

Práce s náhodnými čísly

Úvod

V této kapitole si ukážeme, jak generovat náhodná čísla pomocí třídy `Random`.

Problematika náhodných čísel v IT

Jedna z věcí, kterou počítač neumí udělat, je právě náhodné číslo. Algoritmy na generování takovýchto čísel dokážou vytvořit dlouhou sekvenci čísel, které pouze vypadají náhodně. Náhodná čísla se v programech generují podle tzv. `seedu`. Když nastavíme seed na jednu hodnotu, dostaneme vždy stejnou náhodnou sekvenci.

Třída `Random` a její metody

Abychom mohli začít generovat čísla, musíme nejdříve vytvořit novou instanci třídy `Random`.

Vytvoření instance

```
// Vytvoření generátoru náhodného čísla s "náhodným" seedem
Random r = new Random();

// Vytvoření generátoru n. č. se specifickým seedem
Random r2 = new Random(123);
```

Když neurčíme seed při vytvoření generátoru, nastaví se seed podle času. Tzn. že když v jedné sekundě vytvoříme více instancí třídy `Random`, budou mít všechny tyto instance stejný seed. V praxi to znamená, že bychom se měli vyhnout vytváření instancí třídy `Random` například uvnitř cyklů, jinak nám hrozí situace, že námi vygenerovaná čísla budou všechny shodná.

Nyní máme vytvořený náš generátor a můžeme začít generovat čísla. Třída `Random` nám k tomu poskytuje dvě hlavní metody:

Hlavní metody

```
Random r = new Random();

int i = r.Next();

double d = r.NextDouble();
```

Metoda Next()

Nyní začneme metodou `Next()`. Tu použijeme, pokud budeme chtít vygenerovat náhodné celé číslo (`int`). Když metodu napíšeme bez parametrů, tak integer, který dostaneme, bude náhodné číslo větší nebo rovno nule. Tato metoda má ale více možností, jak ji použít. Pokud do závorky u metody `Next()` dáme jeden parametr typu `int`, nastavíme tím vrchní hodnotu (maximum), ke které se generátor může přiblížit. Když přidáme ještě jeden další parametr, dostaneme tím možnost nastavit celý rozsah. Tzn. minimum i maximum:

Náhodné číslo

```
Random r = new Random();

// Hodnoty pro "ohraničení" náhodného čísla
int min = 5;
int max = 25;

// Číslo větší nebo rovno nule [dostaneme číslo v rozsahu 0 - ...]
r.Next();

// Číslo větší nebo rovno nule a menší než parametr max [dostaneme číslo v rozsahu 0 - 24]
r.Next(max);

// Číslo větší nebo rovno parametru min a menší než parametr max [dostaneme číslo v rozsahu 5
- 24]
r.Next(min, max);
```

Minimum je inkluzivní (hodnota spadá do generované sekvence) zatímco maximum není.

Metoda NextDouble()

Tato metoda narozdíl od ostatních dvou nevyžaduje žádný parametr, neboť má přesně daný rozsah. Pomocí ní můžeme získat čísla datového typu `double`, která jsou v intervalu od 0 do 1:

Desetinné náhodné číslo

```
Random r = new Random();
```

```
// Vypíše náhodný double
Console.WriteLine(r.NextDouble());
```

Realizace náhodného čísla bez opakování

Když generujeme náhodná čísla, našemu generátoru nevadí udělat klidně více stejných čísel za sebou. Někdy se to náhodou stane. Ale v nějakých případech chceme takovému chování zabránit. Například ve sportce, když se losují čísla, nikdy nemůže dostat více stejných čísel za sebou. Teď si ukážeme, jak takového výsledku docílit. Budeme si muset vytvořit list, do kterého postupně budeme přidávat hodnoty, které jsme už použili. Pomocí metody `Contains()` pak budeme kontrolovat, zda se číslo, které jsme vygenerovali, neshoduje s nějakým v našem listu. Proces budeme opakovat, dokud nenajdeme číslo, které jsme ještě nepoužili:

Náhodné číslo bez opakování

```
// List hodnot, které jsme již použili
List<int> pouziteHodnoty = new List<int>();

// Pokusíme se vypsát 100 čísel, které se neopakují
for (int i = 0; i < 100; i++)
{
    // Hodnota, se kterou právě pracujem
    int generovanaHodnota;

    do
    {
        // Necháme vytvořit náhodné číslo
        generovanaHodnota = r.Next();

        } while (pouziteHodnoty.Contains(generovanaHodnota)); // Pokud ho náš List již obsahuje,
zkusíme vytvořit další

        // Až se nám podaří najít hodnotu, kterou jsme ještě nepoužili, tak ji přidáme do listu
        pouziteHodnoty.Add(generovanaHodnota);

        // Vypíšeme unikátní číslo
        Console.WriteLine(generovanaHodnota);
    }
}
```

Pokud bychom chtěli náhodná čísla bez opakování v nějakém intervalu (např. od 0 do 10), museli bychom si pohlídat i situaci, kdy nám žádná náhodná čísla už nezbyvají. Pokud by tato situace nastala, mohli bychom například vyhodit náš vlastní `Exception`.

Shrnutí

V této kapitole jsme se naučili pracovat s třídou `Random` a ukázali jsme si, jak vyřešit časté problémy při jejím používání.

Revision #3

Created 2025-05-21 10:21:44 UTC by Magdalena Dobešová

Updated 2025-05-21 10:27:52 UTC by Magdalena Dobešová