

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет информационной безопасности  
Кафедра инфокоммуникационных технологий

**ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ  
ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ  
Часть 2**

**Лабораторная работа 3  
Наследование и интерфейсы.  
Обработка исключительных ситуаций**



**Минск 2023**

## Лабораторная работа 3

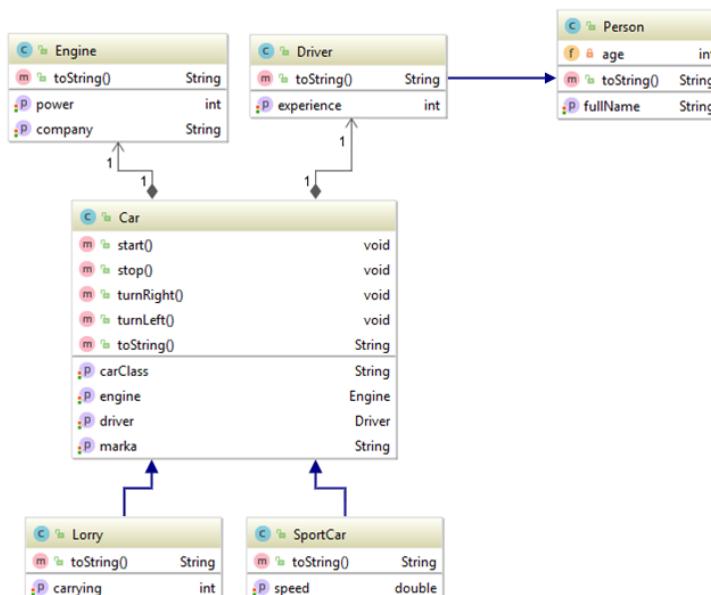
### Наследование и интерфейсы. Обработка исключительных ситуаций

Цель работы: На основании изученного материала в практической работе 3 решить поставленные задачи.

#### Задание к лабораторной работе 3

1. Создайте пример наследования, реализуйте класс `Student` и класс `Magistracy`, магистрант отличается от студента наличием некой научной работы. Класс `Student` должен содержать поля: `String firstName`, `lastName`, `group`. А также, `double averageMark`, содержащее среднюю оценку. Создайте переменную типа `Student`, которая ссылается на объект типа `Magistracy`. Создайте метод `getScholarship()` для класса `Student`, который возвращает сумму стипендии. Если средняя оценка студента равна 8, то сумма 100 руб., иначе 80 руб. Переопределите этот метод в классе `Magistracy`. Если средняя оценка аспиранта равна 8, то сумма 200 руб., иначе 180 руб. Создайте массив типа `Student`, содержащий объекты класса `Student` и `Magistracy`. Вызовите метод `getScholarship()` для каждого элемента массива.

2. Создайте класс `Car` в пакете `com.company.vehicles`, `Engine` в пакете `com.company.details` и `Driver` в пакете `com.company.professions`. Класс `Driver` должен содержать поля – ФИО, стаж вождения. Класс `Engine` должен содержать поля – мощность, производитель. Класс `Car` должен содержать поля – марка автомобиля, класс автомобиля, вес, водитель типа `Driver`, мотор типа `Engine`. Методы `start()`, `stop()`, `turnRight()`, `turnLeft()`, которые выводят на печать: "Поехали", "Останавливаемся", "Поворот направо" или "Поворот налево". А также метод `toString()`, который выводит полную информацию об автомобиле, ее водителе и моторе. Создайте производный от `Car` класс – `Lorry` (грузовой автомобиль), характеризуемый также грузоподъемностью кузова. Создайте производный от `Car` класс – `SportCar`, характеризуемый также предельной скоростью. Класс `Driver` должен расширять класс `Person`.



3. Создайте абстрактный класс "Фрукт" и классы "Яблоко", "Груша", "Слива" расширяющие его. Класс "Фрукт" должен содержать: поле – вес, завершённый метод `printManufacturerInfo(){System.out.print("Made in RB");}`, абстрактный метод, возвращающий стоимость фрукта, который должен быть переопределен в каждом классе наследнике. Метод должен учитывать вес фрукта. Создайте несколько объектов разных классов. Подсчитайте общую стоимость проданных фруктов. А также общую стоимость отдельно проданных яблок, груш и слив.

4. Определите интерфейс `Printable`, содержащий метод `void print()`. Определите класс `Book`, реализующий интерфейс `Printable`. Определите класс `Magazine`, реализующий интерфейс `Printable`. Создайте массив типа `Printable`, который будет содержать книги и журналы. В цикле необходимо пройти по массиву и вызвать метод `print()` для каждого объекта. Создайте статический метод `printMagazines(Printable[] printable)` в классе `Magazine`, который будет выводить на консоль названия только журналов. Создайте статический метод `printBooks(Printable[] printable)` в классе `Book`, который будет выводить на консоль названия только книг. Используйте оператор `instanceof`.

5. Создайте статический метод, который принимает на вход три параметра: `login`, `password` и `confirmPassword`. `login` должен содержать только латинские буквы, цифры и знак подчеркивания. Длина `login` должна быть меньше 20 символов. Если `login` не соответствует этим требованиям, необходимо выбросить `WrongLoginException`. `Password` должен содержать только латинские буквы, цифры и знак подчеркивания. Длина `password` должна быть меньше 20 символов. Также `password` и `confirmPassword` должны быть равны. Если `password` не соответствует этим требованиям, необходимо выбросить `WrongPasswordException`. `WrongPasswordException` и `WrongLoginException` – пользовательские классы исключения с двумя конструкторами – один по умолчанию, второй принимает сообщение исключения и передает его в конструктор класса `Exception`. Обработка исключений проводится внутри метода. Необходимо использовать несколько блоков `catch`. Метод возвращает `true`, если значения верны или `false` в другом случае.