



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

Дисциплина «Программирование на языке Джава»

ОТЧЕТ
ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №9

Выполнил студент группы ИНБО-02-20

Лукияненко Д.В.

Принял

Степанов П.В.

Практическая работа выполнена «__» _____ 2021 г.

«_____»
Отметка о выполнении

«__» _____ 2021 г.

Москва – 2021 г.

СОДЕРЖАНИЕ

Цель работы	3
Задание	3
Репозиторий	4
Выполнение работы	4
Код выполненной работы	5
Вывод.....	9

Цель работы

Цель данной практической работы – Научиться создавать собственные исключения.

Задание

Клиент совершает покупку онлайн. При оформлении заказа у пользователя запрашивается фио и номер ИНН. В программе проверяется, действителен ли номер ИНН для такого клиента. Исключение будет выдано в том случае, если введен недействительный ИНН.

Предлагается модернизировать задачу из предыдущей лабораторной работы (см. методические указания по выполнению лабораторных работ №1-8) – задача сортировки студентов по среднему баллу. Необходимо разработать пользовательский интерфейс для задачи поиска и сортировки (использовать массив интерфейсных ссылок- пример в лекции 5). Дополнить ее поиском студента по фио – в случае отсутствия такого студента необходимо выдавать собственное исключение.

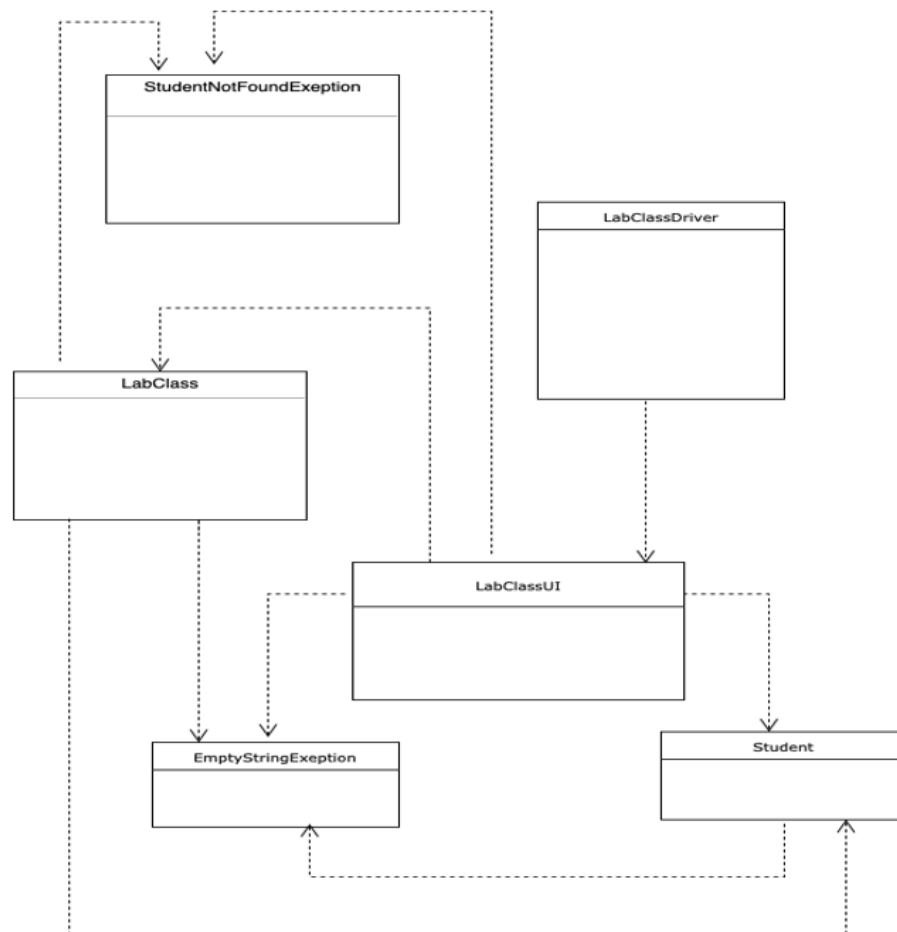


Рисунок 1 – Заданная UML Диаграмма

Репозиторий

Ссылка: https://github.com/neluckoff/mirea-java-lessons/tree/master/src/ru/luckoff/mirea/practice_9

Выполнение работы

Как мы можем понять по диаграмме, данной нам в задании, заполнение каждого класса нужно сделать самому, импровизируя. В классе Student я, конечно же, решил создать самого студента, реализовав там конструктор, геттеры и сеттеры. В классах Exeption мною были реализованы выводы ошибок, характерных этому классу.

