

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Лабораторная работа №3
по дисциплине
Программирование
Вариант: 30324

Выполнил:
Герасюто Фадей Александрович
Группа: Р3119
Преподаватель:
Харитонова Анастасия Евгеньевна

Санкт-Петербург, 2025

Содержание

Задание	3
Диаграмма классов реализованной объектной модели	4
Исходный код программы	5
Результат работы программы	6
Промпты для ИИ-ассистентов, примеры полученного кода, сравнительный анализ	7
Вывод	9

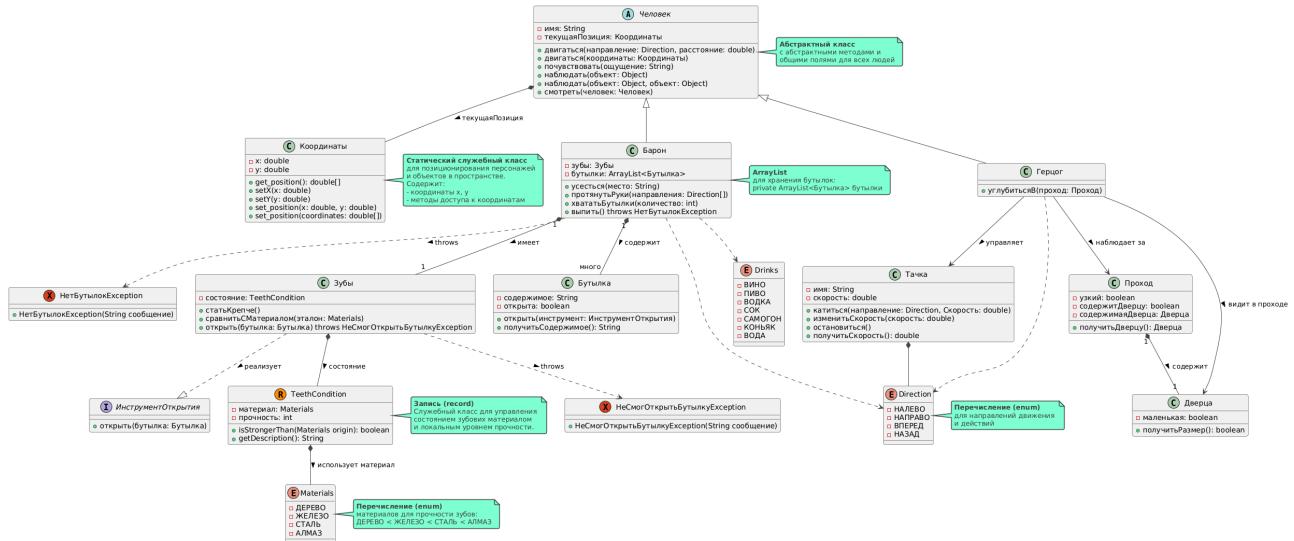
Задание

В соответствии с выданным вариантом на основе предложенного текстового отрывка из литературного произведения создать объектную модель реального или воображаемого мира, описываемого данным текстом. Должны быть выделены основные персонажи и предметы со свойственным им состоянием и поведением. На основе модели написать программу на языке Java. Введите вариант:

Описание предметной области, по которой должна быть построена объектная модель:

В конце концов герцог почувствовал, что тачка катится все медленнее и можно наконец остановиться. Как раз в этом самом месте, в левом ряду бочек, он увидел узкий проход, а в глубине прохода – маленькую дверцу. Барон, удобно усевшись на земле, протягивал руки то направо, то налево и, не тратя даром ни минуты, хватал по две, по три бутылки, вытаскивал пробки зубами, которые у него давно уже стали крепче железа от постоянного упражнения, и опрокидывал содержимое бутылок себе в рот. Он прерывал это занятие только для того, чтобы испустить вздох удовлетворения. Герцог долго глядел на него, а потом махнул рукой и углубился в проход.

Диаграмма классов реализованной объектной модели



Исходный код программы

GitHub

Собрать .jar файл

```
javac -d . lib/Utils/*.java lib/Human/*.java lib/Objects/*.java Main.java  
jar -c -f Main.jar -e Main Main.class lib/Objects/*.class lib/Human/*.class lib/Utils/*.class
```

Результат работы программы

```
$ java -jar Main.jar
```

Герцог почувствовал, что Тачка катится медленнее и медленнее

Герцог почувствовал, что можно наконец остановиться

Герцог заметил объект узкий проход

Герцог заметил объект Маленькая дверь в узкий проход

Барон удобно сел на земле

Барон протягивает руки LEFT RIGHT

Барон схватил 2 бутылки(ок) с напитками

Зубы укрепились! Теперь они крепче - DIAMOND (уровень прочности: 1)

Бутылка с COGNAC успешно открыта!

Барон опрокинул бутылку с COGNAC себе в рот

Зубы стали крепче - DIAMOND (уровень прочности: 2)

Бутылка с COGNAC успешно открыта!

Барон опрокинул бутылку с COGNAC себе в рот

Барон схватил 3 бутылки(ок) с напитками

Бутылка с VODKA успешно открыта!

Барон опрокинул бутылку с VODKA себе в рот

Бутылка с WINE успешно открыта!

