<epam>

Обьектно Ориентированное Программирование

C# для ООП

Кто я такой:

Китар Роман Юрьевич

Email:

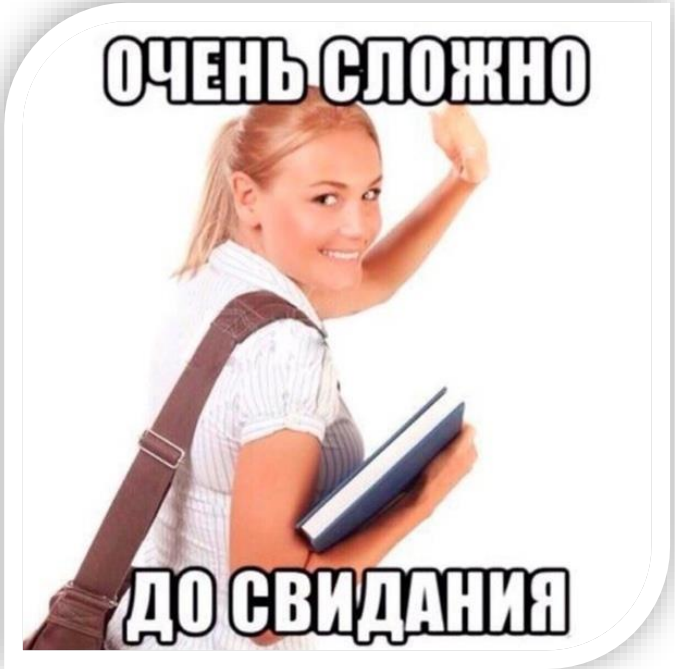
Опыт:

- 1 год девелопер, работаю на американском проекте.
- Учился в МФТИ(ГУ) (физтех)
- 3 раза побеждал в областных олимпиадах по математике, несколько раз по физике и один раз по химии.
- 4 года администрирование производства (металлообработка)

Что такое парадигма программирования?

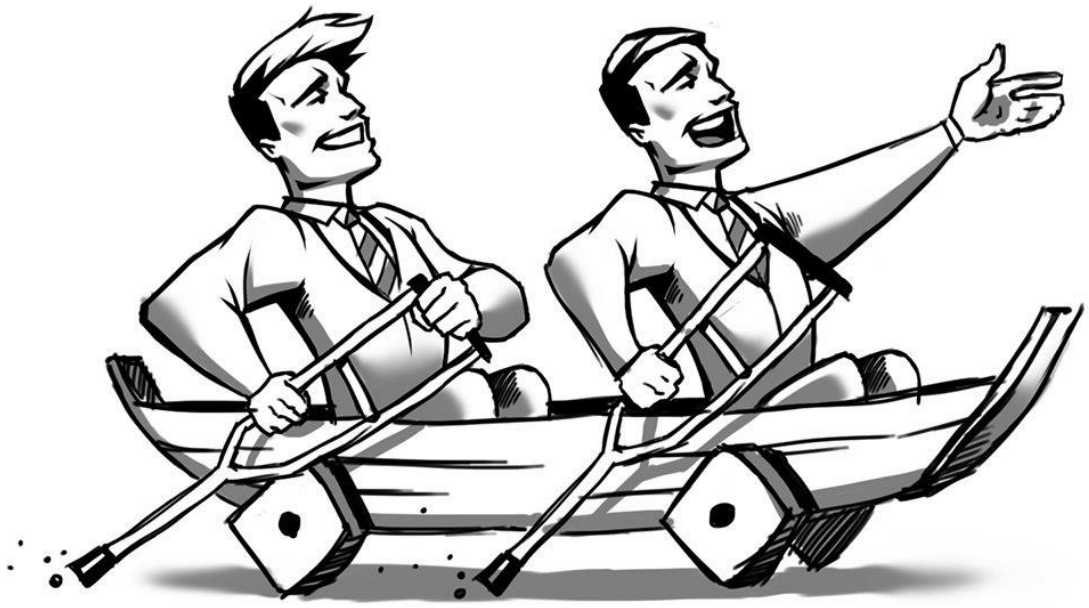
И ПРИЧЕМ ТУТ МЫ

- **Парадигма** - набор концепций или шаблонов мышления, включая теории, методы исследования, постулаты и стандарты, в соответствии с которыми осуществляются последующие построения, обобщения и эксперименты в области
- **Парадигма -** совокупность достижений, признаваемых всем научным сообществом в тот или иной период времни и служащая основой дальнейших исслодований
- **Парадигма** – общепризнанный образец или пример того, как наданном этапе развития стоит подходить к решению проблем в данной области.



