1. Что такое класс

2. Сделайте класс **Worker** без конструктора, в котором будут следующие **public** свойства - name (имя), age (возраст), salary (зарплата).

Создайте два экземпляра этого класса, установив указанные свойства.

Выведите на экран сумму зарплат и сумму возрастов обоих экземпляров.

1. Что такое свойства класса

2. Сделайте класс **Worker** без конструктора, в котором будут следующие **private** свойства - name (имя), age (возраст), salary (зарплата) и методы для доступа к этим свойствам.

Создайте 2 экземпляра этого класса.

Выведите на экран сумму зарплат и сумму возрастов обоих экземпляров.

1. Что такое методы класса

2. Сделайте класс **Worker**, в котором будут следующие **private** свойства - name (имя), age (возраст), salary (зарплата). Сделайте так, чтобы эти свойства заполнялись в основном конструкторе при создании экземпляра класса.

Создайте экземпляр этого класса.

Выведите на экран произведение его возраста и зарплаты.

1. Перечислите основные модификаторы доступа к свойствам и методам класса

2. Сделайте класс **Worker**, в котором будут следующие **private** свойства - name (имя), age(возраст), salary (зарплата). Сделайте так, чтобы эти свойства заполнялись в дополнительном конструкторе при создании экземпляра класса. Сделайте также **public** методы для доступа к этим свойствам.

Создайте экземпляр этого класса.

Выведите на экран произведение его возраста и зарплаты.

1. Что такое конструктор класса

2. Сделайте класс **User**, в котором будут следующие **protected** свойства - name (имя), age (возраст).

Сделайте класс **Worker**, который наследует от класса **User** и вносит дополнительное **private** свойство salary (зарплата), а также методы для доступа к этому свойству с валидацией (зарплата должна быть положительной).

Создайте два экземпляра класса **Worker**.

Найдите сумму зарплата этих экземпляров.

1. Какие виды конструкторов вы знаете

2. Сделайте класс **Worker**, в котором будут следующие **private** свойства - name (имя), salary (зарплата).

Сделайте класс **Driver** (Водитель), который будет наследоваться от класса **Worker**. Этот класс должен содержать следующие **private** свойства: водительский стаж, категория вождения (A, B, C). Сделайте методы доступа к этим свойствам с проверкой правильности вводимых данных (стаж положительный, категория из списка)

Создайте экземпляр класса **Driver** и заполните его свойства

1. Как объявляется основной конструктор класса

2. Сделайте класс **Worker**, в котором будут **private** свойства - name (имя), age (возраст), salary (зарплата) и **public** методы для доступа к этим свойствам.

Сделайте проверку возраста (валидный возраст от 14 до 80 лет).

При попытке задать не правильный возраст выбросить исключение. 1. Как объявляется дополнительный конструктор класса

2. Сделайте класс **User**, в котором будут следующие **protected** свойства - name (имя), age (возраст).

Сделайте класс **Student**, который наследует от класса **User** и вносит дополнительные **private** свойства стипендия, курс, а также методы для доступа ко всем полям.

1. Что такое интерфейс класса

2. Сделайте **абстрактный** класс **User**, в котором будут следующие свойства: *name* (имя), *age* (возраст). Для свойства *age* сделать проверку (только положительные).

Сделайте класс **Student**, который наследует от класса **User** и вносит дополнительные **private** свойства стипендия, курс, а также методы доступа для них.

1. Перечислите основные принципы ООП

2. Сделайте **абстрактный** класс **Animal** (животное) и пару методов в нем (born, die).

Сделайте интерфейс Fly (летающий) c методом полет.

Сделайте класс Муха, наследник класса **Animal** и реализующий интерфейс **Fly**

1. Что такое наследование

2. Сделайте **абстрактный** класс **Animal** (животное) с методом **move** (движение)

Сделайте пару потомков этого класса с разной реализацией метода **move**

1. Что такое полиморфизм

2. Сделайте класс. Добавьте в него свойства со всеми известными модификаторами доступа. Для тех, которым это необходимо сделайте методы доступа (т.е. все свойства в итоге должны быть доступны извне).

1. Что такое инкапсуляция

2. Сделайте два интерфейса (с методами) и класс, реализующий эти интерфейсы.1. Что такое абстрактный класс

2. Сделайте абстрактный класс и интерфейс с методами.

Сделайте класс - их наследник

1. В чем отличие абстрактного класса от интерфейса

2. Сделайте **абстрактный** класс с основным конструктором и парой методов.

Сделайте класс, наследник абстрактного класса.1. Как создать статический метод класса

2. Сделайте **абстрактный** класс с дополнительным конструктором и парой методов.

Сделайте класс, наследник абстрактного класса.

1. Что такое статический метод/свойство класса

2. Сделайте **абстрактный** класс с основным конструктором и парой методов.

Сделайте класс, наследник абстрактного класса.

1. Назовите базовый класс котлин и его методы

2. Сделайте **абстрактный** класс с дополнительным конструктором и парой методов.

Сделайте класс, наследник абстрактного класса.

1. Перечислите типы шаблонов проектирования

2. Сделайте два интерфейса (с методами) и класс, реализующий эти интерфейсы.

1. Шаблон **Simple Factory** (простая фабрика)

2. Сделайте абстрактный класс и интерфейс с методами.

Сделайте класс - их наследник

1. Шаблон **Singleton** (одиночка)

2. Сделайте **абстрактный** класс **Animal** (животное) с методом **move** (движение)

Сделайте пару потомков этого класса с разной реализацией метода **move**

1. Шаблон **декоратор**

2. Сделайте класс. Добавьте в него свойства со всеми известными модификаторами доступа. Для тех, которым это необходимо, сделайте методы доступа (т.е. все свойства в итоге должны быть доступны извне).

1. Шаблон **заместитель** (proxy)

2. Сделайте класс **Worker** без конструктора, в котором будут следующие **public** свойства - name (имя), age (возраст), salary (зарплата).

Создайте два экземпляра этого класса, установив указанные свойства.

Выведите на экран сумму зарплат и сумму возрастов обоих экземпляров.

1. Шаблон **наблюдатель** (observer)

2. Сделайте класс **Worker** без конструктора, в котором будут следующие **private** свойства - name (имя), age (возраст), salary (зарплата) и методы для доступа к этим свойствам.

Создайте 2 экземпляра этого класса.

Выведите на экран сумму зарплат и сумму возрастов обоих экземпляров.