
Konstruktor

```
class Pes
{
    public string Jmeno;
    public int Vaha, Vyska;
    public long KolikLidiPokousal;
    public Pes(string jmeno, int vaha, int vyska, long kolikLidiPokousal)
    {
        this.Jmeno = jmeno;
        this.Vaha = vaha;
        this.Vyska = vyska;
        this.KolikLidiPokousal = kolikLidiPokousal;
    }
    public void Stekni()
    {
        Console.WriteLine("haf! jmenuju se {0} a uz jsem pokousal {1} lidi!", Jmeno,
        KolikLidiPokousal);
    }
}

...
Pes pes = new Pes("alík", 5, 20, 2);
pes.Stekni();
```

Dědičnost - odvozená nová třídy

```
class Kocka
{
}

class Tygr : Kocka
{
}

class VelkyPes : Pes
{
    public VelkyPes(string jmeno, int vaha, int vyska, long kolikLidiPokousal)
        : base(jmeno, vaha, vyska, kolikLidiPokousal+20)
    {
    }
}
```

```
class Kocka
{
}

class Tygr : Kocka
{
}

class VelkyPes : Pes
{
    public VelkyPes(string jmeno, int vaha, int vyska, long kolikLidiPokousal)
        : base(jmeno, vaha, vyska, kolikLidiPokousal)
    {
    }
}
```

```

        public void Stekni()
        {
            Console.WriteLine("HAF! HAF! Uz jsem pokousal {1} lidi!",
                Jmeno, KolikLidiPokousal+1000);
        }
    }
}
-----

```

Problém s předefinovanou metodou

```

class Pes
{
    ...
    public void Stekni2x()
    {
        Stekni();Stekni();
    }
}

```

Obyčejná a virtuální metody

```

class Pes
{
    ...
    public virtual void Stekni()
    {
        Console.WriteLine("haf! jmenuju se {0} a uz jsem pokousal {1} lidi!", Jmeno,
KolikLidiPokousal);
    }
    ...
}

class VelkyPes: Pes
{
    ...
    public override void Stekni()
    {
        Console.WriteLine("HAF! HAF! Uz jsem pokousal {1} lidi!",
            Jmeno, KolikLidiPokousal+1000);
    }
    ...
}

```

Polymorfismus:

```

Pes pes;

for (int i = 1; i <= 2; i++)
{
    if (i==1)
        pes = new Pes("alík", 5, 20, 0);
    else
        pes = new VelkyPes("Hugo", 5, 20, 57);

    pes.Stekni();
}

```

Abstraktní třída - příklad

```

abstract class Zvire
{

```

```

        public string jmeno;
        public abstract void VydejZvuk();
        public void Init(string jmeno)
        {
            this.jmeno = jmeno;
        }
    }

class Tygr: Zvire
{
    public Tygr(string jmeno)
    {
        Init(jmeno);
    }
    public override void VydejZvuk()
    {
        Console.WriteLine("{0}: Uaaaaaa!", jmeno);
    }
}

class Ara : Zvire
{
    public Ara(string jmeno)
    {
        Init(jmeno);
    }
    public override void VydejZvuk()
    {
        Console.WriteLine("{0}: Dobrrrrraaa prrrraaaace!!", jmeno);
    }
}

class Ryba : Zvire
{
    public Ryba(string jmeno)
    {
        Init(jmeno);
    }
    public override void VydejZvuk()
    {
        Console.WriteLine("{0}: ...", jmeno);
    }
}

class Had : Zvire
{
    public Had(string jmeno)
    {
        Init(jmeno);
    }
    public override void VydejZvuk()
    {
        Console.WriteLine("{0}: Sssssssss...", jmeno);
    }
}

...
Zvire[] zoo = new Zvire[10];
zoo[0] = new Tygr("Jonatan");
zoo[1] = new Ara("Lora");
zoo[2] = new Ryba("Pepik");
zoo[3] = new Ryba("Julie");
zoo[4] = new Ara("Flint");
zoo[5] = new Tygr("Samba");

for (int i=0;i<=5;i++)

```

```
zoo[i].VydejZvuk();
```

Static members (metody, data...) patří k třídě a ne instanci.

```
abstract class Zvire
{
    public static int Pocet = 0;
    public Zvire()
    {
        Pocet++;
    }
    ...
}
...

for (int i = 0; i < Zvire.Pocet; i++)
    zoo[i].VydejZvuk();
```

Zapouzdlení

```
abstract class Zvire
{
    private static int Pocet = 0;
    public Zvire()
    {
        Pocet++;
    }
    public string jmeno;
    public abstract void VydejZvuk();
    public void Init(string jmeno)
    {
        this.jmeno = jmeno;
    }
    public static int PocetVsechZvirat()
    {
        return Pocet;
    }
}
...

for (int i = 0; i < Zvire.PocetVsechZvirat(); i++)
    zoo[i].VydejZvuk();
```

Properties - vlastnosti

```
class Zvire
{
    ...
    private int pocetNoh;
    public int PocetNoh
    {
        get
        {
            return pocetNoh;
        }
        set
        {
            if ((value == 0) || (value == 2) || (value == 4) || (value == 6) || (value
== 8) || (value == 100))
            {
                pocetNoh = value;
            }
        }
    }
}
```

```
}  
...  
    Zvire z = new Ryba("pokusna");  
    z.PocetNoh = 4;  
    Console.WriteLine("PocetNoh: {0}", z.PocetNoh);  
    z.PocetNoh = 2;  
    Console.WriteLine("PocetNoh: {0}", z.PocetNoh);  
    z.PocetNoh = 7;  
    Console.WriteLine("PocetNoh: {0}", z.PocetNoh);  
    z.PocetNoh = 8;  
    Console.WriteLine("PocetNoh: {0}", z.PocetNoh);  
    z.PocetNoh = 1;  
    Console.WriteLine("PocetNoh: {0}", z.PocetNoh);  
    z.PocetNoh = 0;  
    Console.WriteLine("PocetNoh: {0}", z.PocetNoh);  
  
    Console.WriteLine();  
    return;  
...  
-----
```