

ЛАБОРАТОРНАЯ РАБОТА 1

Написать систему автотестов для предложенных задач, для каждой функции пишется 1 позитивный и 1 негативный тест. Тесты должны удовлетворять принципам юнит тестирования, предусмотреть их логирование (JUnit+java – можно использовать другие языки)

Уровень 1

Определите класс "Дробь" в виде пары (m, n). Класс должен содержать несколько конструкторов. Реализуйте методы для сложения, вычитания, умножения, деления и сокращения дробей. Методы для сложения и умножения дробей сделайте с переменным числом параметров. Объявите массив из k дробей, введите/выведите значения для массива дробей. Создайте массив объектов и передайте его в метод, который будет изменять каждый элемент массива с четным индексом путем добавления, следующего за ним элемента массива.

Уровень 2

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.

Уровень 3

Напишите класс Student, предоставляющий информацию об имени студента методом getName() и о его курсе методом getCourse(). Напишите метод printStudents(LinkedList students, int course), который получает список студентов и номер курса и печатает имена тех студентов из списка, которые обучаются на данном курсе. Протестируйте метод (для этого предварительно придется создать десяток объектов класса Student и поместить их в список). Напишите методы union(LinkedList set1, LinkedList set2) и intersect(LinkedList

set1, LinkedList set2), реализующие операции объединения и пересечения двух множеств. Протестируйте работу этих методов на двух предварительно заполненных множествах (понадобится написать вспомогательный метод, выводящий все элементы множества на консоль.) Реализуйте интерфейс Comparable так, чтобы студенты сортировались по номеру курса. Проверьте работу, используя класс TreeSet.

Уровень 4

На базе коллекций реализуйте структуру хранения чисел с поддержкой следующих операций: добавление/удаление числа; поиск числа, наиболее близкого к заданному (т.е. модуль разницы минимален).