Барон опрокинул бутылку с WINE себе в рот

Бутылка с WINE успешно открыта!

Барон опрокинул бутылку с WINE себе в рот

Барон схватил 2 бутылки(ок) с напитками

Зубы стали крепче - DIAMOND (уровень прочности: 3)

Бутылка с WINE успешно открыта!

Барон опрокинул бутылку с WINE себе в рот

Зубы стали крепче - DIAMOND (уровень прочности: 4)

Бутылка с WINE успешно открыта!

Барон опрокинул бутылку с WINE себе в рот

Барон схватил 3 бутылки(ок) с напитками

Зубы стали крепче - DIAMOND (уровень прочности: 5)

Бутылка с WINE успешно открыта!

Барон опрокинул бутылку с WINE себе в рот

Бутылка с COGNAC успешно открыта!

Барон опрокинул бутылку с COGNAC себе в рот

Зубы стали крепче - DIAMOND (уровень прочности: 6)

Бутылка с JUICE успешно открыта!

Барон опрокинул бутылку с JUICE себе в рот

Барон схватил 2 бутылки(ок) с напитками

Бутылка с COGNAC успешно открыта!

Барон опрокинул бутылку с COGNAC себе в рот

Зубы стали крепче - DIAMOND (уровень прочности: 7)

Бутылка с COGNAC успешно открыта!

Барон опрокинул бутылку с COGNAC себе в рот

Барон: There are no bottles left

Герцог смотрит на Барона

Герцог углубился в узкий проход

Промпты для ИИ-ассистентов, примеры полученного кода, сравнительный анализ

image.png
PNG

Реализуй метод выпить класса барон по схеме

Ответ 🔊

Для реализации метода `выпить` класса `Барон` по предоставленной схеме, необходимо учесть следующие аспекты:

1. Класс `Барон` наследует от абстрактного класса `Человек`.
2. Метод `выпить` должен быть реализован в классе `Барон`.
3. Метод `выпить` должен вызывать метод `выпить` из класса `Бутылка`.
4. Метод `выпить` из класса `Бутылка` должен изменять состояние бутылки и возвращать результат.

Вот пример реализации метода `выпить` в классе `Барон`:

```
java Копировать
public class Барон extends Человек {
    private Бутылка бутылка;

    public Барон(Бутылка бутылка) {
        this.бутылка = бутылка;
    }

    public void выпить() {
        try {
            бутылка.выпить();
        } catch (НеоткрытаяБутылкаException e) {
            System.out.println("Бутылка не открыта. Сначала откройте бутылку.");
        }
    }
}
```

Gigachat

Реализуй метод выпить класса барон по схеме

Qwen3-Max 12:33 PM

```
java
public void выпить() throws НетБутылкException {
    if (бутылки.isEmpty()) {
        throw new НетБутылкException("Барон не может выпить: нет бутылок!");
    }

    Бутылка текущаяБутылка = бутылки.get(0);
    String содержимое = текущаяБутылка.получитьСодержимое();

    // Проверяем, можно ли открыть бутылку
    try {
        текущаяБутылка.открыть(new ИнструментОткрытия());
    } catch (НесмогОткрытьБутылкуException e) {
        throw new НетБутылкException("Барон не может выпить: не смог открыть бутылку с " + содержимое + "!");
    }

    // Удаляем бутылку из списка после того, как выпили
    бутылки.remove(0);

    // Логика выпивания (можно добавить вывод в консоль)
    System.out.println("Барон выпил " + содержимое + ".");
}
```

Qwen3-Max

```
public void drink() throws NoBottlesException {
    Bottle bottle = null;
    try {
        if (this.bottles.isEmpty()) {
            throw new NoBottlesException("There are no bottles left");
        }
        bottle = this.bottles.removeFirst();
        if (Math.random() > 0.5) {
            teeth.getStronger();
        }
        teeth.open(bottle);
        System.out.println(this.name+" опрокинул бутылку с "+bottle.getContent()+" себе в рот\n");
    } catch (NoBottlesException e) {
        System.out.println(this.name + ": " + e.getMessage());
    } catch (CannotOpenBottleException e) {
        System.out.println(this.name + ": " + e.getMessage());
        this.bottles.addLast(bottle);
    }
}
```

Мой код: Реализует полную логику с обработкой исключений (NoBottlesException, CannotOpenBottleException), включая возврат бутылки в список при неудаче. Учитывает случайное укрепление зубов.

Qwen3-Max: Корректно следует схеме. Правильно обрабатывает исключение открытия и выбрасывает новое исключение НетБутылокException с информативным сообщением. Удаляет бутылку из списка после успешного открытия. Не реализует логику “выпивания” (вывод в консоль о выпитом содержимом). Использует get(0) + remove(0), что менее эффективно, чем removeFirst(). Не учитывает возможность укрепления зубов.

GigaChat: Правильно интерпретирует схему — метод выпить(). Предлагает корректную структуру класса Барон. Предоставленный код неполный. Отсутствует работа со списком бутылок (бутылки) и логика вывода результата.

Вывод

В ходе выполнения лабораторной работы была реализована модель с использованием принципов объектно-ориентированного программирования: наследования, инкапсуляции, полиморфизма и абстракции. Работа позволила углубить понимание ООП, принципов абстракций, использование интерфейсов.