: Р3119

ФИО преподавателя: Письмак Алексей Евгеньевич

Санкт-Петербург, 2021 г.

1. 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 (`Pokemon.jar`).

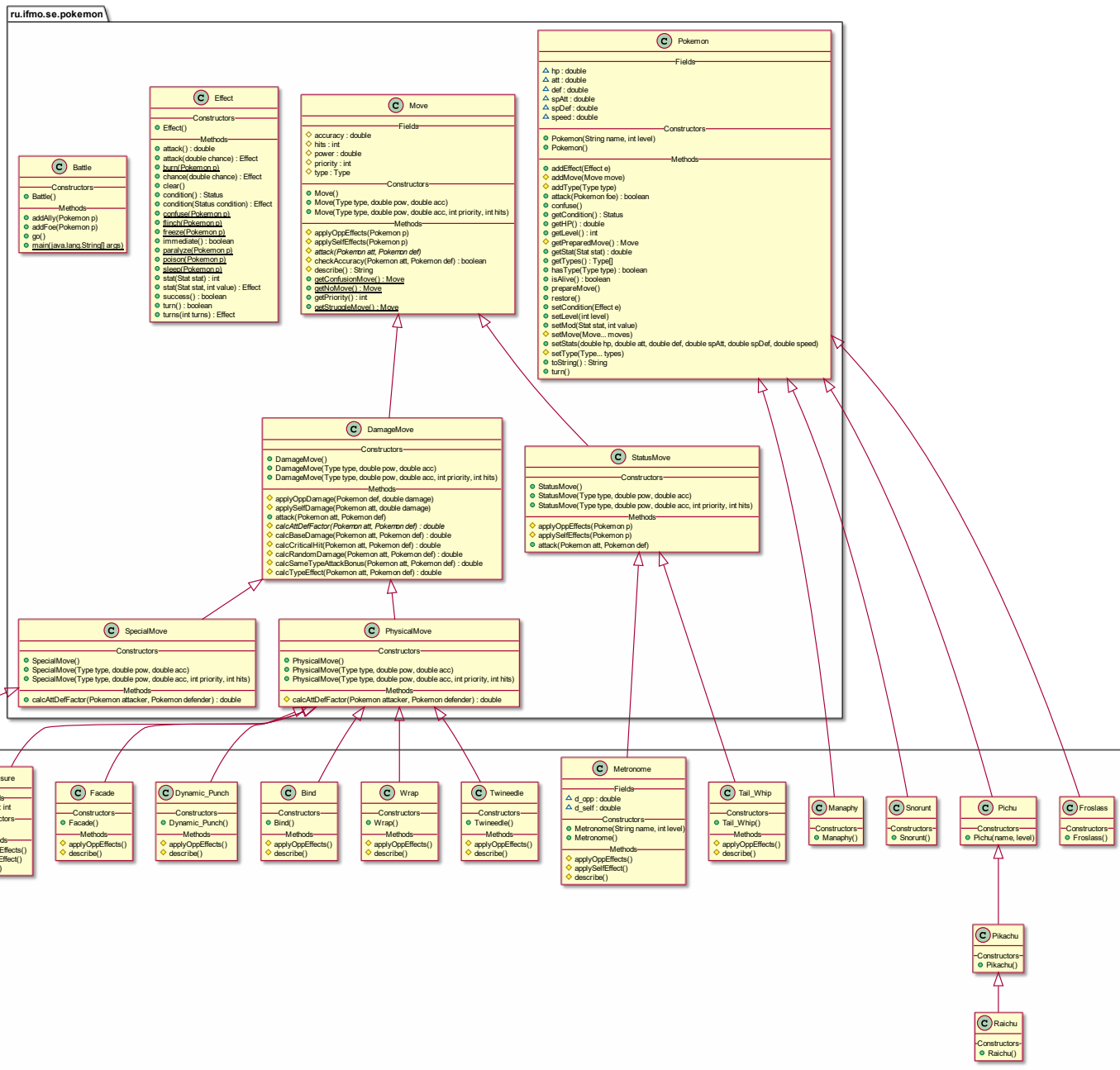
Что надо сделать (краткое описание):

- Ознакомиться с документацией, обращая особое внимание на классы `Pokemon` и `Move`. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
- 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.
- Написать минимально работающую программу и посмотреть, как она работает.

