**Лабораторная работа №2.**

Тема: Объективно-ориентированное программирование и диаграмма классов

Цель работы: освоить навыки использования методов ООП: инкапсуляция, полиформизм, наследование

1. **Необходимые теоретические сведения**

**Объектно-ориентированное программирование** - это методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного класса, а классы образуют иерархию наследования.

Инкапсуляция - упаковка данных и функций в единый компонент. В общем случае, в разных языках программирования термин «инкапсуляция» относится к одному из или обоим определениям:

* механизм языка, позволяющий ограничить доступ одних компонентов программы к другим. Например, ограничивается доступ к переменным объекта класса.
* языковая конструкция, позволяющая связать данные с методами, предназначенными для обработки этих данных.

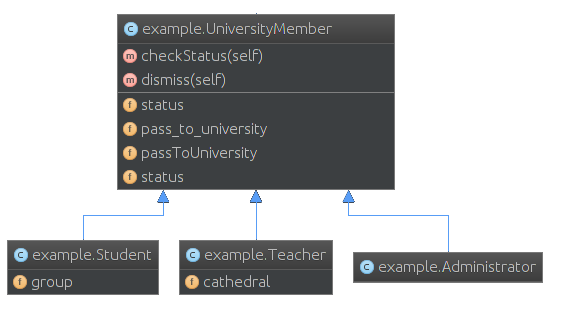
Класс - это способ описания сущности, определяющий состояние и поведение, зависящее от этого состояния, а также правила для взаимодействия с данной сущностью (методы и уровни доступа к переменным класса).

Назначение класса:

1. Для создания сложной структуры данных со сложным поведением;
2. Для поддержки механизмов инкапсуляции, полиморфизма и наследования;
3. Для удобства. Большая задача разбивается на много функциональных блоков меньшего размера, каждый из который реализуется классом.

Объект - это конкретный экземпляр класса, поля которого проинициализированы.

Наследование - это метод расширения функциональности классов и снижения дубликации кода, когда один класс полностью забирает себе (наследует) все поля и методы другого класса (класса родителя) и добавляет новые поля и методы или переопределяет старые, тем самым расширяя/изменяя функциональность класса в сравнении с классом-родителем.



Полиморфизм - это способность объекта использовать методы производного класса, который не существует на момент создания базового.

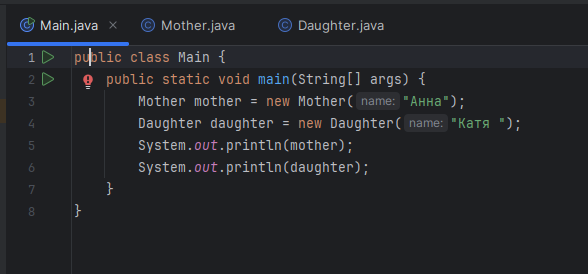
Любой метод можно переопределить, то есть повторно реализовать в подклассе. В этом случае для экземпляров базового класса будет вызываться базовый метод, а для экземпляров производного — перегруженный.

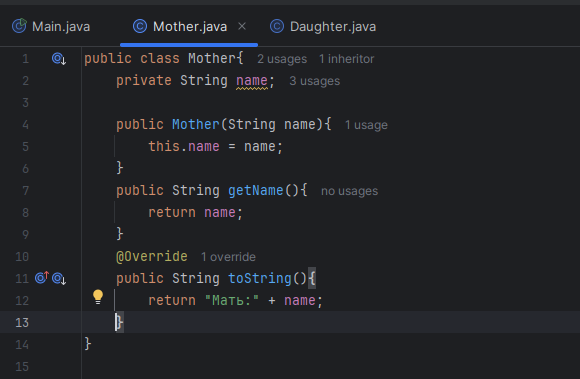
1. **Индивидуальные задания**

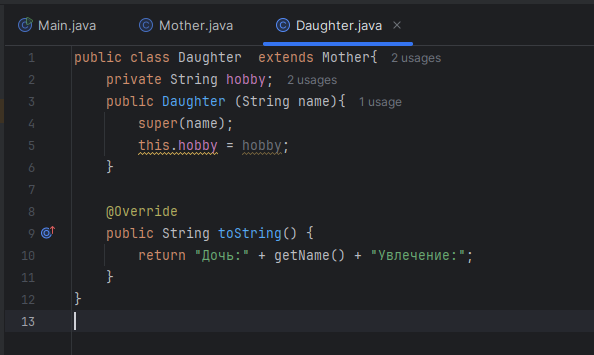
1. Напишите программу с классом Mother от которого наследуется класс Daughter.

Сделайте так, чтобы результат print(object) был разный.

Воспользуйтесь принципами полиморфизма, наследования и инкапсуляции.





