Daniela Nguyenová

2. OOP, třídy, objekty, veřejné a neveřejné atributy, metody, konstruktor, statické třídy a metody. Příklad.

OOP – objektově orientované programování

Skládá se z celků, které lze rozšiřovat, upravovat, používat. Nemusíme vědět, co se děje uvnitř pouze musíme rozvržení celků správně navrhnout. Snažíme se simulovat reálný svět.

Třídy

Definuje obecné vlastnosti a chování objektů. Třída může být chápána jako šablona nebo vzor pro vytváření konkrétních instancí, které jsou známy jako objekty.

Objekty

Základní jednotka OOP.
Instance jsou konkrétní objekty vytvořené na základě třídy. Každá instance má svůj vlastní stav, který je určen hodnotami atributů.

Veřejné a neveřejné atributy

Atributy nebo metody, které jsou označeny jako ***public***, jsou přístupné ze všech částí programu. Můžete je volat z libovolného místa ve vašem programu nebo dokonce z jiných knihoven.

Atributy nebo metody, které jsou označeny jako ***private***, jsou přístupné pouze uvnitř této třídy. Nikdo mimo tuto třídu nemůže přímo přistupovat k soukromým členům.

```
public class Auto
{
    private string interniInformace; // Soukromý atribut
    public string Barva { get; set; } // Veřejný atribut
}
```

Metody

Metody jsou funkce definované uvnitř třídy, které popisují chování objektu. Tyto metody mohou pracovat s atributy třídy a provádět různé operace. Vytváříme je jako podprogramy.

Konstruktor

Konstruktor je speciální metoda volaná při vytváření nové instance třídy. Slouží k inicializaci (přiřazení počátečních hodnot) atributů objektu.

Statické třídy a metody

Mohou být volány přímo pomocí názvu třídy, bez nutnosti vytváření instance této třídy.

Daniela Nguyenová

Statické třídy nelze instancovat pomocí klíčového slova ***new***. Jsou navrženy tak, aby obsahovaly pouze statické metody, statické vlastnosti a statické události. Jsou často používány k poskytování funkcí nebo služeb, které nemají žádný vnitřní stav a nevyžadují vytváření instance.
Statické metody patří k třídě, nikoliv k instanci této třídy. Mohou být volány přímo z názvu třídy bez nutnosti vytváření instance. Nemají přístup k instančním členům třídy (nestatickým členům) a nemohou pracovat s instančními proměnnými. Nemohou být volány na instanci třídy a naopak nestatické metody nemohou být volány pomocí názvu třídy, ale pouze pomocí instance této třídy.

```
public static class MathUtility
{
    // Statická metoda pro výpočet druhé mocniny čísla
    public static double Square(double number)
    {
        return number * number;
    }
}

class Program
{
    static void Main()
    {
        // Použití statické metody
        double result = MathUtility.Square(5);

        Console.WriteLine($"Číslo 5 umocněné na druhou je: {result}");
    }
}
```

- ***MathUtility*** je statická třída obsahující statickou metodu ***Square***, která vypočítá druhou mocninu čísla.
- V metodě ***Main*** v programu je volána statická metoda ***Square*** z třídy ***MathUtility*** bez nutnosti vytváření instance této třídy.
- Výsledek výpočtu je poté vypsán na konzoli.

Příklad

```
class Auto
{
    // Atributy (vlastnosti)
    public string Barva { get; set; }
    public string Model { get; set; }

    // Konstruktor
    public Auto(string barva, string model)
    {
        Barva = barva;
        Model = model;
    }

    // Metoda
    public void Zrychlit()
    {
        Console.WriteLine($"{Model} zrychluje!");
    }
}
class Program
{
    static void Main()
    {
        // Vytvoření instance třídy Auto
        Auto auto1 = new Auto("modré", "sedan");
        Auto auto2 = new Auto("červené", "SUV");

        // Volání metody
        auto1.Zrychlit();
        auto2.Zrychlit();
    }
}
```

V tomto příkladu:

- Třída ***Auto*** má dvě vlastnosti (***barva a model***), konstruktor, který inicializuje tyto vlastnosti, a metodu **Zrychlit**.

Daniela Nguyenová

- Vlastnosti jsou definovány pomocí tzv. ***auto-implemented properties ({ get; set; })***, které automaticky vytvářejí privátní pole pro ukládání hodnoty.
- Konstruktor ***Auto*** je volán při vytváření nové instance třídy.
- V metodě ***Main*** jsou vytvořeny dvě instance třídy ***Auto*** a zavolána je metoda ***Zrychlit()*** pro každý objekt.