

Na začátek...

DateTime

TimeSpan

měření času

String vs StringBuilder

Delegáty

```
public delegate int Funkce(int x, int y);
```

```
public static int Soucet(int x, int y)  
{ return x + y; }
```

```
public static int Soucin(int x, int y)  
{ return x * y; }
```

```
Funkce f = Soucet;  
Console.WriteLine(f(10, 20));  
Funkce f = Soucin;  
Console.WriteLine(f(10, 20));
```

Delegáty 2

```
static void Vypis(int x, int y, Funkce f)
{
    Console.WriteLine ("f({x},{y})={f(x,y)}");
}
```

...

```
Vypis(11, 22, Soucet);
Vypis(11, 22, Soucin);
```

Delegáty 3

```
public delegate void Tisk(string zprava);
```

```
Tisk t = Jednou;
```

```
Tisk t += Jednou; // Click
```

```
Tisk t -= Jednou;
```

Delegáty lze dosazovat, přidávat, ubírat i předávat jako parametry funkcím.

Anonymní funkce

```
t = delegate( string s )  
    { Console.WriteLine("Anonym: {0}:{0}", s); };  
  
Vypis(11, 22, delegate (int x, int y){ return x*y-10; });
```

Lambda funkce / lambda notace

```
Vypis(11, 22, (x,y) => x * y - 10 );
```

Neuvádí se typy, takže univerzálnější...

Vlákna

```
Thread th = new Thread(VytvorNaplnASetridPole);  
th.Start();
```

```
Thread th = new Thread(delegate(){ ... });  
th.Start();
```

```
(new Thread(delegate(){ ... })).Start();
```

`Environment.ProcessorCount`

Synchronizace, problémy

Race-condition

Zámky

```
lock (a)
{
    zbyva++;
}
```

Deadlock...

Diskrétní simulace 2

Jiný model:

- jaké procesy
- jaké události
- jak budou procesy reagovat na události

Příklad: obchodní dům...

oop/obchod.txt

oop/obchod-data.txt

KONEC

Hygiena programování (praktické tipy)

1. Oddělovat různé činnosti

1. analýza
 2. design
 3. GUI design
 4. kódování
 5. texty, grafika, zvuky
 6. testování (psaní testů)
 7. ladění
 8. dokumentace
- resources, šablony výstupů...

2. Pracovní deník

1. ujasnění CO budu dělat teď
2. rozhovor (kačenka)
3. záznam pro soustředění na jednu věc
(viz ToDo-list)
4. pozdější dohledání (...a viz verse)
5. sledování, kolik času mi co trvá

3. ToDo-list, bug-list, issue tracker

- * //TODO ve zdrojovém kódu
- * deník, konec nebo vyhledatelná značka
- * samostatný soubor
- * software (issue tracker, GitHub, Trello)

4. Správa verzí

- * CSV, SVN (Subversion),
Hg (Mercurial), GiT
- * možnost návratu k minulým verzím
- * sledování rozdílů

I při práci na jednom počítači.

I pro ne-programátorské projekty.

5. Automatické testy, TDD

* unit-testy

1. Napsat testy
2. Spustit testy a ujistit se, že selžou
3. Napsat kód
4. Ladit kód, dokud nesplní testy
5. Refaktorovat kód
6. Opakovat

* testování funkcionalit

* výkonnostní testy

* regresní testy

i dnes funguje, co fungovalo včera