Что же такое парадигма программирования?

И ЧТО НАМ С НЕЙ ДЕЛАТЬ



Что такое парадигма программирования?

И ПРИЧЕМ ТУТ, ВСЕ ТАКИ, МЫ?

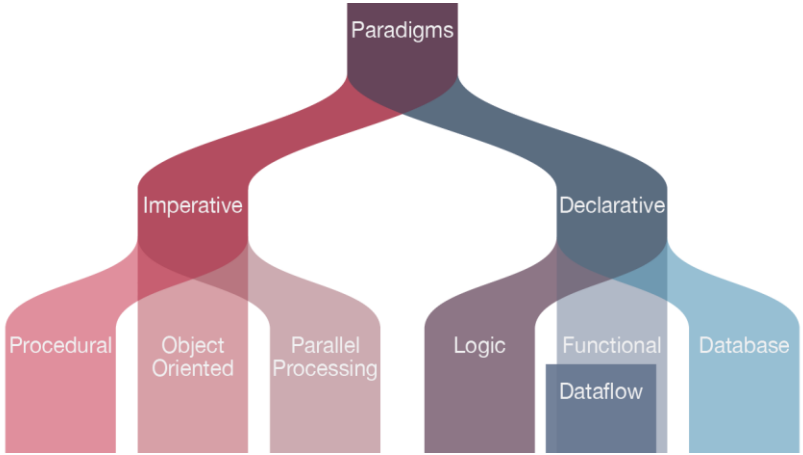
Парадигма - подход к решению задачи.

Парадигмам необходимо следовать.

А какие вообще бывают парадигмы программирования?

И ЭТО ВСЕ НАДО ЗНАТЬ?

- **Императивное программирование**
- **Декларативное программирование**
- **Структурное программирование**
- **Функциональное программирование**
- **Логическое программирование**
- **Объектно-ориентированное программирование (ООП)**
- **И многие, многие другие**



В чем заключается ООП?

ШЕСТЬ ПРИНЦИПОВ АЛАНА КЕЯ

1. Все является объектом
2. Каждый объект является экземпляром класса
3. Класс определяет поведение объекта
4. Классы организованы в иерархию наследования
5. Каждый объект обладает независимой памятью
6. Вычисления производятся путем взаимодействия между объектами

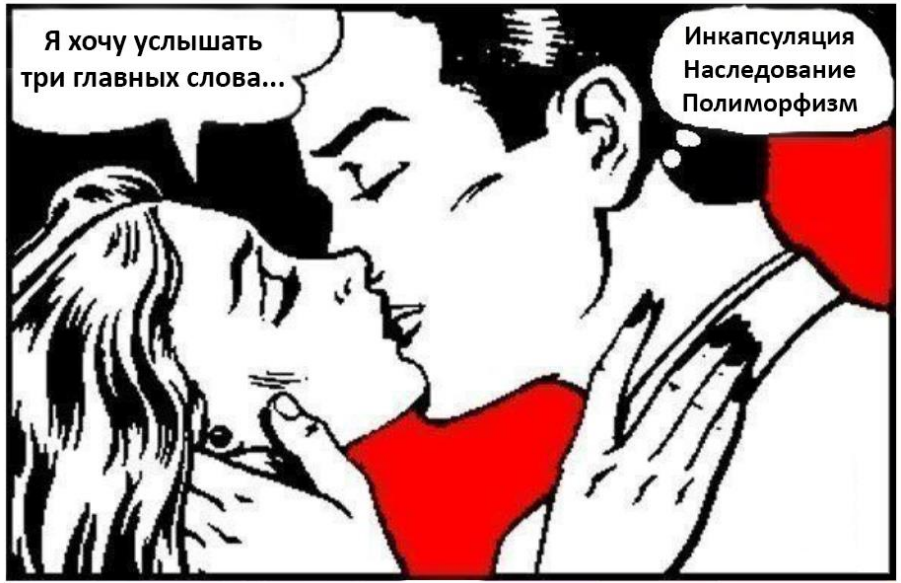
Алан Кей
Создатель ООП
Автор языка Smalltalk

В чем заключается ООП?

