# Практические задания

**День 3**. **Исключения, вложенные классы, обобщения**

1. **Проверка логина и пароля**

Создать 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.

Используем multi-catch block.

Метод возвращает **true**, если значения верны или **false** в другом случае.

1. **Обычный внутренний класс**

Создать класс User, содержащий private переменные login, password.

Создать внутренний класс Query в классе User. Класс Query содержит метод printToLog(), который распечатывает на консоль сообщение о том что пользователь с таким то логином и паролем отправил запрос.

Класс User, содержит метод createQuery(), в котором создается объект класса Query и вызывается метод printToLog().

В методе main(): создать экземпляр класса User и вызвать метод createQuery(); создать экземпляр класса Query и вызвать метод printToLog() используя конструкцию user.new Query(); создать экземпляр класса Query и вызвать метод printToLog() используя конструкцию new User().new Query();

1. **Анонимный класс Cookable**

Создать интерфейс Cookable, содержащий метод void cook(String str).

Создать класс Food, содержащий метод public void prepare(Cookable c, String str) { c.cook(str);}

Создать экземпляр класса Food и вызвать его метод prepare().

На вход метода prepare() передать экземпляр анонимного класса, расширяющего интерфейс Cookable.

1. **Обобщенный класс с тремя параметрами**

Создать обобщенный класс с тремя параметрами (T, V, K).

Класс содержит три переменные типа (T, V, K), конструктор, принимающий на вход параметры типа (T, V, K), методы возвращающие значения трех переменных. Создать метод, выводящий на консоль имена классов для трех переменных класса.

Наложить ограничения на параметры типа: T должен реализовать интерфейс Comparable (классы оболочки, String), V должен реализовать интерфейс Serializable и расширять класс Animal, K должен расширять класс Number.

1. **Нахождения минимального и максимального элемента массива**

Написать обобщенный класс MinMax, который содержит методы для нахождения минимального и максимального элемента массива. Массив является переменной класса. Массив должен передаваться в класс через конструктор.