Задание 1

Цель

- Опробовать принцип инкапсуляции на практике.
- Научиться самостоятельно реализовывать в классах методы и переменные с использованием принципа инкапсуляции.

Что нужно сделать

Папка в репозитории не содержит готового проекта, вам необходимо создать проект с нуля. Создайте новый проект Encapsulation и в нём класс Elevator, эмулирующий работу пассажирского лифта. В классе создайте:

- Переменные currentFloor (текущий этаж), minFloor и maxFloor (минимальный и максимальный этажи). Тип переменных — int.
- Конструктор с двумя параметрами minFloor и maxFloor, сохраняющий эти параметры в соответствующих переменных класса.
- Значение переменной currentFloor изначально должно быть равно 1.
- Метод getCurrentFloor, возвращающий текущий этаж, на котором находится лифт.
- Метод moveDown, перемещающий лифт на один этаж вниз (уменьшающий значение переменной currentFloor на единицу).
- Метод moveUp, перемещающий лифт на один этаж вверх.
- Метод move(int floor), перемещающий лифт на заданный в параметре этаж, если он задан верно. Если параметр у метода задан неверно, ничего не делать и выводить в консоль сообщение об ошибке. Этот метод может перемещать лифт только последовательно, по одному этажу, с помощью циклов и методов moveUp и moveDown, и он должен выводить в консоль текущий этаж после каждого перемещения между этажами.

Создайте в этом же проекте класс Main с методом main и напишите в нём следующий код:

```
Elevator elevator = new Elevator(-3, 26);
while (true) {
  System.out.print("Введите номер этажа: ");
  int floor = new Scanner     (System.in).nextInt();
  elevator.move(floor);
}
```

Этот код поможет вам протестировать созданный класс Elevator: он будет создавать лифт и в консоли запрашивать этаж, на который нужно переместить лифт, после чего вызывать у него метод move с указанием полученного из консоли этажа. Запустите получившийся код и убедитесь, что он работает корректно.

Критерии оценки работы

Принято:

- Выполнены все указанные в заданиях требования по написанию кода в классе Elevator.
- Все методы классов работают без ошибок, код компилируется и запускается.
- В результате выполнения кода для тестирования в консоль выводится:
  - сообщение об ошибке, если в консоль введён номер этажа меньше −3 или больше 26;
  - последовательно все номера этажей, если введён этаж, отличный от текущего;
  - ничего, если в консоль введён номер этажа, на котором лифт находится сейчас.

На доработку: задание не выполнено, выполнено неточно либо частично.

Задание 2

Цель

Научиться реализовывать механизм копирования объектов, в том числе объектов иммутабельных классов.

Что нужно сделать

Продолжайте работу в проекте Encapsulation, в котором вы выполняли предыдущее задание. Создайте в этом проекте иммутабельный класс для хранения информации о грузах, передаваемых в курьерскую службу. Название класса придумайте самостоятельно.

Создайте у класса следующие поля:

- габариты;
- масса;
- адрес доставки;
- свойство — можно ли переворачивать;
- регистрационный номер (может содержать буквы);
- является ли груз хрупким.

Названия полей придумайте самостоятельно таким образом, чтобы по ним было понятно, что в них находится. Типы полей задайте в соответствии с данными, которые в них содержатся.

Габариты — ширина, высота и длина — должны храниться в отдельном иммутабельном классе Dimensions. Создайте класс Dimensions с соответствующими полями и реализуйте в нём метод вычисления объёма груза (название метода придумайте самостоятельно).

Реализуйте в классе методы, дающие возможность изменять адрес доставки, габариты и массу груза без изменения исходного объекта путём создания его копии.

Напишите в методе main класса Main дополнительный код, который будет создавать экземпляр класса (объект) груза и его копии при изменении тех или иный полей. Напишите также код, который позволит проверить, что копирование действительно происходит.

Критерии оценки работы

Принято:

- Выполнены все указанные в задании требования по написанию кода в классе для хранения данных о заказе на доставку и классе Dimensions.
- Оба реализованных класса являются иммутабельными.
- Имена методов соответствуют действиям, которые они выполняют.
- Имена и типы переменных соответствуют хранимым в них данным.
- Методы, изменяющие адрес доставки, габариты и массу груза, создают копии исходного объекта, а исходный объект при этом остаётся неизменным.
- Все методы классов работают без ошибок, код компилируется.

На доработку: задание не выполнено полностью или частично.