

# Программирование на языке C++

## Лекция 4

Объектно-ориентированное программирование

Александр Смаль

## Ещё раз об ООП

*Объектно-ориентированное программирование* — концепция программирования, основанная на понятиях объектов и классов.

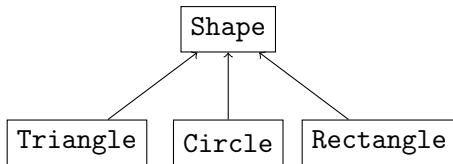
### Основные принципы:

- инкапсуляция,
- наследование,
- полиморфизм,
- абстракция.

Подробнее о принципах проектирования ООП-программ можно узнать по ключевым слову „шаблоны проектирования”.

# Как правильно построить иерархию?

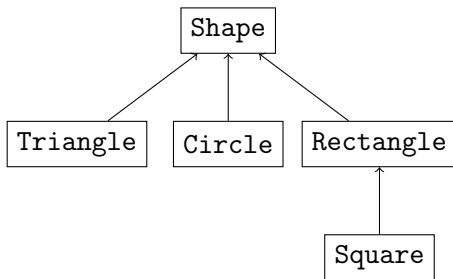
Иерархия геометрических фигур:



Куда добавить класс Square?

## Как правильно построить иерархию?

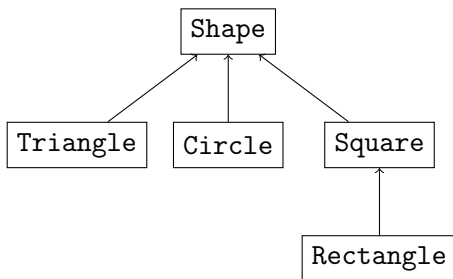
Квадрат — это прямоугольник, у которого все стороны равны.



```
void double_width(Rectangle & r) {  
    r.set_width(r.width() * 2);  
}
```

## Как правильно построить иерархию?

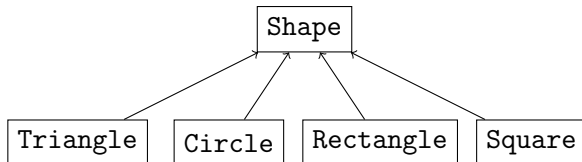
Прямоугольник задаётся двумя сторонами, а квадрат — одной.



```
double area(Square const& s) {  
    return s.width() * s.width();  
}
```

## Как правильно построить иерархию?

Правильное решение — сделать эти классы независимыми:



# Агрегирование vs наследование

- *Агрегирование* — это включение объекта одного класса в качестве поля в другой.
- Наследование устанавливает более сильные связи между классами, нежели агрегирование:
  - приведение между объектами,
  - доступ к `protected` членам.
- Если наследование можно заменить легко на агрегирование, то это нужно сделать.

## Примеры некорректного наследования

- Класс `Circle` унаследовать от класса `Point`.
- Класс `LinearSystem` унаследовать от класса `Matrix`.

# Принцип подстановки Барбары Лисков

## Liskov Substitution Principle (LSP)

*Функции, работающие с базовым классом, должны иметь возможность работать с подклассами не зная об этом.*

Этот принцип является важнейшим критерием при построении иерархий наследования.

## Другие формулировки

- Поведение наследуемых классов не должно противоречить поведению, заданному базовым классом.
- Подкласс не должен требовать от вызывающего кода больше, чем базовый класс, и не должен предоставлять вызывающему коду меньше, чем базовый класс