**ÚVOD DO PROGRAMOVANIA**

Program je sled príkazov. Každý program sa dá rozdeliť na menšie elementárne úlohy, ktoré sú riešiteľné príkazmi konkrétneho jazyka. Príkazy vykonáva procesor, údaje/dáta sú uložené v operačnej pamäti RAM.

Procesor teda vykonáva jednoduché príkazy (inštrukcie – RISC/CISC). Ich zápis sa nazýva strojový kód a je pre ľudí nečitateľný. Čiastočne čitateľnejšia forma je zápis kódu v jazyku assembler, každý procesor má však inú syntax tohto jazyka.

Čitateľnosť zlepšil príchod vyšších programovacích jazykov (napr. C). Tie sú písané príkazmi ako napr. while (pokiaľ), break (preruš) a pod. Takýto jazyk patrí do skupiny tzv. štruktúrovaných. Pre ešte ľahšie/názornejšie programovanie vznikli objektovo orientované jazyky (napr. C++/C#/Java), keďže všetko vôkol nás sa dá vnímať ako objekty s vlastnosťami a činnosťami, ktoré sa s nimi dajú robiť.

Kód programu písaný v prog. jazyku sa volá zdrojový kód. Tento kód spracuje program zvaný kompilátor, ktorý ho pretransformuje do strojového kódu konkrétnej platformy. Zdrojový kód môže byť vstupom aj pre interpréter, čo je program, ktorý priamo spúšťa program v sebe. Niečo ako emulátor. Ponúka multiplatformovosť, no zaťažuje HW oproti natívnemu programu.

Vývojové prostredie je program, ktorý slúži na pohodlné písanie kódu. Okrem editora môže obsahovať kompilátor, simulátor, opravovač/dopisovač kódu, modul na komunikáciu a pod.

Programy vytvárame v jazyku C#, ako konzolové aplikácie .NET Framework vo Visual Studio IDE. Konverziu a spustenie robíme cez klávesovú skratku CTRL+F5 (pozdrží zatvorenie okna).

**JAZYK C# - PREMENNÁ**

Základným prvkom každého programu je premenná, keďže program pracuje s dátami. Programátor nevie, aké dáta budú, preto ich univerzálne v kóde reprezentuje premennou (odkaz na časť pamäte, kde sa dáta nachádzajú).

Typy premenných:

*int cislo = 1; // cele cislo*

*float desatinne = 3.14f; // zakladne desatinne cislo*

*double dvojnasobne = 3.14; // presnejsie desatinne cislo*

*char znak = ‘1‘; // znak sa pise do apostrofu*



*string text = “1“; // pouzivaju sa uvodzovky*

*bool bolean = true; // resp. false*

*var univerzalnyTyp; // typ sa prisposoby podla toho, aka hodnota sa don PRVA zapise*

-deklarácia / inicializácia / definícia

-lokálne / globálne

-existencia v rámci blokov a prekrytie

-pravidlá pre tvorbu názvov (znaky / čísla / ťaví zápis / prvé písmeno )

-modifikátory (unsigned, signed, short, long, const)

Príklad:

static void Main(string[] args)

{

int c1; // vytvorenie premennej pre cele cislo

int c2 = 0; // vytvorenie a naplnenie nulou

Console.WriteLine("Zadaj číslo 1:"); // vypis textu

c1 = int.Parse( Console.ReadLine() ); // nacitanie a konvertovanie



Console.WriteLine("Zadaj číslo 2:"); /\* koment od-do \*/

string cislo2 = Console.ReadLine(); // nacitanie textu

c2 = Convert.ToInt32(cislo2); // konverzia stringu na integer

Console.WriteLine("Súčet: " + (c1 + c2) ); // automaticka konv. i->s

Console.WriteLine("Rozdiel: " + (c1 - c2).ToString() ); // cez metodu

Console.WriteLine("Podiel: " + Convert.ToString(c1 / c2) ); // cez inu

int sucin = c1 \* c2; // zapis vypoctu do pomocnej premennej

Console.WriteLine("Súčet: " + sucin ); // vypis pomocnej premennej

}

*Úloha: Vytvorte program, ktorému zadáte rok narodenia a aktuálny rok a on vypíše, koľko máte rokov.*

**JAZYK C# - OPERÁTORY**

Operátory slúžia na prácu s premennými (aritmetické, binárne / logické, relačné):

|  |  |  |
| --- | --- | --- |
| **SYNTAX** | **VÝZNAM** | **POUŽITIE** |
| + | aritmetický súčet (sčítanie) | 5 + 2 (7) |
| - | aritmetický rozdiel (odčítanie) | 5 – 2 (3) |
| \* | aritmetický súčin (násobenie) | 5 \* 2 (10) |
| / | aritmetický podiel (delenie) | 5 / 2 (2) |
| % | zvyšok po celočíselnom delení (modulo) | 5 % 2 (1) |
| ++ | inkremenácia (zvýšenie o 1) | 5++ (6) |
| -- | dekrementácia (zníženie o 1) | 5-- (4) |
| = | priradenie | a = 5 |



