

# Объекты и классы

## Взаимодействие между ними




Олег  
Булыгин



## Олег Булыгин

IT-аудитор в ПАО Сбербанк

 [obulygin91@ya.ru](mailto:obulygin91@ya.ru)

|  [fb.me/obulygin91](https://fb.me/obulygin91)



# План занятия

1. [Что такое IDE](#)
2. [Различные подходы к программированию](#)
3. [Декомпозиция и абстракция](#)
4. [Объекты и классы](#)
5. [Атрибуты и методы классов](#)
6. [Взаимодействие объектов](#)
7. [Домашнее задание](#)



# Что такое IDE?

---

# Что такое IDE?

**I**ntegrated **D**evelopment **E**nvironment — система программных средств, предназначенная для разработки программного обеспечения.

Возможности:

- подсветка синтаксиса,
- автоматическое форматирование кода,
- отладка кода — возможность пошагового выполнения программы,
- интеграция с системой контроля версий,
- и многое другое.

---

# Какую IDE выбрать?

- [Pycharm \(версия Community\)](#)
- Visual Studio
- Jupyter
- WingWare
- Eclipse

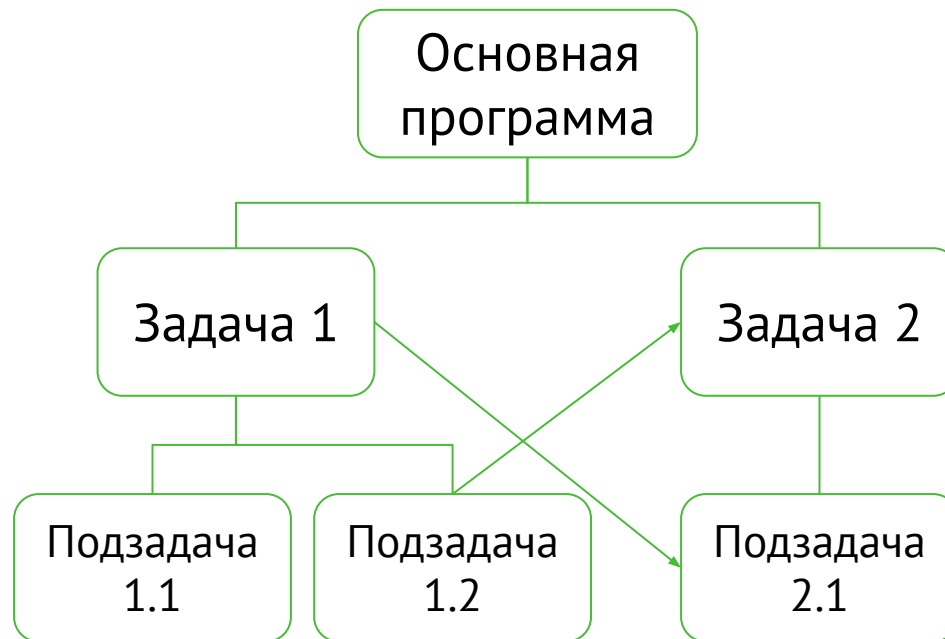
[Руководство по использованию Pycharm](#) вы найдёте в грануле этого занятия.



# **Различные подходы к программированию**

# Процедурное программирование

— это парадигма программирования, в которой используется линейный или нисходящий подход. Он полагается на процедуры или подпрограммы для выполнения вычислений.

