

Maven-зависимости. Параметризованные классы.

1. Maven-зависимости.

- 1.1. Создайте maven-проект. В нем создайте класс для хранения квадратного трехчлена с методом решения соответствующего квадратного уравнения (параметров нет, а результат — массив корней). Напишите набор unit-тестов для этого класса.
- 1.2. Создайте отдельный maven-проект. В нем — класс с полем типа «Квадратный трехчлен» из предыдущего проекта. Класс содержит метод, который возвращает больший корень или выбрасывает исключение, если корней нет. Напишите для него тесты.

2. Параметризованные классы.

- 2.1. Разработать интерфейс для понятия «функция одного вещественного аргумента, определенная на отрезке $[a; b]$ ». Интерфейс должен содержать метод вычисления значения функции при заданном аргументе и методы получения границ отрезка.
- 2.2. Написать набор классов, реализующих этот интерфейс: функции вида $f(x) = Ax + B$, $f(x) = A \sin(Bx)$, $f(x) = (Ax + B) / (Cx + D)$, $f(x) = A \exp(x) + B$.
- 2.3. Функционалом в математике называют отображение, аргументы которого являются функциями, а результат — числом. Разработайте параметризованный интерфейс «Функционал от одного аргумента» с методом вычисления значения функционала от заданной функции. Параметром является тип функций, т.е. функционалы для работы с линейными и тригонометрическими функциями должны быть синтаксически разные.
- 2.4. Напишите набор классов, реализующих интерфейс функционала:
 - сумма значений функции на концах отрезка и в его середине,
 - определенный интеграл на отрезке $[a; b]$ (пределы интегрирования хранятся как поля и устанавливаются конструктором, если область определения функции не содержится в $[a; b]$, то выбрасывается исключение), интегрирование производить методом прямоугольников.

3. *Компараторы.

- 3.1. Для класса «Товар» из предыдущего задания (про наследование) напишите класс-компаратор. Два товара сравниваются по названию, если названия равны, то далее — по описанию.
- 3.2. Изучите методы сортировки массивов из стандартного класса Arrays (<https://docs.oracle.com/javase/8/docs/api/java/util/Arrays.html>).
- 3.3. Напишите класс ComparatorDemo со статическим методом sortGoods, который получает на вход массив товаров и компаратор и сортирует массив по возрастанию (метод будет простой отсылкой к соответствующему методу класса Arrays, но для него надо написать тесты. Метод должен быть в состоянии принять на вход массив товаров любого производного типа (элементы в массиве одного типа, но это может быть производный класс от класса «Товар»), но при этом использовать общий компаратор для класса «Товар».

4. Проверьте, что на все классы написаны необходимые unit-тесты.