Туториал по переопределению метода ToString()

Доброго времени суток, уважаемые первокурсники программной инженерии!

Сегодня мы попытаемся разобрать строчку кода в программах   
public override string ToString()

Для чего нужно переопределение методов? Каждый класс, который был когда-либо написан и будет написан на C#, наследуется от класса object. В классе object определен метод ToString(), который приводит все объекты классов к строке. По умолчанию, метод ToString() возвращает строку, которая содержит название класса и название пространства имен. Так происходит из-за того, что одна из перегрузок метода ToString() принимает на вход объект типа object, а все классы неявным образом приводятся к типу object.

Рассмотрим пример:

using System;

namespace Tutorial

{

class Man

{

string name;

int year;

public Man(int year, string name)

{

this.year = year;

this.name = name;

}

}

class Program

{

static void Main(string[] args)

{

Man man = new Man(10, "Mark");

Console.WriteLine(man);

}

}

}

Данная программа выведет: “Tutorial.Man”

Для того, чтобы вывести на экран с помощью Console.WriteLine(man);  
всю информацию об объекте, мы должны переопределить метод ToString()

Переопределять мы можем некоторые методы из классов-родителей, помеченные модификатором virtual, но об этом вы поговорите подробнее на следующих семинарах.

Переопределим данный метод следующем образом для класса Man:

public override string ToString()

{

return $"{year}, {name}";

}

И тогда после запуска предыдущего кода, мы получим на экране “10 Mark”.

Будьте аккуратны с переопределением методов! При явном привидении типа к классу-родителю, переопределенные методы будут реализовывать методы из класса-наследника. Рассмотрим следующий пример

using System;

namespace Tutorial

{

class Man

{

string name;

int year;

public Man(int year, string name)

{

this.year = year;

this.name = name;

}

public override string ToString()

{

return $"{year}, {name}";

}

}

class Program

{

static void Main(string[] args)

{

Man man = new Man(10, "Mark");

Console.WriteLine(man);

object obj = man;

Console.WriteLine(obj);

object obj2 = new object();

Console.WriteLine(obj2);

}

}

}

Мы неявно привели объект класса Man к классу object. Но после того, как мы вывели на экран объект obj, мы получили то, что должны получить при выводе man. При выводе obj2 мы получим “System.object”.