# Лабораторная работа № 3 – Наследование

**1 Цель работы**

Изучить основные понятия наследования классов в C# и получить представление о построении иерархии.

1. **Порядок выполнения работы**
   * прочесть краткие теоретические сведения;
   * выполнить задания раздела;
   * составить отчет по лабораторной работе и защитить его у преподавателя.
2. **Общая часть**

**3.1 Основные понятия**

Наследование, вместе с инкапсуляцией и полиморфизмом, является одной из трех основных характеристик (или базовых понятий) объектно-ориентированного программирования. Наследование позволяет создавать новые классы, которые повторно используют, расширяют и изменяют поведение, определенное в других классах. Класс, члены которого наследуются, называется базовым классом, а класс, который наследует эти члены, называется производным классом. Производный класс может иметь только один непосредственный базовый класс. Однако наследование является транзитивным. Если **сlass C** является производным от **сlass B**, и **сlass B** является производным от **сlass A**, **сlass C** наследует члены, объявленные в **сlass B** и **сlass A**. Основное назначение наследования заключается в построении иерархии объектов и повторном использовании кода.

## 3.2 Разработка производного класса

1) Базовым классом является класс Person, имеющий поля, конструктор и метод. Задача заключается в разработке производного класса Student.

class Person

{

public string Name;

public string Surname;

public Person(string name, string surname)

{

Name = name;

Surname = surname;

}

public void ShowInfo()

{

Console.WriteLine($"Имя: {Name} \nФамилия: {Surname}");

}

}

2) В окне Обозревателя Решений щелкнуть правой кнопкой по имени проекта, затем Добавить -> Класс. В появившемся окне ввести имя класса (Student) и нажать кнопку Добавить.

3) Чтобы класс Student являлся наследником класса Person, необходимо после имени класса через двоеточие указать имя базового класса:

class Student:Person

4) Зададим в классе Student дополнительное поле «Название группы» (GroupName):

class Student:Person

{

public string GroupName;

}

5) Для корректной работы программы также необходимо разработать конструктор, который использует ссылку на наследуемый класс. С этой целью используется ключевое слово **base**:

class Student:Person

{

public string GroupName;

public Student(string name, string surname, string groupName):base(name, surname)

{

GroupName = groupName;

}

}

Аналогичная конструкция будет использоваться при перегрузке конструктора.

## 3.3 Вывод информации и виртуальные методы

Когда базовый класс объявляет метод как **виртуальный**, производный класс может переопределить метод с помощью своей собственной реализации.

Пусть в классе Person метод ShowInfo будет виртуальным:

public virtual void ShowInfo()

Тогда, для использования в классе Student необходимо переопределить этот метод, добавив свойства класса-наследника:

public override void ShowInfo()

{

Console.WriteLine($"Имя: {Name} \nФамилия: {Surname} \nГруппа: {GroupName}");

}

Примечательно, что для вызова метода из базового класса, нам необходимо просто обратиться к нему с помощью ключевого слова **base**. Таким образом, переопределение метода может выглядеть так:

public override void ShowInfo()

{

base.ShowInfo();

Console.WriteLine($"Группа: {GroupName}");

}

1. **Общее задание для выполнения на паре**

Разработать базовый класс Товар (Product), реализующий общую сущность в интернет-магазине. Класс должен содержать конструктор, метод вывода информации о товаре, а также общие свойства для всех товаров:

- Идентификатор (id);

- Наименование;

- Цена.

Разработать класс-наследника, реализующий конкретный тип товаров (по своему усмотрению - одежда, обувь, электроника и т.д.) и содержащий дополнительные свойства (размер, цвет, диагональ экрана и пр.). Переопределить конструктор и метод вывода информации о товаре.

\*Добавить в базовый класс статический лист Корзина с типом Product и метод добавления текущего товара в корзину.

\*\*Реализовать метод, возвращающий суммарную цену товаров в корзине.

1. **Индивидуальные** **задания для выполнения работы**

В качестве базового класса использовать задание предыдущей лабораторной работы. В производный класс добавить несколько полей, свойств, методов, написать конструктор и перегрузить метод вывода.