ОСНОВНЫЕ КОНЦЕПЦИИ

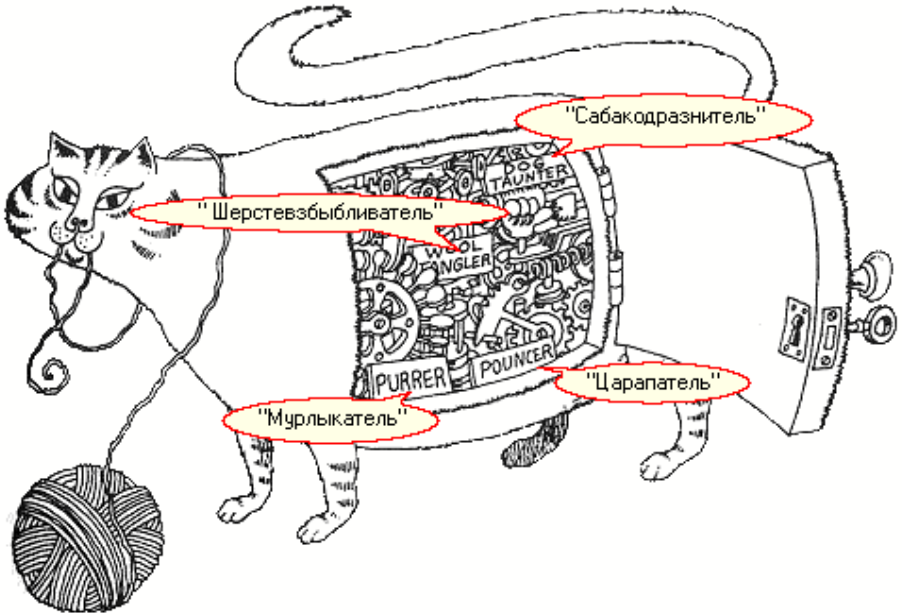
- ИНКАПСУЛЯЦИЯ
- НАСЛЕДОВАНИЕ
- ПОЛИМОРФИЗМ

- АБСТРАКЦИЯ



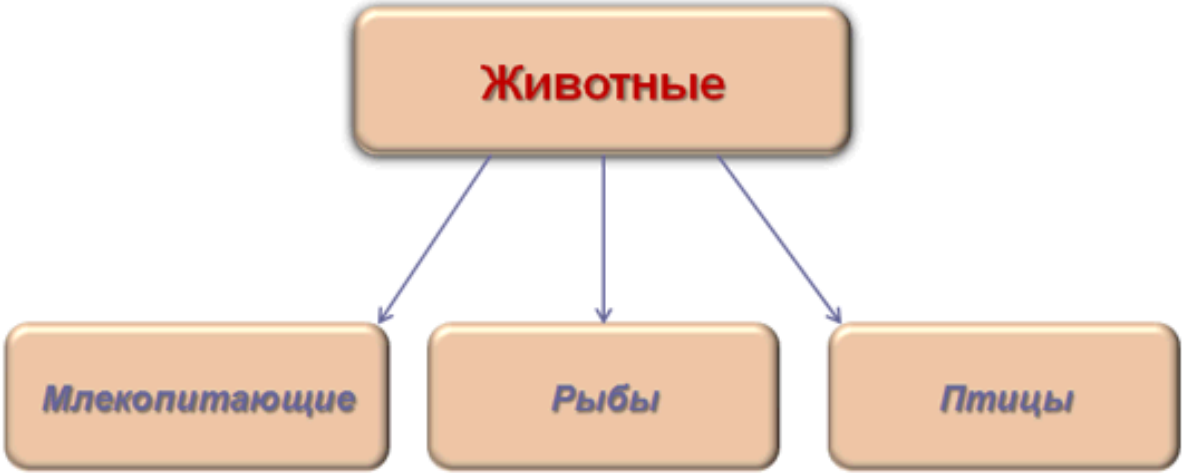
В чем заключается ООП?

ИНКАПСУЛЯЦИЯ



В чем заключается ООП?

НАСЛЕДОВАНИЕ



В чем заключается ООП?

ПОЛИМОРФИЗМ

В чем заключается ООП?

АБСТРАКЦИЯ

Так почему же ООП?

КАКИЕ ПРОБЛЕМЫ ОН РЕШАЕТ?