|  |  |  |
| --- | --- | --- |
| **SYNTAX** | **VÝZNAM** | **POUŽITIE** |
| & | bitový súčin (AND) | 1001 & 1010 (1000) |
| | | bitový súčet (OR) | 1001 | 1010 (1011) |
| ^ | bitové vylúčenie (XOR) | 1001 ^ 1010 (0011) |
| ~ | bitová negácia (komplement) | ~1001 (0110) |
| << | bitový posun vľavo (shift vľavo) | 01 << 2 (0100) |
| >> | bitový posun vpravo (shift vpravo) | 0110 >> 2 (01) |
| && | logický súčin (AND) | POD1 && POD2 |
| || | logický súčet (OR) | POD1 || POD2 |
| ! | logická negácia (NOT) | !true (false) |
| ? | podmienkový operátor | POD ? PR1 : PR2 |

|  |  |  |
| --- | --- | --- |
| **SYNTAX** | **VÝZNAM** | **POUŽITIE** |
| == | test rovnosti | A == B |
| != | test nerovnosti | A != B |
| < | test menšieho | A < B |
| <= | test menšieho alebo rovného | A <= B |
| > | test väčšieho | A > B |
| >= | test väčšieho alebo rovného | A >= B |

Pre generovanie náhodných čísel treba vytvoriť objekt, z ktorého čísla budeme generovať. Metóda na generovanie sa volá Next() a má tri druhy preťaženia.

Random rrr = new Random(); // vytvorenie objektu pre generovanie

int nahoda = rrr.Next(1, 7); // vygenerovanie cisla od 1 do 6

*Úloha: Vytvorte program, ktorý vypíše 3 mená a k ním náhodne tri čísla známok (1-5)*

**JAZYK C# - VETVENIE**

Vetvenie sa používa vtedy, pokiaľ chceme, aby sa určitá časť kódu vykonala za určitých podmienok a za iných podmienok sa vykonala nevykonalo nič alebo niečo iné.

if() – jednoduché vetvenie, pokiaľ platí podmienka, kód sa vykoná, inak sa nevykoná

if() – kód v bloku za if sa vykoná pokiaľ platí podmienka, inak sa vykoná kód v bloku za else  
else

if() – postupne sa testujú podmienky, pokiaľ niektorá platí, vykoná sa daný blok, zvyšok sa preskočí  
else if()  
else if()  
else

switch() – vetvenie, ktoré testuje rovnosť dvoch hodnôt. Používa sa pri výberoch, stavoch a pod.

Príkazy if a if-else if-else if-else nie sú samostatné príkazy, ide o modifikáciu vetvenia if()-else. Platí totiž, že časť else sa nemusí písať.

Pri vetvení aj cykloch platí, že pokiaľ je v bloku jeden príkaz, zátvorky sa môžu a nemusia písať.

Príklad:

int cislo1; // premenné pre dve čísla a text

int cislo2;

string operacia;

Console.WriteLine("Zadaj číslo 1: "); // výpis na obrazovku

cislo1 = int.Parse( Console.ReadLine() ); // načítanie INT

Console.WriteLine("Zadaj číslo 2: ");

cislo2 = Convert.ToInt32( Console.ReadLine() );

Console.WriteLine("Zadaj operáciu (+-\*/): ");

operacia = Console.ReadLine(); // načíta sa STRING a NEkonvertuje sa

if(operacia == "+") // ak sa zadal znak +, tak ...

Console.WriteLine("Súčet bol: " + (cislo1+cislo2).ToString() );

else if (operacia == "-") // ak sa zadal znak -, tak ...

Console.WriteLine("Rozdiel bol: " + (cislo1 - cislo2).ToString());

else if (operacia == "\*")

Console.WriteLine("Súčin bol: " + (cislo1 \* cislo2).ToString());

else if (operacia == "/")

Console.WriteLine("Podiel bol: " + (cislo1 / (float)cislo2 ).ToString());

else

Console.WriteLine("ERROR. Neznáma operácia!");

Príklad:

Random r = new Random(); // vytvorenie objektu pre náhodné generovanie

int znamka = r.Next(1, 6); // vygenerovanie čísla medzi 1 a 5

string slovne; // pomocná premenná pre text

switch(znamka)

{

case 1: // ak sa znamka == 1 ,tak...

slovne = "výborný"; // zapíš text do premennej

break; // ukonči vetvenie switch

case 2: // ak sa znamka == 2 ,tak...

slovne = "chválitebný";

break;

case 3: // ak sa znamka == 3 ,tak...

slovne = "dobrý";

break;

case 4:

slovne = "dostatočný";

break;

case 5:

slovne = "nedostatočný";

break;

default: // ak sa znamka nerovnala 1,2,3,4 alebo 5 ,tak...

slovne = "ERROR";

break;

}

Console.WriteLine("Monika bola hodnotená: " + slovne); // výpis