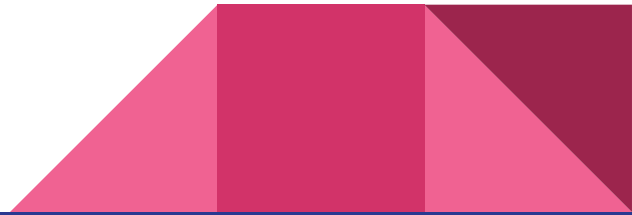
Лекция 14.

Отношения между классами: наследование, вложение

Между классами возможны два типа отношений:

1. Отношение типа **is-a** (есть, является), при котором один класс есть подвидом другого класса. При таком отношении один класс расширяет (детализирует) возможности другого класса. Расширение возможностей класса осуществляется благодаря использованию **наследования**.

2. Отношение, при котором существует взаимосвязь между двумя классами.



Подвиды отношений:

2.1. Отношение типа **has-a** (класс содержит другой класс). В этом случае в классе объявляется одних или несколько экземпляров другого класса. При данном отношении возможны два случая взаимодействия. Первый случай, это когда объект (**экземпляр**), который объявлен в классе, не является составной частью класса (**агрегация**) и его использование не влияет на функциональную работу класса. Второй случай, когда объект, который объявлен в классе, есть составной частью этого класса (**композиция**).

2.2. Отношение типа **uses** (класс «использует другой класс). В этом случае класс содержит программный код другого вложенного класса, к которому он имеет доступ.

Наследование

Наследование - механизм, позволяющий написать производный класс-потомок на основе уже существующего (родительского, базового) класса. В класс–потомок добавляются поля и методы и базового класса. Наследование бывает одиночным и множественным.

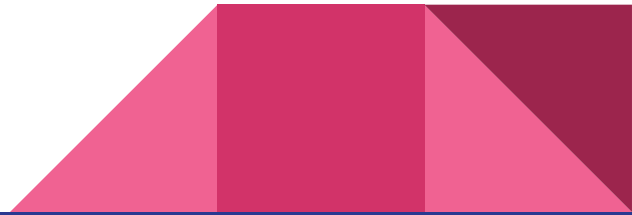
Одиночное наследование – вид наследования, при котором один потомок наследуется от одного родителя.

Множественное наследование – вид наследования, при котором один потомок наследуется от нескольких родителей.

Наследование как средство специализации.

Специализация наиболее очевидное применение наследования в ООП. Сосредоточив в классах, призванных быть базовыми, общие свойства группы связанных, но тем не менее различных объектов мы получаем возможность сравнительно быстро создавать конкретные классы, уточняя характеристики базового.

Специализация может заключаться как в добавлении новых методов и атрибутов (например, газета и газета с приложением), так и в переопределении уже имеющихся методов (например, иерархия классов-счетов).



Пример простейшего типа отношения is-a (наследование)

Суть отношения типа **is-a** состоит в том, что класс есть подвидом другого класса. В данном примере базовый класс ***Circle*** расширяется классом ***CircleColor***. Класс ***CircleColor*** есть подвидом класса ***Circle*** и добавляет к нему поле цвета.

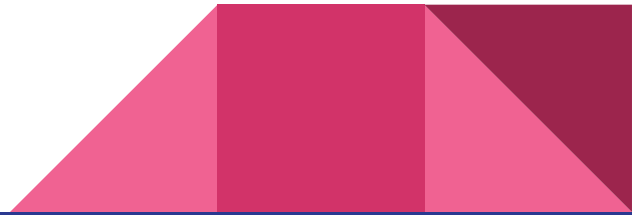
В классе **Circle** реализованы следующие элементы:

- внутренние скрытые (private) поля ***x, y, r***;
- конструктор ***Circle()*** с 3 параметрами, которые заполняют значения внутренних полей;
- методы доступа ***GetXYR()***, ***SetXYR()*** к полям класса;
- функция ***Area()*** вычисления площади круга.

Пример простейшего типа отношения is-a (наследование)

В производном классе ***CircleColor*** реализованы поля и методы, которые дополняют класс ***Color***:

- внутреннее скрытое поле класса ***color***;
- конструктор ***CircleColor()*** с 4 параметрами, который обращается к конструктору класса ***Color***;
- методы ***GetColor()***, ***SetColor()*** для доступа к скрытой переменной ***color***.



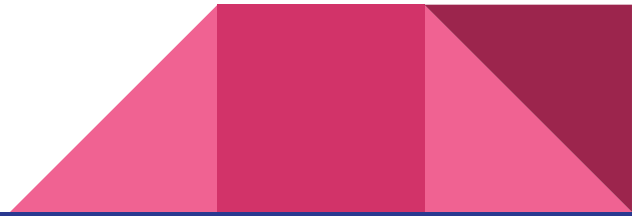
Отношения между классами типа has-a

При отношении **has-a** класс содержит один или несколько объектов (экземпляров) другого класса. Существуют два вида отношения **has-a**:

- **агрегация**. Это случай, когда один или несколько вложенных объектов не являются частью класса. Класс может содержать любое количество таких объектов (даже 0). Для лучшего изучения агрегации смотрите приведенные ниже примеры;
- **композиция**. В этом случае один или несколько вложенных объектов есть частью класса, то есть без этих объектов невозможно логическое существование самого класса.

Пример агрегации.

Класс ***BusStation*** (Автостанция) содержит массивы экземпляров классов ***Bus*** (Автобус), ***Car*** (Автомобиль). Количество элементов в массивах классов ***Bus*** и ***Car*** может изменяться. Даже, если на автостанции в данный момент времени не будет ни одного автобуса или автомобиля, автостанция будет функционировать. Это есть агрегация.



Пример композиции.

Класс ***Bike*** (Велосипед) содержит экземпляры классов ***Wheel*** (Колесо) и ***Saddle*** (седло), которые являются его составной частью. Велосипед не может быть без колес или седла, поэтому это есть композиция.