- 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.
- 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.
- 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.
- Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

2. Диаграмма классов

Lab. 2 | Pokemon class diagram



3. Исходный код программы

```
1  import ru.ifmo.se.pokemon.*;
2
3  public class Main {
4      public static void main(String[] args) {
5          Battle battle = new Battle();
6          Pokemon manaphy = new Manaphy("Boris 'The Blade' Yurinov", 1);
7          Pokemon snorunt = new Snorunt("Turkish", 1);
8          Pokemon froslass = new Froslass("Franky Four-Fingers", 1);
9          Pokemon pichu = new Pichu("Sol", 1);
10         Pokemon pikachu = new Pikachu("Bullet Tooth Tony", 1);
11         Pokemon raichu = new Raichu("Gorgeous George", 1);
12         battle.addAlly(pichu);
13         battle.addAlly(pikachu);
14         battle.addAlly(raichu);
15         battle.addFoe(manaphy);
16         battle.addFoe(snorunt);
17         battle.addFoe(froslass);
18         battle.go();
19     }
20 }

1  import ru.ifmo.se.pokemon.*;
2
3  public class Manaphy extends Pokemon {
4      public Manaphy(String name, int level) {
5          super(name, level);
6          setStats(90.4, 75.0, 82.7, 80.1, 84.4, 84.3);
7          setType(Type.WATER);
8          setMove(new Bind(), new Metronome(), new Tail_Whip(), new Fire_Blast());
9      }
10 }

1  import ru.ifmo.se.pokemon.*;
2
3  public class Froslass extends Pokemon {
4      public Froslass(String name, int level) {
5          super(name, level);
6          setStats(56.6, 54.5, 50.5, 62.6, 50.6, 91.0);
7          setType(Type.ICE, Type.GHOST);
8          setMove(new Metronome(), new Fire_Blast(), new Facade(), new Wrap());
9      }
10 }

1  import ru.ifmo.se.pokemon.*;
2
3  public class Snorunt extends Pokemon {
4      public Snorunt(String name, int level) {
5          super(name, level);
6          setStats(22.7, 17.1, 23.1, 27.2, 22.7, 29.7);
7          setType(Type.ICE);
8          setMove(new Metronome(), new Fire_Blast(), new Facade());
9      }
10 }
```

```
1  import ru.ifmo.se.pokemon.*;
2
3  public class Pichu extends Pokemon {
4      public Pichu(String name, int level) {
5          super(name, level);
6          setStats(0.6, 8.6, 0.5, 9.2, 6.2, 41.3);
7          setType(Type.ELECTRIC);
8          setMove(new Metronome(), new Fissure());
9      }
10 }
```

```
1  import ru.ifmo.se.pokemon.*;
2
3  public class Pikachu extends Pichu {
4      public Pikachu(String name, int level) {
5          super(name, level);
6          setStats(4.5, 23.5, 10.6, 27.2, 22.7, 73.2);
7          addMove(new Twineedle());
8      }
9  }
```

```
1  import ru.ifmo.se.pokemon.*;
2
3  public class Raichu extends Pikachu {
4      public Raichu(String name, int level) {
5          super(name, level);
6          setStats(38.5, 65.1, 29.3, 70.6, 64.0, 91.0);
7          addMove(new Dynamic_Punch());
8      }
9  }
```

```

1  import ru.ifmo.se.pokemon.*;
2
3  class Bind extends PhysicalMove {
4      public Bind() {
5          accuracy = 0.85;
6          power = 15;
7          priority = 0;
8          type = Type.NORMAL;
9
10         double r = Math.random() * 100 / 8;
11         if (r < 37.5) hits = 2;
12         else if (r < 75) hits = 3;
13         else if (r < 87.5) hits = 4;
14         else hits = 5;
15     }
16
17     protected void applyOppEffects(Pokemon opp) {
18         double T = Math.random();
19         int t = 0;
20         if (T < 0.25) t = 2;
21         else if (T < 0.5) t = 3;
22         else if (T < 0.75) t = 4;
23         else t = 5;
24         opp.addEffect(new Effect().turns(t).stat(Stat.HP, (int) -(opp.getHP() / 16.0)));
25     }
26
27     protected String describe() {
28         return "uses bind! Enemy's going to lose 1/16 of his HP for the next several turns!";
29     }
30 }
31
32 class Tail_Whip extends StatusMove {
33     public Tail_Whip() {
34         accuracy = 1;
35         hits = 1;
36         power = 0;
37         priority = 0;
38         type = Type.NORMAL;
39     }
40
41     protected void applyOppEffects(Pokemon opp) {
42         opp.addEffect(new Effect().stat(Stat.DEFENSE, -1));
43     }
44
45     protected String describe() {
46         return "uses tail whip! Opponent's thick skin have just become a bit thinner!";
47     }
48 }
49
50 class Fire_Blast extends SpecialMove {
51     public Fire_Blast() {
52         accuracy = 0.85;
53         hits = 1;
54         power = 110;
55         priority = 0;
56         type = Type.FIRE;
57     }
58
59     protected void applyOppEffects(Pokemon opp) {
60         opp.addEffect(new Effect().chance(0.1).condition(Status.BURN));
61     }
62
63     protected String describe() {
64         return "uses fire blast! Smells like shashlik. Mmm, delicious!";
65     }
66 }
67
68 class Facade extends PhysicalMove {
69     public Facade() {

