

Богатов Р.Н.

Программирование на языке высокого уровня

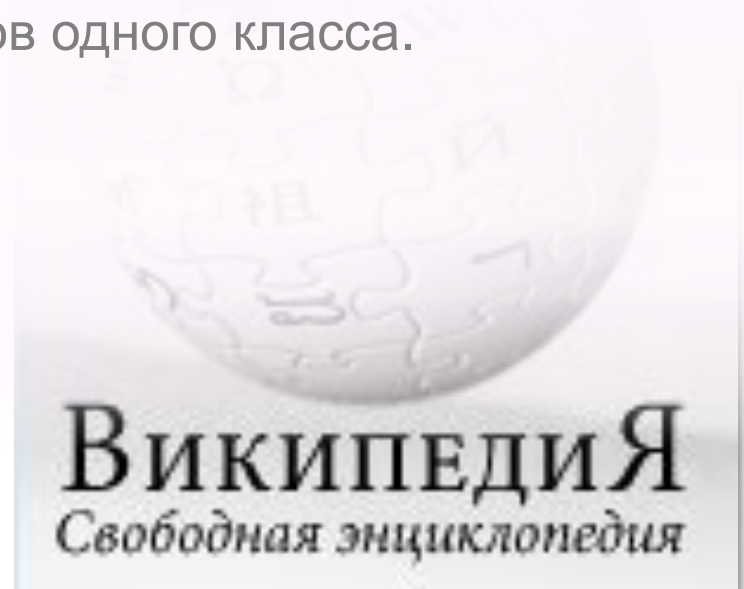
Лекция 11.

Основы объектно-ориентированного программирования

Кафедра АСОИУ ОмГТУ, 2012

Основные понятия ООП – оп ооп ☺

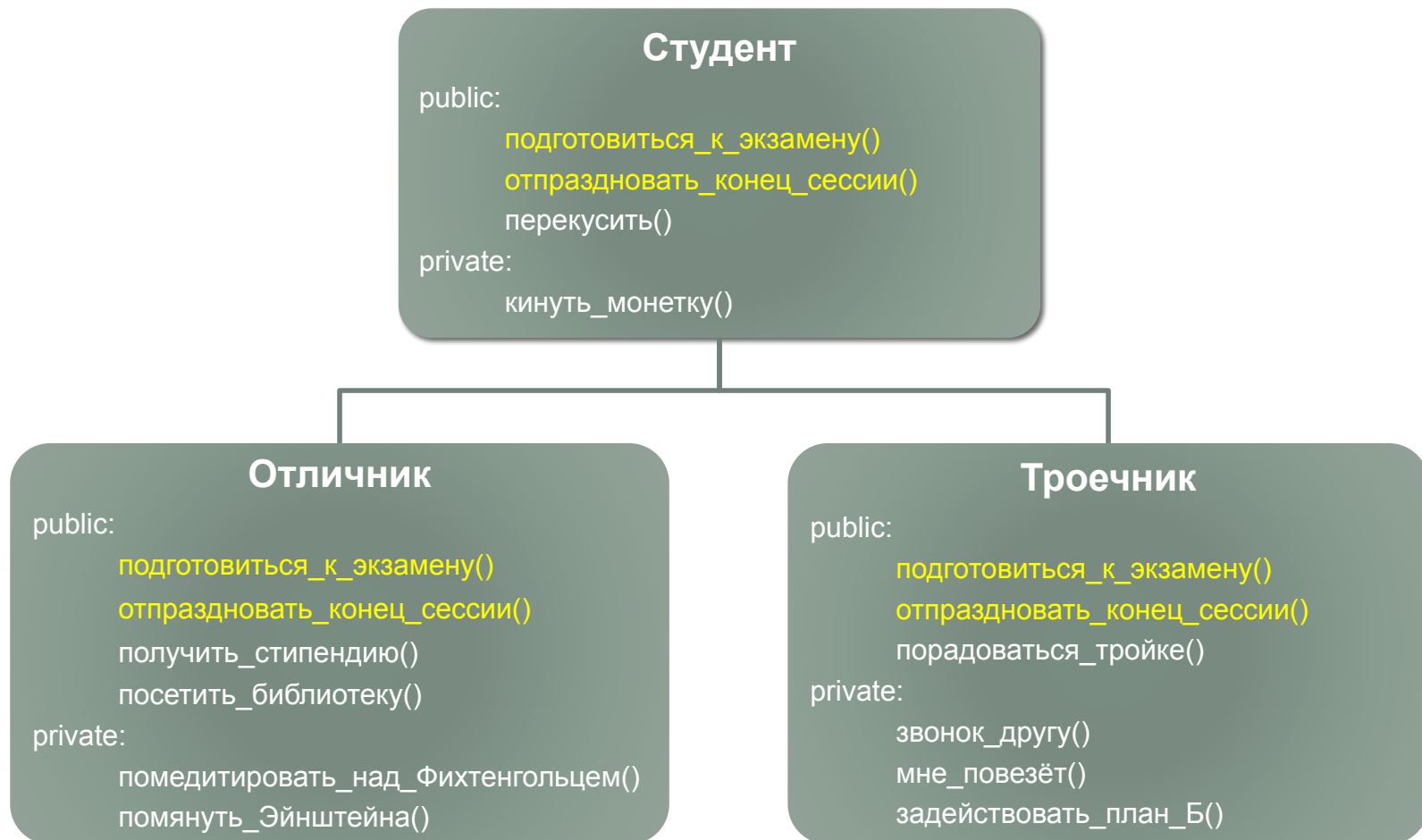
- **Класс** — описание устройства и поведения объектов. Класс можно сравнить с чертежом, согласно которому создаются объекты.
- **Объект** — сущность в памяти компьютера, появляющаяся при создании экземпляра класса. Одновременно может существовать несколько экземпляров одного класса.



