

# Лекция 7

Проектирование приложений с  
использованием ООП

# Сложность программного обеспечения

## Любительское ПО

- 1-10 разработчиков
- Короткий срок разработки и поддержк

## Промышленное ПО

- Десятки и сотни разработчиков
- десятки лет поддержки

# Примеры промышленных программ

- Диспетчеризация авиа и ж/д транспорта
- Системы машинного обучения и искусственного интеллекта
- Операционные системы

# Составляющие сложности ПО

- сложность реальной предметной области;
- трудность управления процессом разработки;
- необходимость обеспечить достаточную гибкость;

# Сложность реального мира

- Сложность системы для которой разрабатывается ПО
- Трудности взаимодействия с пользователем
- Эволюция системы и проблемы во время разработки

# Трудность управления процессом разработки

- Объемы кода ПО достигают миллионов строк кода
- Команды разработчиков состоят из сотен и тысяч людей

# Гибкость ПО

- Высокая стоимость разработки требует максимального переиспользования наработок
- Разработанные системы в идеальном случае могут использоваться различными компаниями

# Декомпозиция задачи

Разделяй и властвуй

Алгоритмическая декомпозиция

- задача разбивается на последовательность шагов необходимых для решения

Объектно-ориентированная декомпозиция

- задача разбивается на объекты из предметной области взаимодействие которых определяет решение



# 3 этапа разработки

- Анализ
- Проектирование
- Программирование

# 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.

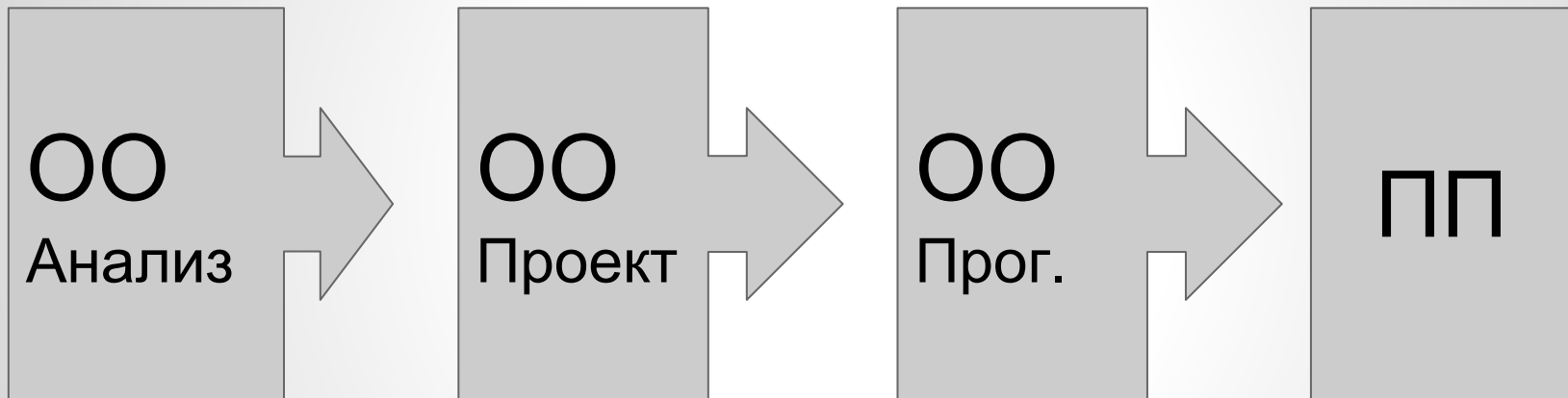
# Объектно-ориентированное проектирование

это методология проектирования, соединяющая в себе процесс объектной декомпозиции и приемы представления логической и физической, а также статической и динамической моделей проектируемой системы.

# Объектно-ориентированный анализ

это методология, при которой требования к системе воспринимаются с точки зрения классов и объектов, выявленных в предметной области.

# Связь понятий



# Абстрагирование

Абстракция выделяет существенные характеристики некоторого объекта, отличающие его от всех других видов объектов и, таким образом, четко определяет его концептуальные границы с точки зрения наблюдателя.

```
class TemperatureSensor {  
    private double m_temperature;  
    public double TemperatureC {  
        get {  
            return m_temperature;  
        }  
    }  
    public double TemperatureF {  
        get {  
            return m_temperature * 1.8 + 32;  
        }  
    }  
}
```



# Инкапсуляция и абстракция

- Инкапсуляция скрывает сложное внутреннее устройство
- Абстракция оставляет видимым только значимые детали



# Модульность

Модули - связанные наборы классов решающие конкретную задачу

- Модуль должен иметь минимальный интерфейс с остальной частью программы

# Иерархия

упорядочение абстракций и расположение их по уровням

- Наследование (человек -> примат -> млекопитающее -> хордовое ...)
- Агрегация (человек -> пищеварительная система + нервная система + ...)

# Преимущества объектной модели

- Максимально используются выразительные средства ЯП
- Повышает пригодность для повторного использования программ
- Разбивает сложность задачи на несвязанные друг с другом модули
- Ориентирована на человеческое восприятие мира