```

```

70         accuracy = 1;
71         hits = 1;
72         power = 70;
73         priority = 0;
74         type = Type.NORMAL;
75     }
76
77     protected void applyOppEffects(Pokemon opp) {
78         Status currentStatus = opp.getCondition();
79         if (currentStatus == Status.BURN || currentStatus == Status.PARALYZE ||
            currentStatus == Status.POISON) hits++;
80     }
81
82     protected String describe() {
83         return "uses facade! What a slaughter!";
84     }
85 }
86
87 class Wrap extends PhysicalMove {
88     public Wrap() {
89         accuracy = 0.9;
90         power = 15;
91         priority = 0;
92         type = Type.NORMAL;
93
94         double r = Math.random() * 100 / 8;
95         if (r < 37.5) hits = 2;
96         else if (r < 75) hits = 3;
97         else if (r < 87.5) hits = 4;
98         else hits = 5;
99     }
100
101     protected void applyOppEffects(Pokemon opp) {
102         double T = Math.random();
103         int t = 0;
104         if (T < 0.25) t = 2;
105         else if (T < 0.5) t = 3;
106         else if (T < 0.75) t = 4;
107         else t = 5;
108         opp.addEffect(new Effect().turns(t).stat(Stat.HP, (int) -(opp.getHP() / 16.0)));
109     }
110
111     protected String describe() {
112         return "uses wrap! So now opponent is kinda... Wrapped? Like a christmas present?..";
113     }
114 }
115
116 class Fissure extends PhysicalMove {
117     public Fissure() {
118         accuracy = 0.3;
119         hits = 1;
120         power = 0;
121         priority = 0;
122         type = Type.GROUND;
123     }
124
125     int selfLevel;
126     protected void applySelfEffect(Pokemon self) {
127         selfLevel = self.getLevel();
128     }
129
130     protected void applyOppEffects(Pokemon opp) {
131         int oppLevel = opp.getLevel();
132         if (selfLevel >= oppLevel) {
133             opp.addEffect(new Effect().stat(Stat.HP, (int) -(opp.getHP())));
134         }
135     }
136
137     protected String describe() {

```



```

138         return "uses fissure! Oh boy, that's painful!";
139     }
140 }
141
142 class Twineedle extends PhysicalMove {
143     public Twineedle() {
144         accuracy = 1;
145         hits = 2;
146         power = 25;
147         priority = 0;
148         type = Type.BUG;
149     }
150
151     protected void applyOppEffects(Pokemon opp) {
152         opp.addEffect(new Effect().chance(0.2).condition(Status.POISON));
153     }
154
155     protected String describe() {
156         return "uses twineedle! Powerful bug bite! How he managed to do that if he isn't
            even a bug?..";
157     }
158 }
159
160 class Dynamic_Punch extends PhysicalMove {
161     public Dynamic_Punch() {
162         accuracy = 0.5;
163         hits = 1;
164         power = 100;
165         priority = 0;
166         type = Type.FIGHTING;
167     }
168
169     protected void applyOppEffects(Pokemon opp) {
170         opp.confuse();
171     }
172
173     protected String describe() {
174         return "uses dynamic punch! Big-bam-boom! Watch your face!";
175     }
176 }
177
178
179 class Metronome extends StatusMove {
180     public Metronome() {
181         accuracy = 1;
182         hits = 1;
183         power = 0;
184         priority = 0;
185         type = Type.NORMAL;
186     }
187
188     double d_self = Math.random();
189     double d_opp = Math.random();
190
191     protected void applyOppEffects(Pokemon opp) {
192         opp.addEffect(new Effect().stat(Stat.HP, (int) (opp.getHP() * d_opp)));
193     }
194     protected void applySelfEffect(Pokemon self) {
195         self.addEffect(new Effect().stat(Stat.HP, (int) -(self.getHP() * d_self)));
196     }
197
198     protected String describe() {
199         return "uses metronome! Oh, wait.. No way! They are playing russian roulette!";
200     }
201 }
202