- Снабжение данных смыслом
- Контроль использования данных
- Способность к расширению программ

В чем заключается ООП?

КЛАССЫ

```
1  namespace OOP_essentials
2  {
       2 references
3      public class User
4      {
5          public string Name;
6          public string Surname;
7      }
8  }
9
```

```
1  namespace OOP_essentials
2  {
       0 references
3      class Program
4      {
           0 references
5          static void Main(string[] args)
6          {
7              User user = new User();
8          }
9      }
10 }
11
```

В чем заключается ООП?

ЧЛЕНЫ КЛАССА

- Конструкторы
- Поля
- Свойства
- Методы
- …

КОНСТРУКТОРЫ КЛАССА

```
3 references
public class User
{
    public string Name;
    public string Surname;

    1 reference
    public User(string name, string surname)
    {
        Name = name;
        Surname = surname;
    }
}
```

```
0 references
class Program
{
    0 references
    static void Main(string[] args)
    {
        User user = new User(name:"John", surname:"Doe");
    }
}
```

ООП в C#

ИНКАПСУЛЯЦИЯ

```
0 references
static void Main(string[] args)
{
    User user = new User(name: "John", surname: "Doe");
    user.Name = "Bill";
}
```

```
public class User
{
    private string _name;
    private string _surname;

    0 references
    public string GetName()
    {
        return _name;
    }

    0 references
    public string GetSurname()
    {
        return _surname;
    }
```

ООП в C#

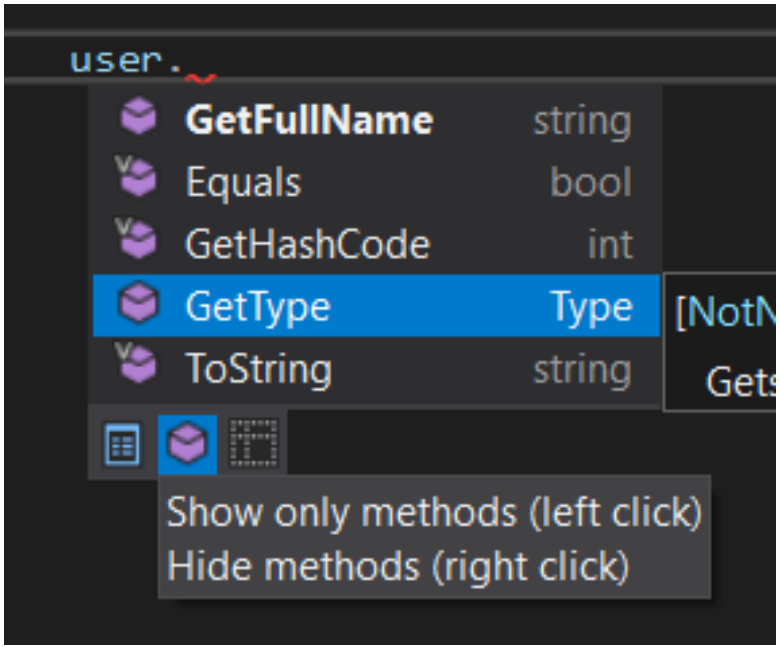
СВОЙСТВА

```
2 references
public class User
{
    1 reference
    public string Name { get; private set; }
    1 reference
    public string Surname { get; }

    1 reference
    public User(string name, string surname)
    {
        Name = name;
        Surname = surname;
    }
}
```

```
0 references
static void Main(string[] args)
{
    User user = new User(name: "John", surname: "Doe");
    user.Name = "Bill";
}
```

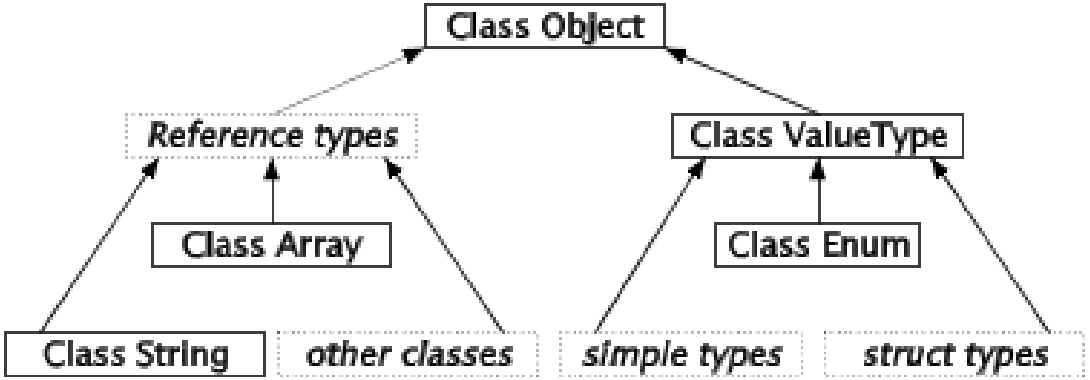
МЕТОДЫ

```
0 references
public string GetFullName()
{
    return Name + " " + Surname;
}
```



