Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

**Отчет**

по лабораторной работе №2

по программированию

**Вариант: 94782**

Выполнил:

Тахватулин Михаил Витальевич

Группа: P3107

Преподаватель:

Гаврилов Антон Валерьевич

г. Санкт-Петербург

г. 2022

**Оглавление**

[Задание 2](#_Toc118260148)

[Комментарии 3](#_Toc118260149)

[Покемоны 5](#_Toc118260150)

[Выполнение задания 6](#_Toc118260151)

[Диаграмма классов реализованной объектной модели 8](#_Toc118260152)

[Результат работы программы 9](#_Toc118260153)

# **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.

Базовые классы и симулятор сражения находятся в [jar-архиве](https://se.ifmo.ru/documents/10180/660917/Pokemon.jar/a7ce60af-6ee6-47d0-a95e-e5ed9a697bd2) (обновлен 9.10.2018, исправлен баг с добавлением атак и кодировкой). Документация в формате javadoc - [здесь](https://se.ifmo.ru/~tony/doc/).

Информацию о покемонах, цепочках эволюции и атаках можно найти на сайтах <http://poke-universe.ru>, <http://pokemondb.net>, [http://veekun.com/dex/pokemon](%20http://veekun.com/dex/pokemon)

# **Комментарии**

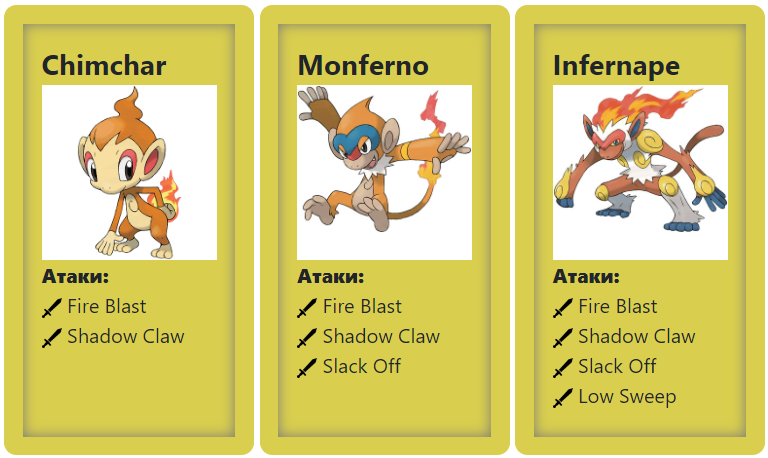
Цель работы: на простом примере разобраться с основными концепциями ООП и научиться использовать их в программах.

Что надо сделать (краткое описание)

1. Ознакомиться с [документацией](https://se.ifmo.ru/~tony/doc/), обращая особое внимание на классы Pokemon и Move. При дальнейшем выполнении лабораторной работы читать документацию еще несколько раз.
2. 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.
3. Написать минимально работающую программу и посмотреть как она работает.
4. Battle b = new Battle();
5. Pokemon p1 = new Pokemon("Чужой", 1);
6. Pokemon p2 = new Pokemon("Хищник", 1);
7. b.addAlly(p1);
8. b.addFoe(p2);
9. b.go();
10. 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.
11. 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.
12. 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.
13. Доделать все необходимые атаки и всех покемонов, распределить покемонов по командам, запустить сражение.

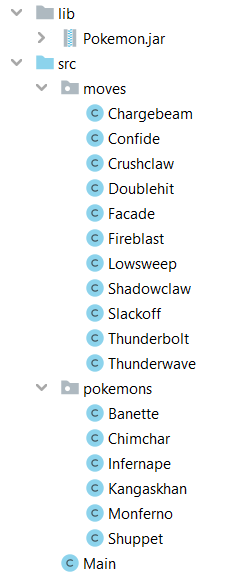
# **Покемоны**





# **Выполнение задания**

1. Создаем ветку



1. Изображение выглядит как текст

   Автоматически созданное описаниеОзнакомившись с документацией и покемонами, начинаем писать код.

Пример заполнения действия покемона

Изображение выглядит как текст

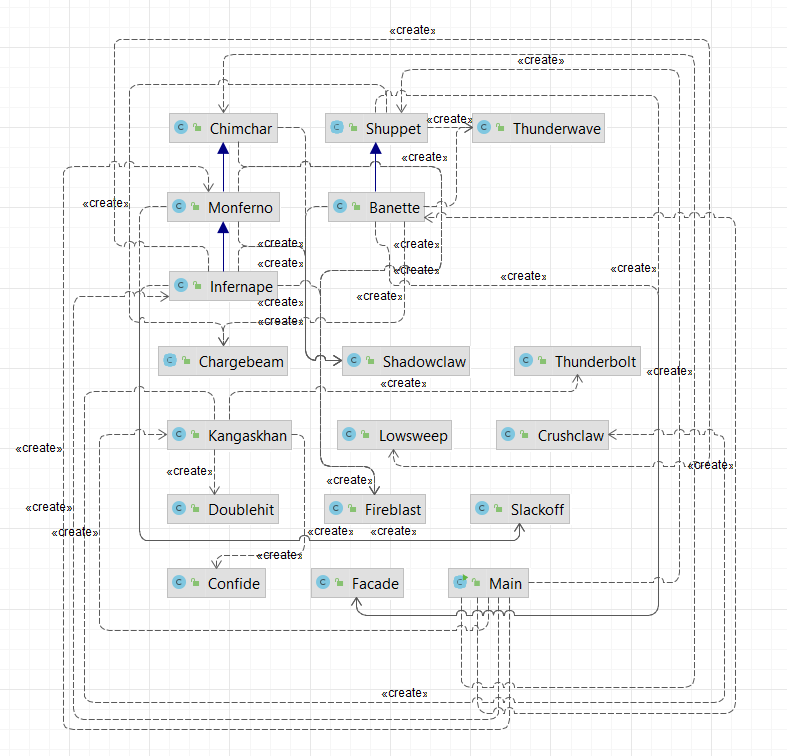
Автоматически созданное описаниеИзображение выглядит как текст

Автоматически созданное описание

Реализация битвы покемонов

Пример заполнения покемона

# **Диаграмма классов реализованной объектной модели**



# **Результат работы программы**

Banette banette1 из команды желтых вступает в бой!