```

4. Результат работы программы

Pichu Sol from the team White enters the battle!
Manaphy Boris 'The Blade' Yurinov from the team Red enters the battle!
Manaphy Boris 'The Blade' Yurinov uses fire blast! Smells like shashlik. Mmm, delicious!.
Pichu Sol loses 4 hit points.

Pichu Sol misses

Manaphy Boris 'The Blade' Yurinov uses bind! Enemy's going to lose 1/16 of his HP for the next several turns!.
Pichu Sol loses 3 hit points.
Manaphy Boris 'The Blade' Yurinov uses bind! Enemy's going to lose 1/16 of his HP for the next several turns!.
Pichu Sol loses 3 hit points.

Pichu Sol misses

Manaphy Boris 'The Blade' Yurinov uses fire blast! Smells like shashlik. Mmm, delicious!.
Pichu Sol loses 6 hit points.
Pichu Sol faints.
Pikachu Bullet Tooth Tony from the team White enters the battle!
Manaphy Boris 'The Blade' Yurinov misses

Pikachu Bullet Tooth Tony uses twineedle! Powerful bug bite! How he managed to do that if he isn't even a bug?...
Manaphy Boris 'The Blade' Yurinov loses 3 hit points.
Pikachu Bullet Tooth Tony uses twineedle! Powerful bug bite! How he managed to do that if he isn't even a bug?...
Manaphy Boris 'The Blade' Yurinov loses 4 hit points.

Manaphy Boris 'The Blade' Yurinov uses bind! Enemy's going to lose 1/16 of his HP for the next several turns!.
Pikachu Bullet Tooth Tony loses 3 hit points.
Manaphy Boris 'The Blade' Yurinov uses bind! Enemy's going to lose 1/16 of his HP for the next several turns!.
Pikachu Bullet Tooth Tony loses 4 hit points.

Pikachu Bullet Tooth Tony uses twineedle! Powerful bug bite! How he managed to do that if he isn't even a bug?...
Manaphy Boris 'The Blade' Yurinov loses 3 hit points.
Pikachu Bullet Tooth Tony uses twineedle! Powerful bug bite! How he managed to do that if he isn't even a bug?...
Manaphy Boris 'The Blade' Yurinov loses 4 hit points.
Manaphy Boris 'The Blade' Yurinov faints.
Snorunt Turkish from the team Red enters the battle!
Pikachu Bullet Tooth Tony uses twineedle! Powerful bug bite! How he managed to do that if he isn't even a bug?...
Snorunt Turkish loses 4 hit points.
Pikachu Bullet Tooth Tony uses twineedle! Powerful bug bite! How he managed to do that if he isn't even a bug?...
Snorunt Turkish loses 4 hit points.

Snorunt Turkish uses metronome! Oh, wait.. No way! They are playing russian roulette!.

Pikachu Bullet Tooth Tony loses 2 hit points.
Pikachu Bullet Tooth Tony misses

Snorunt Turkish uses metronome! Oh, wait.. No way! They are playing russian roulette!.

Pikachu Bullet Tooth Tony loses 2 hit points.
Pikachu Bullet Tooth Tony loses 1 hit points.
Raichu Gorgeous George from the team White enters the battle!
Raichu Gorgeous George misses

Snorunt Turkish uses facade! What a slaughter!.
Raichu Gorgeous George loses 6 hit points.

Raichu Gorgeous George uses dynamic punch! Big-bam-boom! Watch your face!.
Snorunt Turkish loses 15 hit points.
Snorunt Turkish faints.
Froslass Franky Four-Fingers from the team Red enters the battle!
Froslass Franky Four-Fingers uses fire blast! Smells like shashlik. Mmm, delicious!.
Raichu Gorgeous George loses 7 hit points.
Raichu Gorgeous George faints.
Team White loses its last Pokemon.
The team Red wins the battle!

5. Вывод

В процессе выполнения лабораторной работы удалось познакомиться с понятиями класса и объекта, модификаторами доступа, инструкцией `import`; научиться писать код в парадигме ООП, создавать диаграммы классов объектной модели.