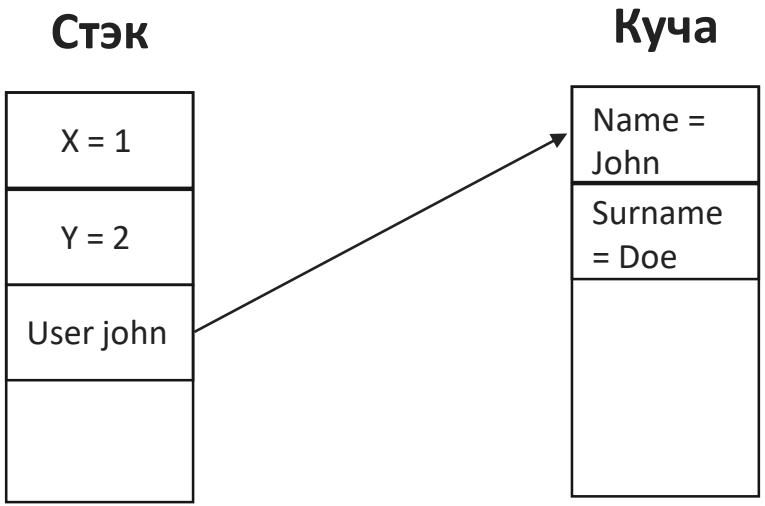
В чем заключается ООП?

ИЕРАРХИЯ КЛАССОВ



В чем заключается ООП?

ЗНАЧИМЫЕ И ССЫЛОЧНЫЕ ТИПЫ



В чем заключается ООП?

ЗНАЧИМЫЕ И ССЫЛОЧНЫЕ ТИПЫ

```
0 references
static void Main(string[] args)
{
    int x = 1;
    int y = x;
    y = 2;
    Console.WriteLine("x = " + x);
    Console.WriteLine("y = " + y);
}
```

```
x = 1
y = 2
```

В чем заключается ООП?

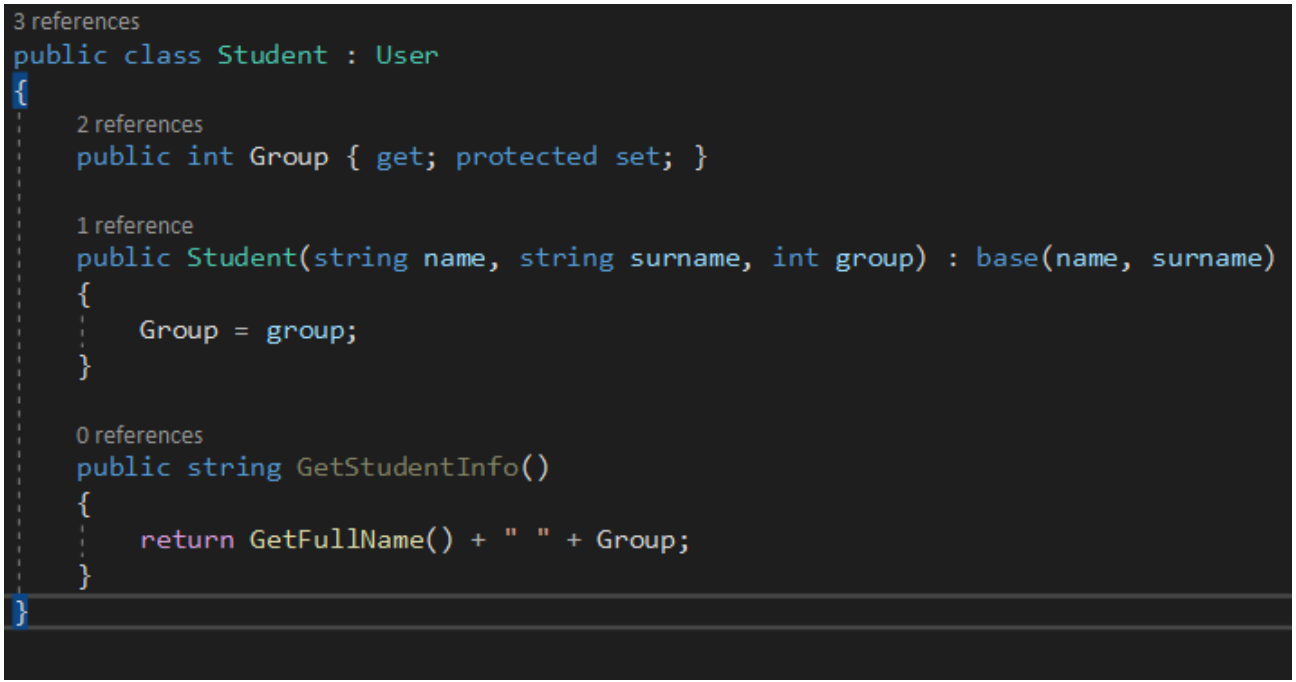
ЗНАЧИМЫЕ И ССЫЛОЧНЫЕ ТИПЫ

```
User john = new User(name: "John", surname: "Doe");
User bill = john;
bill.Name = "Bill";
Console.WriteLine("John = " + john.Name);
Console.WriteLine("Bill = " + bill.Name);
```

```
John = Bill
Bill = Bill
```