Shuppet shuppet1 из команды зеленых вступает в бой!

Shuppet shuppet1 facadeeeeee.

Banette banette1 теряет 1 здоровья.

Banette banette1 не замечает воздействие типа NORMAL

Banette banette1 facadeeeeee.

Shuppet shuppet1 теряет 1 здоровья.

Shuppet shuppet1 не замечает воздействие типа NORMAL

Shuppet shuppet1 промахивается

Banette banette1 shadow claw.

Shuppet shuppet1 теряет 10 здоровья.

Shuppet shuppet1 facadeeeeee.

Banette banette1 теряет 1 здоровья.

Banette banette1 не замечает воздействие типа NORMAL

Banette banette1 facadeeeeee.

Shuppet shuppet1 теряет 1 здоровья.

Shuppet shuppet1 не замечает воздействие типа NORMAL

Shuppet shuppet1 промахивается

Banette banette1 бзбззбзззззбзбззббззззз.

Shuppet shuppet1 теряет 3 здоровья.

Shuppet shuppet1 теряет сознание.

Kangaskhan kangaskhan1 из команды зеленых вступает в бой!

Kangaskhan kangaskhan1 thunderbolt.

Banette banette1 теряет 6 здоровья.

Banette banette1 thunderwave.

Kangaskhan kangaskhan1 парализован

Kangaskhan kangaskhan1 промахивается

Banette banette1 shadow claw.

Kangaskhan kangaskhan1 теряет 1 здоровья.

Kangaskhan kangaskhan1 не замечает воздействие типа GHOST

Kangaskhan kangaskhan1 дабл боньк.

Banette banette1 теряет 1 здоровья.

Banette banette1 не замечает воздействие типа NORMAL

Kangaskhan kangaskhan1 дабл боньк.

Критический удар!

Banette banette1 теряет 1 здоровья.

Banette banette1 не замечает воздействие типа NORMAL

Banette banette1 thunderwave.

Kangaskhan kangaskhan1 thunderbolt.

Banette banette1 теряет 8 здоровья.

Banette banette1 теряет сознание.

Infernape infernape1 из команды желтых вступает в бой!

Kangaskhan kangaskhan1 промахивается

Infernape infernape1 промахивается

Kangaskhan kangaskhan1 thunderbolt.

Infernape infernape1 теряет 8 здоровья.

Infernape infernape1 low sweep.

Kangaskhan kangaskhan1 теряет 15 здоровья.

Kangaskhan kangaskhan1 боньк с ослаблением.

Infernape infernape1 теряет 11 здоровья.

Infernape infernape1 теряет сознание.

Chimchar chimchar1 из команды желтых вступает в бой!

Kangaskhan kangaskhan1 дабл боньк.

Chimchar chimchar1 теряет 6 здоровья.

Kangaskhan kangaskhan1 дабл боньк.

Chimchar chimchar1 теряет 5 здоровья.

Chimchar chimchar1 shadow claw.

Kangaskhan kangaskhan1 теряет 1 здоровья.

Kangaskhan kangaskhan1 не замечает воздействие типа GHOST

Kangaskhan kangaskhan1 thunderbolt.

Chimchar chimchar1 теряет 7 здоровья.

Chimchar chimchar1 shadow claw.

Kangaskhan kangaskhan1 теряет 1 здоровья.

Kangaskhan kangaskhan1 не замечает воздействие типа GHOST

Kangaskhan kangaskhan1 дабл боньк.

Chimchar chimchar1 теряет 5 здоровья.

Kangaskhan kangaskhan1 дабл боньк.

Chimchar chimchar1 теряет 8 здоровья.

Chimchar chimchar1 теряет сознание.

В команде желтых не осталось покемонов.

Команда зеленых побеждает в этом бою!

# **Создание Jar-файла**

Есть два способа, чтобы собрать проект в Jar.

Первый способ (при помощи IDE):

В нашем случае будем пользоваться программой IntelliJ IDEA. Первым делом нам нужно открыть окно структуры проекта и перейти в раздел Artifacts.

Изображение выглядит как текст

Автоматически созданное описание

Там нажимаем на +, и создаем JAR.

Изображение выглядит как текст

Автоматически созданное описание

В появившемся окне выбираем главный класс и путь к манифесту.

Изображение выглядит как текст

Автоматически созданное описание

Выбираем название файла и ставим галочку у пункта Include in project build.

Изображение выглядит как текст

Автоматически созданное описание

Все, сохраняем и запускаем.

Второй способ:

Загрузим папку проекта на Хелиос. Затем, используя команду

*javac -d out -cp lib/Pokemon.jar -sourcepath src src/Main.java*

компилируем проект в байт-код. После успешной компиляции создадим файл манифеста в папке, которую указали после ключа -d. Заполним манифест, не забывая про пустую последнюю строчку:

Изображение выглядит как текст

Автоматически созданное описание

Создадим JAR-файл командой

*jar cmf MANIFEST.MF project.jar .*

И запустим командой

*java -jar project.jar*

# **Вывод**

В процессе данной лабораторной я изучил принципы работы ООП, научился применять объектно-ориентированное программирование в своих программах. Также написал программу, построенную на использовании разных классов и объектов.