В LabClass я реализовал создание массива, добавление студентов и поиск. В классе LabClassDriver был сделан ввод с клавиатуры элементов массива объектов (Имя, оценка). А в LabClassUI запуск программы и реализация исключений.

Что получилось по итогу Вы можете посмотреть ниже на рисунке 2.

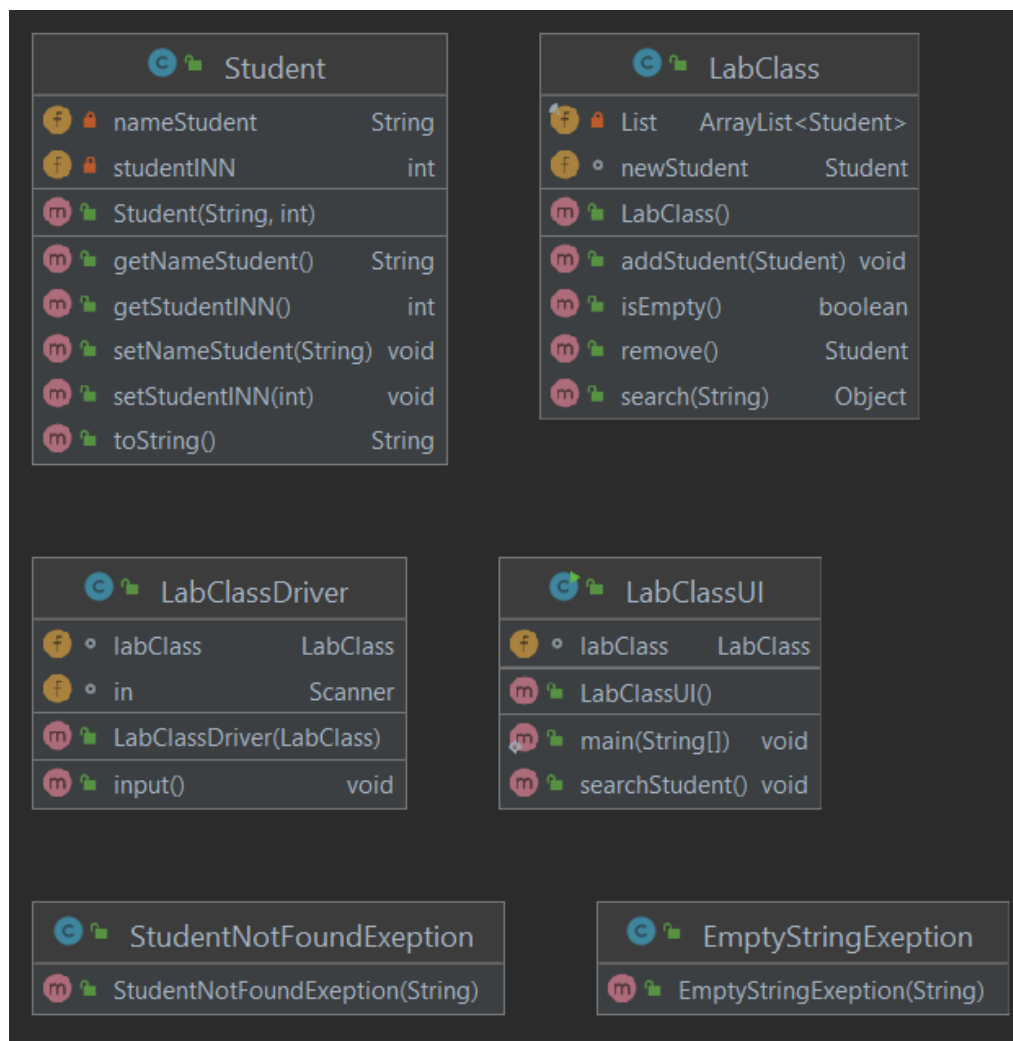


Рисунок 2 – Полученная UML Диаграмма

Код выполненной работы

Здесь в нескольких скриншотах можно увидеть как выглядит код выполненного задания и результат его работы.

```
public class Student {  
    private String nameStudent;  
    private int studentINN;  
  
    public Student(String nameStudent, int studentINN) {  
        this.nameStudent = nameStudent;  
        this.studentINN = studentINN;  
    }  
  
    public String getNameStudent() { return nameStudent; }  
  
    public void setNameStudent(String nameStudent) { this.nameStudent = nameStudent; }  
  
    public int getStudentINN() {  
        return studentINN;  
    }  
  
    public void setStudentINN(int studentINN) {  
        this.studentINN = studentINN;  
    }  
  
    @Override  
    public String toString() {  
        return "Name - " + nameStudent + "; Score - " + studentINN;  
    }  
}
```

Рисунок 3 – Класс Student

```

public class LabClass {
    private final ArrayList<Student> List;
    Student newStudent;

    public LabClass() {
        this.List = new ArrayList<>();
    }

    public void addStudent(Student student) {
        List.add(student);
    }

    public Object search(String nameStudent) throws StudentNotFoundException {
        for (Student student : List) {
            if(student.getNameStudent().equals(nameStudent)) {
                newStudent = student;
                return newStudent;
            }
        }
        throw new StudentNotFoundException( err: "Student " + nameStudent + " is not found!");
    }

    public boolean isEmpty() { return List.isEmpty(); }
    public Student remove() { return List.remove( index: 0); }
}

```

Рисунок 4 – Класс LabClass

```

public class LabClassDriver {
    LabClass labClass;
    Scanner in = new Scanner(System.in);

    public LabClassDriver(LabClass labClass) { this.labClass = labClass; }

    public void input() {
        String nameStudent;
        int num;
        System.out.println("Name Student (0 - End) - ");