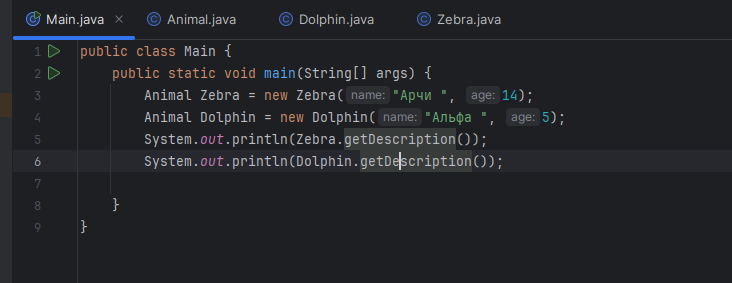


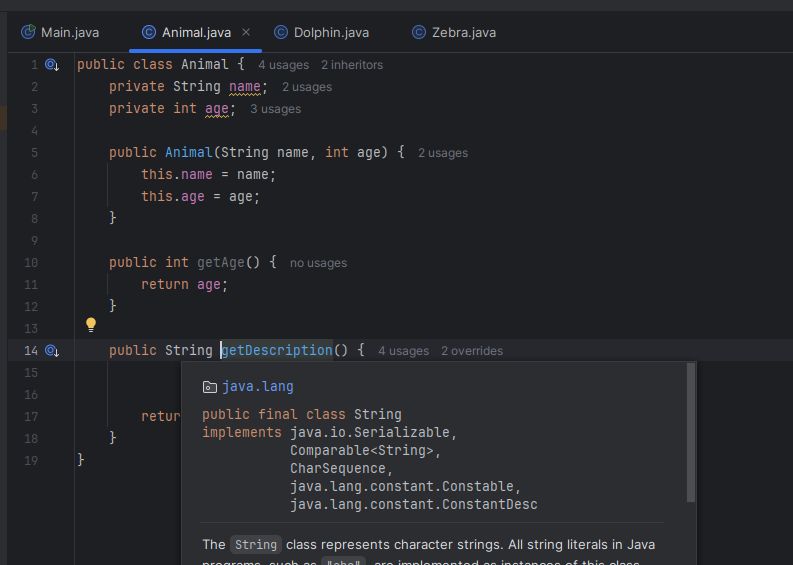
2. Реализйте класс Animal. Внутри объявите поле для имени и возраста.

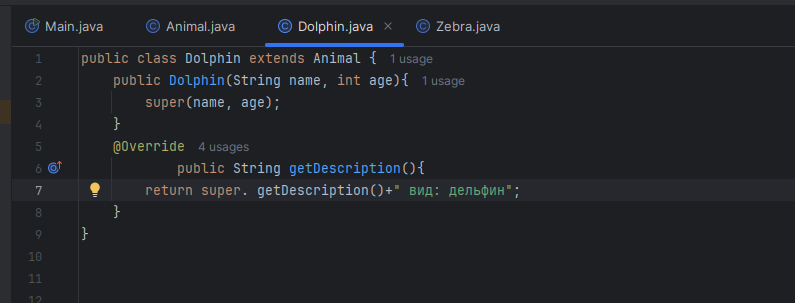
От класса Animal унаследуйте класс Zebra и Dolphin.

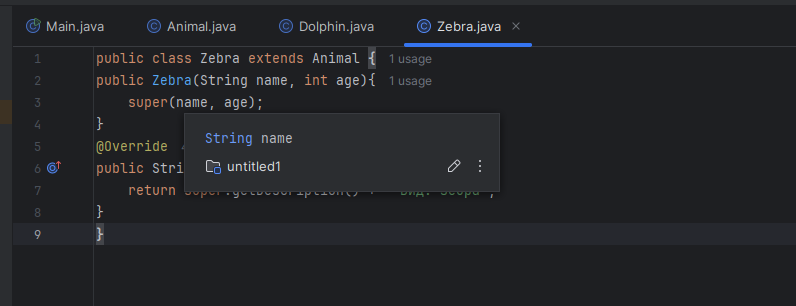
Оба класса могут вернуть описание, содержащее имя, возраст и какую-то доп.информацию, например, что это за вид животного.

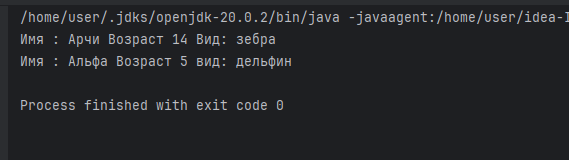
Воспользуйтесь принципами полиморфизма, наследования и инкапсуляции.











3. Игра "Арифметические драконы" предназначена для обучения детей арифметике. На героя нападает дракон, который задаёт вопрос на сложение (если дракон зелёный), вычитание (красный) или умножение (чёрный).

Реализуйте следующие классы:

