

# ООП. Классы и Объекты

Объектно-ориентированное программирование

- Определения
- Переменные и методы класса
- Объекты и конструкторы
- Объекты и деструкторы
- Перегрузка функций и конструкторов
- Доступ к переменным класса
- Примеры
- Задачи

# Определения

- **Класс** - Именованная совокупность переменных и функций, определенная в соответствии с установленными правилами
- **Объект** — некоторая сущность в виртуальном пространстве, обладающая определённым состоянием и поведением, имеющая заданные значения свойств (атрибутов) и операций над ними (методов). Как правило, при рассмотрении объектов выделяется то, что объекты принадлежат одному или нескольким классам, которые определяют поведение (являются моделью) объекта. Термины «экземпляр класса» и «объект» взаимозаменяемы.

# Переменные и методы класса

# Определение класса в C#

```
public class Person {  
    // Field  
    private string name;  
    // Constructor  
    public Person()  
    {  
        name = "unknown";  
    }  
    // Method  
    public void SetName(string newName)  
    {  
        name = newName;  
    }  
}
```

- Классы
- Функции
- Свойства (Property)
- Переменные

- [public](#) Тип или член класса может быть доступен отовсюду.
- [private](#) Тип или член класса может быть доступен только из текущего класса.
- [protected](#) Тип или член класса может быть доступен из текущего класса или же наследованного класса.
- [internal](#) Тип или член класса может быть доступен только из кода данной сборки(?).



# Объекты и конструкторы

```
public class Person {  
    // Field  
    private string name;  
    // Constructor  
    public Person(string prm)  
    {  
        name = prm;  
    }  
    // Method  
    public void SetName(string newName)  
    {  
        name = newName;  
    }  
}
```

```
public class Person {  
    // Field  
    private string name;  
    // Constructor  
    public Person(Person prm)  
    {  
        name = prm.name;  
    }  
    // Method  
    public void SetName(string newName)  
    {  
        name = newName;  
    }  
}
```

# Деструкторы

- Отрабатывают при уничтожении класса
- Unmanaged деструкторы
- Managed деструкторы
- IDisposable

```
#include <stdio.h>
//Определение класса с именем Point
class Point {
    // Закрытые члены
private:
    double m_x, m_y;
// Интерфейс
public:
    Point(double x, double y);
    ~Point();
    // Другие методы...
};
```

- Garbage collector
- Memory Management

# Перегрузка функций и конструкторов



- Бинарные (сложение, вычитание) - математическая операция, принимающая два аргумента и возвращающая один результат (то есть с арностью два).
- Унарные (возведение в степень) - на множестве называется отображение множества в себя, которое каждому элементу множества, называемому операндом, ставит в соответствие некоторый элемент того же множества, называемый результатом.

- Сложение
- Вычитание
- Сравнение
- Инкремент (++)
- Декремент (--)

- Переопределение ToString()

```
public override string ToString()  
{  
    return String.Format("Name: {0}", name);  
}
```

## ■ Переопределение функций

```
public virtual string MyGenom()
```

```
{  
    return MyGenom("");  
}
```

```
public virtual string MyGenom(string knownSubType)
```

```
{  
    return "xy"+knownSubType;  
}
```

## ■ Принципы

- Старайтесь сделать один метод с большим количеством параметров, остальные пусть дефолтные значения базовому методу

# Доступ к переменным класса

- Автоматические свойства

```
public string Gender { get; set; }
```

- Классический подход к свойтвам

```
public string Name
```

```
{
```

```
    get { return name; }
```

```
    set { name = value; }
```

```
}
```

# Примеры

# Задачи



- **L1T1** Создать класс с именем Equation, объекты которого предназначены для хранения алгебраических уравнений второго порядка в виде  $ax^2+bx+c=0$ . Определить переменные класса, конструкторы, метод вычисления количества корней и метод отображения хранящейся в объекте информации.
- **L1T2** Создать класс с именем Person, объекты которого предназначены для хранения данных о конкретном человеке (можно ограничиться только фамилией и годом рождения). Определить переменные класса, конструкторы, деструктор и метод отображения хранящейся в объекте информации.

- **L1T3** Создать класс с именем `Polynom`, объекты которого предназначены для хранения алгебраических многочленов в виде  $a_0 + a_1x + \dots + a_nx^n$ . Определить переменные класса, конструкторы, деструктор и метод отображения хранящейся в объекте информации.
- **L1T4** Создать класс с именем `Fraction`, объекты которого предназначены для хранения рациональных чисел в виде правильных дробей. Определить переменные класса, конструкторы (выполняющие приведение дроби к правильному виду), деструктор и метод отображения хранящейся в объекте информации.