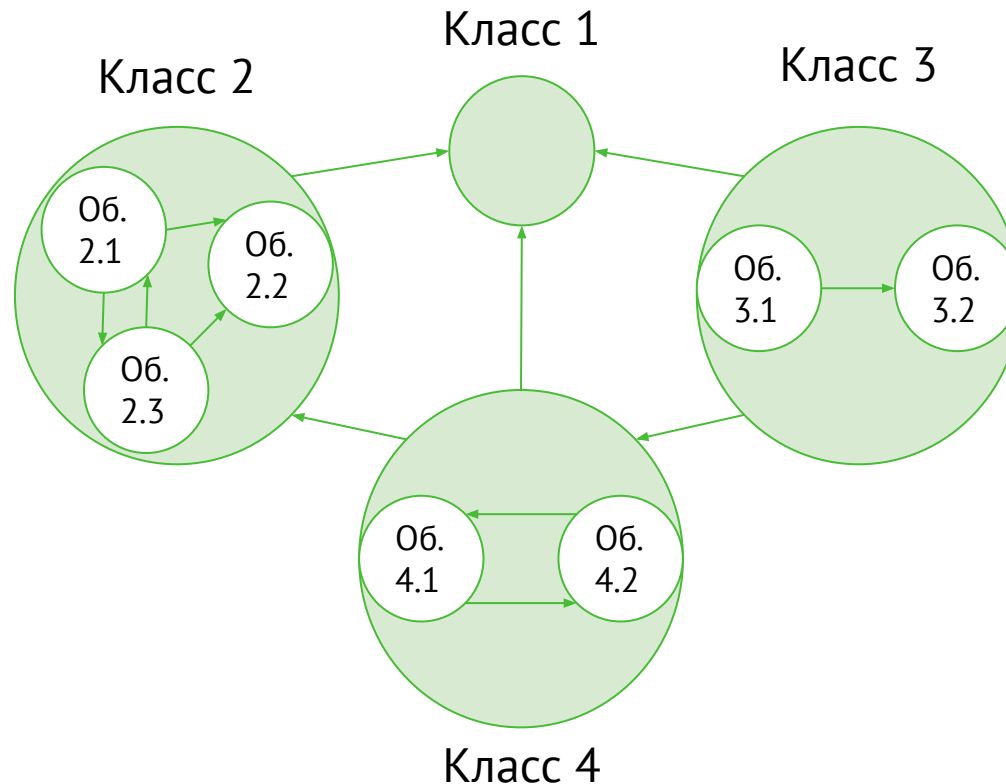


Пример взаимосвязей функций в программе



# Объектно-ориентированное программирование

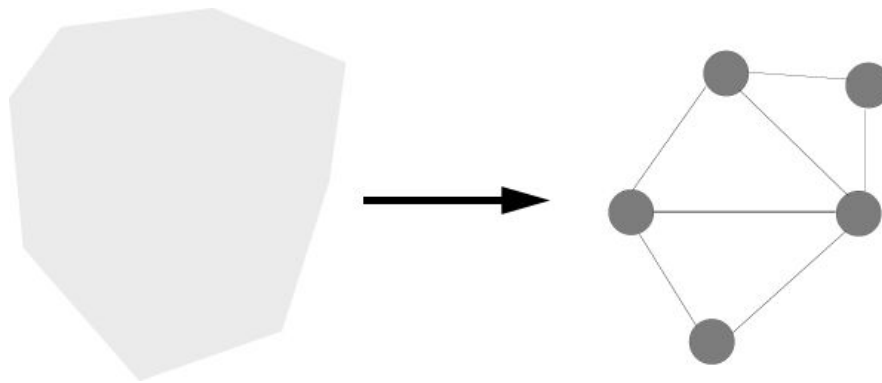
— это парадигма программирования, построенная вокруг объектов. Она разделяет данные на объекты и описывает содержимое и поведение объекта посредством объявления классов.



# Декомпозиция и абстракция

От сложного к простому:

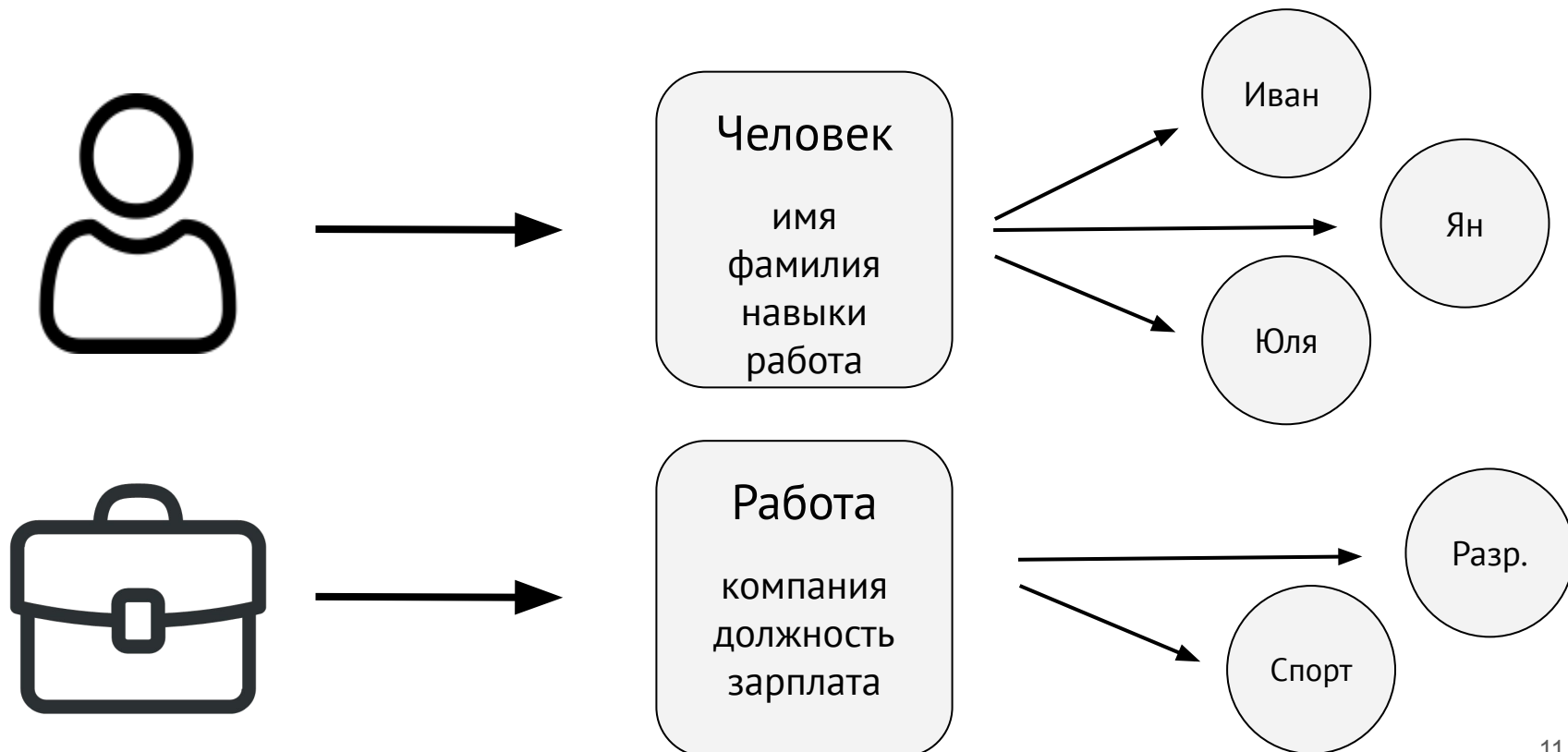
- Как нам разделить программу на менее сложные подпрограммы?
- Как выделить значимые характеристики объектов и зачем нам это надо?



