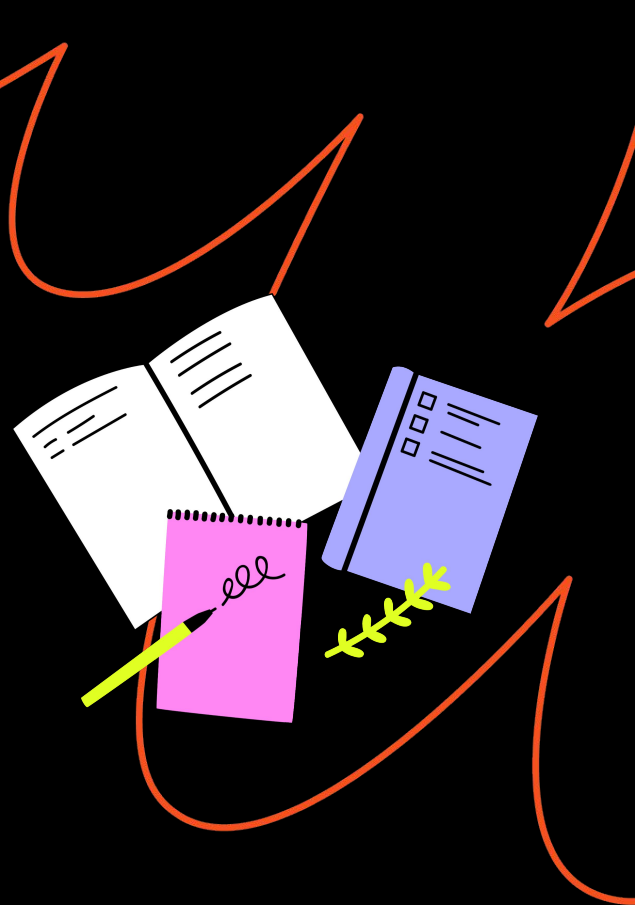




# Введение в ООП. Концепты и свои типы

ООП





**Разговор о...**



# Разговор о...

1. Введение
2. Класс и объект: детали
3. Наполнение
4. Концепты ООП
5. Примеры использования
6. Итоги



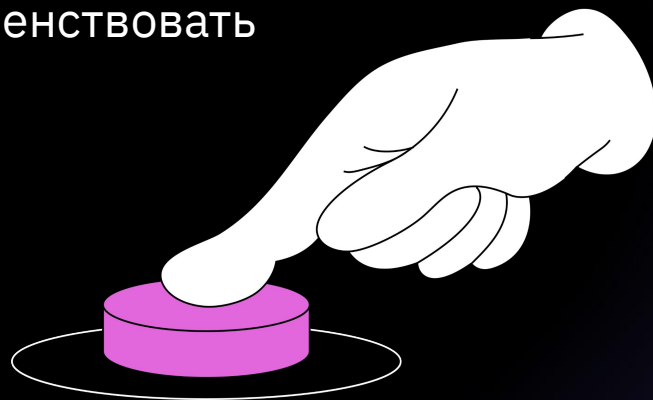
# Введение



# Введение

Способ управления сложными системами:  
«Разделяй и властвуй».

Это значит, что программную систему нужно разбить на подсистемы так, чтобы работу каждой из них можно было рассматривать и совершенствовать независимо от других.



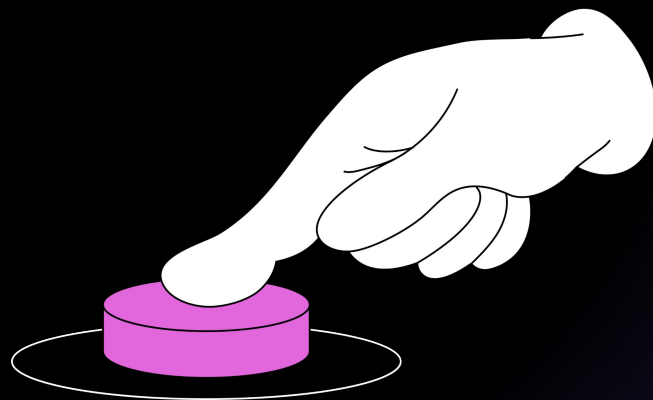
# Введение

Правильное план – реализация только после абстракций

Абстракция – что делает?

Поведение – как делает?

Спецификация – набор правил, описывающих API



# Класс и его экземпляры



# Класс

**Класс** — это «чертеж» (описание) сущности предметной области, позволяющий выделить некоторые общие характеристики, состояние и поведение, зависящее от состояния.

**Предметная область** — множество всех предметов (явлений) решаемой проблемы.





# Экземпляр класса

**Экземпляр класса** – отдельный представитель класса, имеющий КОНКРЕТНОЕ состояние и поведение, которое полностью определяется описанием класса.

**Состояние** – набор данных (полей, атрибутов, членов класса).

**Поведение** – функции для работы с данными и выполнения полезной работы.



# Экземпляр класса

- Поля
- Константы
- События
- Конструкторы
- Методы



# ООП

**ООП** – объектно-ориентированное программирование.

Парадигма (стиль, шаблон) разработки ПО, основными понятиями которой являются классы и объекты. Говорят, что разработка в стиле ООП ведется с использованием классов объектов, которые обладают состоянием и поведением, зависящим от этого состояния.



# ООП: Инкапсуляция

**Инкапсуляция** – это свойство системы, позволяющее объединить данные и методы, работающие с ними в классе, скрыв детали реализации и защитив от пользователя этого класса объектов.



# ООП: Инкапсуляция

```
public class Point2D {  
    int x;  
    int y;  
}
```



# ООП: Инкапсуляция. Задача

Описать робота, который...



# ООП: Наследование

**Наследование** — это свойство системы, позволяющее описать новый класс на основе уже существующего, с частично или полностью заимствуемой функциональностью.

Класс, от которого производится наследование, называется базовым или родительским. Новый класс — потомком, наследником или производным классом.



# ООП: Наследование

```
public class Point2D {  
    int x;  
    int y;  
}  
  
public class Point3D extends Point2D {  
    int z;  
}
```





# ООП: Наследование

Описать игру, которая...



# ООП: Полиморфизм

**Полиморфизм** – это свойство системы, использовать объекты с одинаковым интерфейсом без информации о типе и внутренней структуре объекта.

Полиморфизм - способность использовать объект вне зависимости от его реализации, благодаря, полиморфной переменной – это переменная, которая может принимать значения разных типов.



# ООП: Полиморфизм

```
Point2D av = new Point3D();
```



# ООП: Полиморфизм

Переделать игру, которая...



# Хотелось бы подвести итоги...

Хотелось бы, но не могу




# Тренировка

Сделать работу с точками трёхмерного пространства.  
Добавить в игру новый персонаж “лекарь”.



**Спасибо  
за внимание**

A yellow smiley face is drawn over the text. It has two vertical lines for eyes and a wide, curved line for a mouth.