

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ «ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А. Н. Тихонова

Департамент компьютерной инженерии

Отчет по практической работе №5
«Обработка исключений в Java»

по курсу «Распределенные базы данных и сетевые
вычисления» Вариант №13

Выполнили:

Студенты группы МКС212

Журсунова Найля

Шабалина Анастасия Владимировна

Приняла:

Байбикова Татьяна Николаевна

Москва, 2022 г.

Оглавление

Задание	3
Листинг	4
Изменения в классе PassengerCar	4
Изменения в классе Truck	5
Изменения в классе Main	6
Результаты работы	7
Выводы	8

Задание

В ходе работы необходимо было добавить в разработанные в предыдущих работах классах возможность обработки исключений и подготовить отчет.

Подготовьте отчет. В отчет включите:

- задание;
- разработанную программу или набор программ (укажите в листингах, где ввод-вывод);
- опишите в отчете, какие использовали классы для обработки исключений;
- результаты работы программы;
- выводы.

Листинг

В данном разделе приведены листинги только тех частей кода, которые были добавлены во время выполнения этой практической работы. Остальная часть кода была подробно описана в предыдущем отчете.

Изменения в классе PassengerCar

В классе PassengerCar в ранее реализованные методы writeAutomobileToFile и readAutomobileFromFile, были добавлены блоки try/catch, которые могут генерировать исключение. При этом был использован класс IOException.

```
76 // переопределение метода записи информации об автомобиле в файл
77 @Override
78 public void writeAutomobileToFile(String filePath) {
79     // подготовка строки для записи
80     String info = "Марка машины: " + this.brand + ". Модель: " + this.model + ".
Год выпуска: " + this.release + " г. (" + autoYear() + " лет)." + "\n" + "Вместимость:
" + this.carCapacity + ". Цвет: " + getColor() + ". Тип коробки передач: " + this.gearBox
+ ". Тип привода: " + this.drive + "." + "\n" + "Тип топлива: " + this.fuelType + ".
Емкость топливного бака: " + this.tankCapacity + " л." + "\n" + "Объем двигателя: " +
engine.capacity + " см3. Мощность двигателя: " + engine.enginePower() + " кВт." + "\n"
+ "Длина кузова: " + this.length + " м. Ширина кузова: " + this.width + "м. Тип кузова:
" + this.bodyType + "." + "\n" + carClass() + "\n" + "Пробег: " + getMileage() + " км.";
81     try(FileWriter writer = new FileWriter(filePath, false))
82     {
83         // запись всей строки
84         writer.write(info);
85         writer.flush();
86     }
87     catch(IOException ex){
88         System.out.println(ex.getMessage());
89     }
90 }
```

```

92     // переопределение метода чтения информации о автомобиле из файла
93     @Override
94     public PassengerCar readAutomobileFromFile(String filePath) {
95         try(FileReader reader = new FileReader(filePath))
96         {
97             char[] buf = new char[256];
98             reader.read(buf);
99             String line = String.valueOf(buf);
100             String[] lines = line.split(" ");
101
102             PassengerCar car = new PassengerCar(lines[0],
Integer.parseInt(lines[1]), Integer.parseInt(lines[2]), lines[3], lines[4],
lines[5], Integer.parseInt(lines[6]), lines[7], lines[8],
Double.parseDouble(lines[9]), Double.parseDouble(lines[10]));
103
104             reader.close();
105             return car;
106         }
107         catch(IOException ex){
108             System.out.println(ex.getMessage());
109             return null;
110         }
111     }

```

Изменения в классе Truck

В классе Truck в ранее реализованные методы writeAutomobileToFile и readAutomobileFromFile, были добавлены блоки try/catch, которые могут генерировать исключение. При этом был использован класс IOException.

```

69     // переопределение метода записи информации об автомобиле в файл
70     @Override
71     public void writeAutomobileToFile(String filePath) {
72         // подготовка строки для записи
73         String info = "Марка машины: " + this.brand + ". Модель: " + this.model + ".
Год выпуска: " + this.release + " г. (" + autoYear() + " лет)." + "\n" + "Вместимость:
" + this.carCapacity + ". Цвет: " + getColor() + ". Тип коробки передач: " + this.gearBox
+ ". Тип привода: " + this.drive + ". " + "\n" + "Тип топлива: " + this.fuelType + ".
Емкость топливного бака: " + this.tankCapacity + " л." + "\n" + "Объем двигателя: " +
engine.capacity + " см3. Мощность двигателя: " + engine.enginePower() + " кВт." + "\n"
+ "Количество осей: " + this.numberChassis + ". Грузоподъемность: " + this.loadCapacity
+ "т (" + truckLoadCapacity() + "). " + "\n" + "Пробег: " + getMileage() + " км.";
74         try(FileWriter writer = new FileWriter(filePath, false))
75         {
76             // запись всей строки
77             writer.write(info);
78             writer.flush();
79         }
80         catch(IOException ex){
81             System.out.println(ex.getMessage());
82         }
83     }

```

```

85     // переопределение метода чтения информации о автомобиле из файла
86     @Override
87     public Truck readAutomobileFromFile(String filePath) {
88         try(FileReader reader = new FileReader(filePath))
89         {
90             char[] buf = new char[256];
91             reader.read(buf);
92             String line = String.valueOf(buf);
93             String[] lines = line.split(" ");
94             Truck truck = new Truck(lines[0], Integer.parseInt(lines[1]),
Integer.parseInt(lines[2]), lines[3], lines[4], lines[5],
Integer.parseInt(lines[6]), lines[7], lines[8], Integer.parseInt(lines[9]),
Integer.parseInt(lines[10]));
95
96             reader.close();
97             return truck;
98         }
99         catch(IOException ex){
100             System.out.println(ex.getMessage());
101             return null;
102         }
103     }

```

Изменения в классе Main

В главной функции мы добавили несколько строк для проверки обработки исключений (указали имена несуществующих файлов и т.п.):

```

45     //проверка обработки исключений
46     car1.writeAutomobileToFile("");
47     car1.readAutomobileFromFile("*");
48     truck1.readAutomobileFromFile(null);

```

Результаты работы

Так выглядит результат выполнения программы:

```
nastya@nastya-ThinkPad-T430:~/Зарпузки$ java com/ddbase/main/Main
Автомобиль №1
Марка машины: Toyota. Модель: Corolla. Год выпуска: 2016 г. (6 лет).
Вместимость: 5. Цвет: Серый. Тип коробки передач: Автомат. Тип привода: Передний.
Тип топлива: Бензин. Емкость топливного бака: 50 л.
Объем двигателя: 1500 см3. Мощность двигателя: 20400.0 кВт.
Длина кузова: 4.63 м. Ширина кузова: 1.78м. Тип кузова: Седан.
Класс D - второй средний класс.
Пробег: 10000 км.

Автомобиль №2
Марка машины: BMW. Модель: X6. Год выпуска: 2012 г. (10 лет).
Вместимость: 5. Цвет: Белый. Тип коробки передач: Автомат. Тип привода: Полный.
Тип топлива: Дизель. Емкость топливного бака: 85 л.
Объем двигателя: 1500 см3. Мощность двигателя: 20400.0 кВт.
Длина кузова: 4.87 м. Ширина кузова: 1.69м. Тип кузова: Кроссовер.
Класс F - представительский класс.
Пробег: 12100 км.

Автомобиль №3
Марка машины: MERCEDES-BENZ. Модель: Actros 3336. Год выпуска: 2011 г. (11 лет).
Вместимость: 3. Цвет: Красный. Тип коробки передач: Механика. Тип привода: Задний.
Тип топлива: Дизель. Емкость топливного бака: 400 л.
Объем двигателя: 6000 см3. Мощность двигателя: 76500.0 кВт.
Количество осей: 3. Грузоподъемность: 20т (Особо большая грузоподъемность).
Пробег: 35000 км.

Марка машины: KIA. Модель: RIO. Год выпуска: 2022 г. (0 лет).
Вместимость: 5. Цвет: null. Тип коробки передач: Механика. Тип привода: Передний.
Тип топлива: Бензин. Емкость топливного бака: 60 л.
Объем двигателя: 1500 см3. Мощность двигателя: 20400.0 кВт.
Длина кузова: 4.42 м. Ширина кузова: 1.74м. Тип кузова: Седан.
Класс D - второй средний класс.
Пробег: 0 км.

Марка машины: MAN. Модель: TGL. Год выпуска: 2005 г. (17 лет).
Вместимость: 2. Цвет: null. Тип коробки передач: Механика. Тип привода: Полный.
Тип топлива: Дизель. Емкость топливного бака: 400 л.
Объем двигателя: 6000 см3. Мощность двигателя: 76500.0 кВт.
Количество осей: 4. Грузоподъемность: 20т (Особо большая грузоподъемность).
Пробег: 0 км.
(Нет такого файла или каталога)
* (Нет такого файла или каталога)
Exception in thread "main" java.lang.NullPointerException
    at java.base/java.io.FileInputStream.<init>(FileInputStream.java:149)
    at java.base/java.io.FileInputStream.<init>(FileInputStream.java:112)
    at java.base/java.io.FileReader.<init>(FileReader.java:60)
    at com.ddbase.autotype.Truck.readAutomobileFromFile(Truck.java:88)
    at com.ddbase.main.Main.main(Main.java:48)
nastya@nastya-ThinkPad-T430:~/Зарпузки$
```

Выводы

При выполнении практической работы были изучены различные средства для обработки исключений в Java. После чего, некоторые из данных средств (блоки `try` и `catch`, пакет `java.io` – Class `IOException`) были применены для выполнения задания. Кроме этого, нами были закреплены знания, полученные в ходе выполнения предыдущей работы.