        nameStudent = in.next();
        while (!nameStudent.equals("0")) {
            num = in.nextInt();
            labClass.addStudent(new Student(nameStudent, num));
            nameStudent = in.next();
        }
    }
}

```

Рисунок 5 – Класс LabClassDriver

```

public class LabClassUI {
    LabClass labClass;

    public LabClassUI() {
        labClass = new LabClass();
        LabClassDriver labDriver = new LabClassDriver(labClass);
        labDriver.input();
    }

    public void searchStudent() {
        String nameStudent = null;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter name of search Student: ");
        try {
            nameStudent = in.next();
            if (nameStudent.isEmpty())
                throw new EmptyStringException( err: "String is Empty!");
        } catch (EmptyStringException elem) {
            System.out.println(elem.getMessage());
            searchStudent();
        }
        try {
            System.out.println(labClass.search(nameStudent).toString());
        } catch (StudentNotFoundException elem) {
            System.out.println(elem.getMessage());
        }
    }
}

```

Рисунок 6 – Класс LabClassUI

```

    public static void main(String[] args) {
        LabClassUI start = new LabClassUI();
        start.searchStudent();
        start.searchStudent();
        start.searchStudent();
    }
}

```

Рисунок 7 – Продолжение класса LabClassUI

```
public class EmptyStringException extends Exception{
    public EmptyStringException(String err) {
        super(err);
    }
}
```

Рисунок 8 – Класс EmptyStringException

```
public class StudentNotFoundException extends Exception{
    public StudentNotFoundException(String err) {
        super(err);
    }
}
```

Рисунок 9 – Класс StudentNotFoundException

```
Name Student (0 - End) -
Dima 5
Vanya 3
Gosha 4
Kolya 2
0
Enter name of search Student:
Dima
Name - Dima; Score - 5
Enter name of search Student:
gaga
Student gaga is not found!
Enter name of search Student:
Gosha
Name - Gosha; Score - 4

Process finished with exit code 0
```

Рисунок 10 – Результат запуска программы

Вывод

В результате выполнения данной практической работы я научился создавать собственные исключения и применять их на практике в программах на языке программирования Java.