В чем заключается ООП?

НАСЛЕДОВАНИЕ



```
3 references
public class Student : User
{
    2 references
    public int Group { get; protected set; }

    1 reference
    public Student(string name, string surname, int group) : base(name, surname)
    {
        Group = group;
    }

    0 references
    public string GetStudentInfo()
    {
        return GetFullName() + " " + Group;
    }
}
```

В чем заключается ООП?

ПЕРЕОПРЕДЕЛЕНИЕ

```
4 references
public virtual string GetInfo()
{
    return Name + " " + Surname;
}
```

```
4 references
public override string GetInfo()
{
    return base.GetInfo() + " " + Group;
}
```

```
John Doe
Peter Nash 1
```

В чем заключается ООП?

АБСТРАКТНЫЕ КЛАССЫ

```
4 references
public abstract class Shape
{
    4 references
    public abstract void Draw();
}
```

В чем заключается ООП?

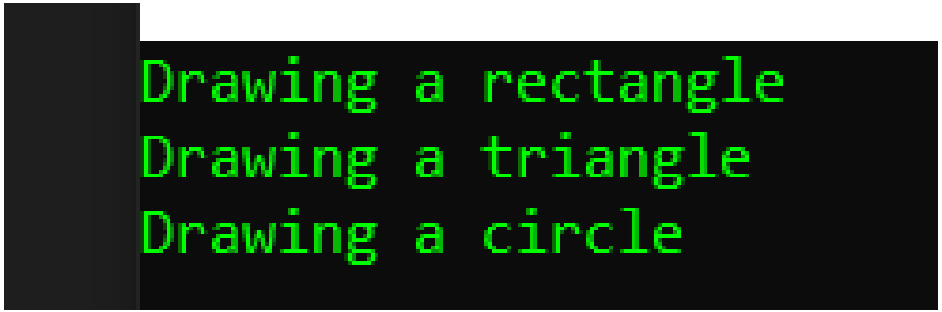
ПОЛИМОРФИЗМ

```
4 references
public class Shape
{
    4 references
    public virtual void Draw() { }
}

1 reference
public class Circle : Shape
{
    4 references
    public override void Draw()
    {
        Console.WriteLine("Drawing a circle");
    }
}
```

```
var shapes = new List<Shape>
{
    new Rectangle(),
    new Triangle(),
    new Circle()
};

foreach (var shape in shapes)
{
    shape.Draw();
}
```

В чем заключается ООП?

ПОЛИМОРФИЗМ



Что в итоге?

C# СОЗДАН ДЛЯ ООП

Для компонентно-ориентированного программирования и для всей индустрии в целом чрезвычайно важно встроить в языки программирования поддержку концепции компонента. Это и была одна из ключевых целей создания C#. – Андерс Хейльсберг. Создатель языка C#.