Основные понятия ООП

- **Инкапсуляция** — объединение данных и кода в единый объект с целью 1) их защиты и 2) сокрытия реализации объекта.
- **Наследование** — возможность порождать один класс от другого 1) с сохранением свойств и методов базового (родительского) класса и 2) с добавлением, при необходимости, новых свойств и методов.
- **Полиморфизм** — изменение реализации метода базового класса в классах-потомках

Пример иерархии и полиморфизма



Если бы был класс для рациональных дробей...

```
РацДробь class РацДробь
A = new {
B = new
...
A.Числитель
A.Знаменатель
...

// Вычисление произведения
E = B.Y * A.X;
E = A.Y * B.X;
E = E.X * B.Y;

// E = (D);
label3:
    Console.WriteLine("Произведение: " + E);
}

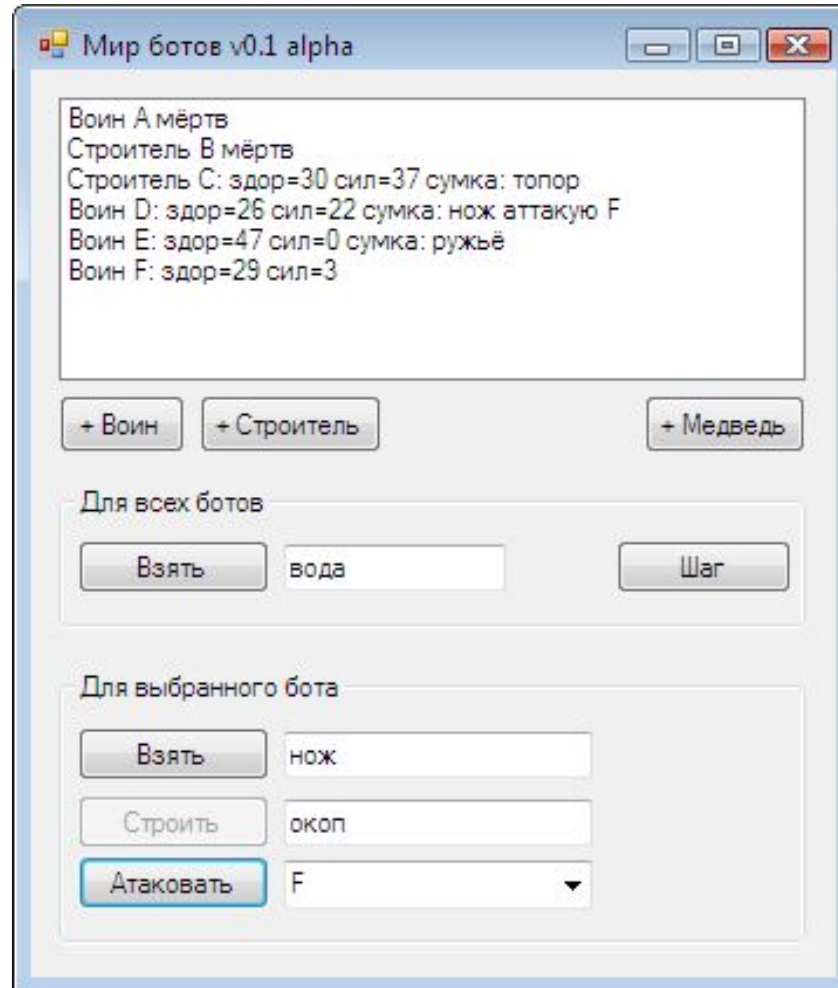
...
}

class РацДробь
{
    ...
    public РацДробь(ulong num, ulong den)
    {
        Числитель = num;
        Знаменатель = den;
    }

    public static РацДробь operator +(РацДробь x, РацДробь y)
    {
        РацДробь t = new РацДробь(
            x.Числитель*y.Знаменатель + y.Числитель*x.Знаменатель,
            x.Знаменатель*y.Знаменатель);
        t.Сократить();
        return t;
    }

    public override string ToString()
    {
        Сократить();
        string s = Convert.ToString(Числитель);
        if (Знаменатель!=1)
            s += "/" + Convert.ToString(Знаменатель);
        return s;
    }
    ...
}
```

Мир ботов v0.1 alpha



Программируем мир ботов

```
abstract class Bot
```

```
class Builder: Bot
```

```
class Warrior: Bot
```

```
public partial class Form1 : Form
```

```
{
```

```
...
```

```
// команда боту "Использовать"
```

```
private void buUtilize_Click(object sender, EventArgs e)
```

```
{
```

```
    // i - индекс избранного бота
```

```
    // tbUtilize.Text - что нужно ему дать
```

```
    Bots[i].Utilize( tbUtilize.Text );
```

```
    ... // обновить информацию о ботах
```

```
}
```

```
// команда "Строить"
```

```
private void buBuild_Click(object sender, EventArgs e)
```

```
{
```

```
    ... // проверить, что i-й бот - строитель
```

```
    Bots[i].Build( tbBuilding.Text );
```

```
    ... // обновить информацию о ботах
```

```
}
```

```
...
```

```
}
```

```
}
```

```
base.MakeStep();
```

```
}
```

```
0;
```

```
)
```

```
-= 10;
```

```
-= 3;
```