Пример декомпозиции

# Декомпозиция и абстракция

**Абстракция** — выделение существенных характеристик объекта, отличающих его от других объектов.



# Объекты и Классы

**Класс** — способ описания сущности, определяющий состояние и поведение, зависящее от этого состояния, а также правила для взаимодействия с данной сущностью.

`class` Person



```
class Person:
    name = ...
    last_name = ...

    def run(self, time):
        ...
```

# Объекты и Классы

**Объект** — отдельный представитель класса, имеющий конкретное состояние и поведение, полностью определяемое классом.

*Peter*



```
peter = Person()
```

# Атрибуты и методы классов

**Атрибут** — переменная, которая хранит некоторые данные.

*Peter*



Peter  
Parker  
Age 26  
Photographer

```
peter = Person()  
peter.name = 'Peter'  
peter.last_name = 'Parker'  
peter.age = 26  
peter.job = 'photographer'
```

# Атрибуты и методы классов

**Метод** — функция, с помощью которой объект может совершать действия.

*Peter*

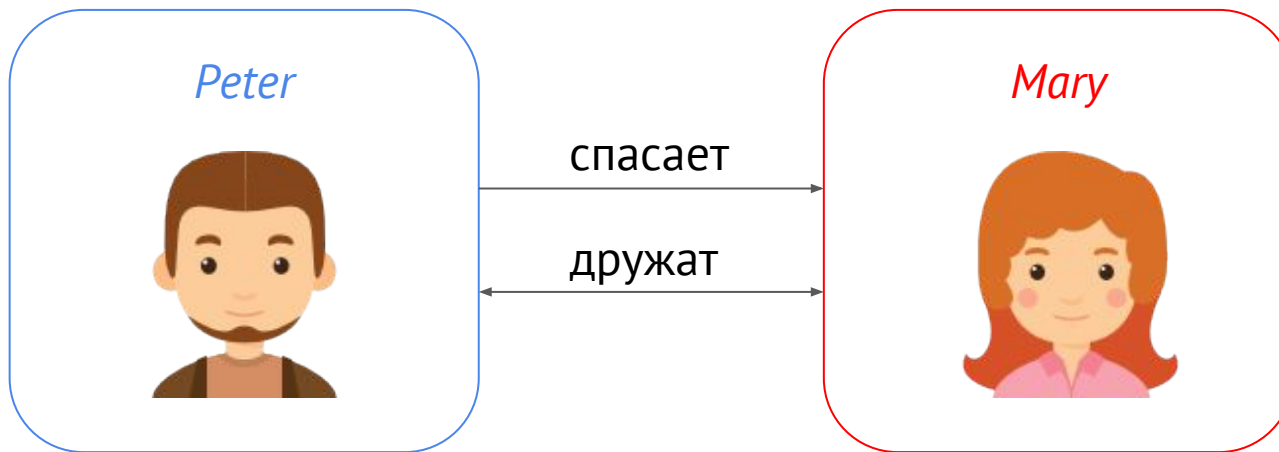


бегает  
прыгает  
спасает  
делает фото

```
peter = Person()  
peter.run(distance=100)  
peter.jump(height=10)  
peter.rescue('Mary Jane')  
peter.photo('Spider Man')
```

# Взаимодействие объектов

**Интерфейс** — совокупность всех методов, доступных для использования.







## Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и  
пишите отзыв о лекции!**

**Олег Булыгин**



[obulygin91@ya.ru](mailto:obulygin91@ya.ru)



[fb.me/obulygin91](https://fb.me/obulygin91)