

Vježba 1

Programiranje 1 (550) - ak. god. 2024./2025.

Uvod

VB.NET programi se sastoje od jednog ili više *modula*, koji služe za logičko grupiranje dijelova programa.

Moduli se sastoje od procedura, funkcija, klasa, struktura, nabiranja, itd. Svi navedeni elementi programa će biti obrađeni tijekom kolegija.

Posebnu važnost ima procedura `Main`, jer ona predstavlja polaznu točku izvršavanja programa. U slučaju da u programu imamo više modula, svaki od njih može imati proceduru imena `Main`, ali samo jedna od njih može biti polazišna točka programa.

Varijable i tipovi podataka

Varijable koristimo za pohranu podataka u memoriju računala. Varijabla upućuje na memorijsko mjesto gdje će biti pohranjena vrijednost varijable. Različiti tipovi varijable rezerviraju različite količine mjesta u memoriji kako bismo mogli pohraniti različite vrste podataka.

Većina osnovnih tipova podataka u VB.NET je predstavljena strukturama ili klasama što nam, osim samog pohranjivanja u memoriji, omogućuje još neke korisne operacije nad podacima pohranjenim u varijabli.

Sintaksa za deklariranje varijabli je:

VB

```
1 Dim <identifikator> As <tip>
```

Primjeri:

VB

```
1 Dim broj As Integer
2 Dim slovo As Char
3 Dim recenica As String
4 Dim a, b As Float 'dviije varijable istog tipa u istoj liniji
```

Osnovni tipovi podataka koji se koriste u VB.NET-u su dani u tablici. Za brojčane tipove podataka, minimalne i maksimalne vrijednosti su određene veličinom brojčanog zapisa.

VB.NET tip	CTS tip	Veličina	Opis
Boolean	Boolean	4 bajta	true ili false
Byte	Byte	1 bajt	Pozitivna cjelobrojna vrijednost
Short	Int16	2 bajta	Cjelobrojna vrijednost s predznakom
Integer	Int32	4 bajta	Cjelobrojna vrijednost s predznakom
Long	Int64	8 bajta	Cjelobrojna vrijednost s predznakom
Single	Single	4 bajta	Vrijednost s pomičnim zarezom
Double	Double	8 bajta	Vrijednost s pomičnim zarezom
Decimal	Decimal	16 bajta	
Date	DateTime	8 bajta	Datum i vrijeme
Char	Char	2 bajta	Jedan UNICODE znak
String	String	Promjenjivo	Niz znakova

Pridruživanje vrijednosti varijablama se obavlja korištenjem operatora pridruživanja: `=`, `+=`, `*=`, itd. Primjeri:

VB

```
1 broj = 123
2 broj += 25 'Uzima trenutnu vrijednost varijable broj i uvećava je za 25
3 broj *= 2 'Uzima trenutnu vrijednost varijable broj i množi je s 2
```

Pridruživanje vrijednosti nekoj varijabli je moguće u istoj liniji gdje se varijabla deklarira, npr.

VB

```
1 Dim broj As Integer = 37
```

Implicitno deklariranje varijabli

Varijable je u VB.NET programskom jeziku moguće definirati i implicitno, što nam omogućuje da pridružujemo vrijednost i varijablama koje nisu prethodno deklarirane. Ova funkcionalnost se može uključiti navođenjem naredbe `VB Option Explicit Off` na vrhu datoteke s izvornim kodom programa. Sve implicitno deklarirane varijable su tipa `VB Object`, pa se time smanjuje memorijska efikasnost. Stoga će se na ovim vježbama uvijek koristiti eksplicitno deklariranje varijabli, što je automatski uključeno i nije potrebno pisati nikakve dodatne naredbe da bi se koristio ovaj način deklariranja varijabli.

Konzolne aplikacije

Konzolne aplikacije su aplikacije bez grafičkog sučelja, gdje se sva interakcija odvija u tzv. komandnoj liniji (ili naredbenom retku). Ti programi putem komandne linije ispisuju rezultate izvršavanja programa, kao što i uzimaju ulaze koje upisuje korisnik.

Za interakciju s konzolnom aplikacijom koristi se klasa `VB Console`, koja opisuje standardni ulazno-izlazni tok podataka. Standardni ulaz je povezan s tipkovnicom, pa tako postoji mogućnost da se ono što korisnik unese putem tipkovnice na jednostavan način „pročita” i prebaci u program. Standardni izlaz je usmjeren na zaslon računala.

Klasa `VB Console` sadrži postupke za interakciju s komandnom linijom. Najčešće ćemo iz nje koristiti postupke `VB Write()` i `VB WriteLine()` za ispis u komandnu liniju, te `VB Read()` i `VB ReadLine()` za čitanje što je korisnik upisao.

Ispis teksta u konzoli

Za ispis teksta u konzolu koristimo postupke `VB Write()` i `VB WriteLine()`. Ova dva postupka su vrlo slična, način njihova korištenja je identičan, a jedina razlika je što ova druga pozicionira kursor u novi red, dok ga prva ostavlja tamo gdje je završen ispit teksta koji je ona ispisivala.

Ovi postupci se pozivaju tako da se unutar zagrada stavi tekst koji želimo da se ispiše u konzoli. Taj tekst može biti „konstantan”, pa ćemo staviti u navodnike, ili može biti pohranjen u jednoj ili više varijabli, koje možemo koristiti samostalno, ili u kombinaciji s konstantnim tekstom navedenim u navodnicima.

Tekst koji će se ispisati u konzoli je moguće formatirati i pomoću predloška u kojemu se rezerviraju mjesta koja se pri izvođenju programa popune nekim vrijednostima koje se nalaze u varijablama, rezultatima izvršavanja funkcija i sl. Npr:

VB

```
1 Console.WriteLine("Suma brojeva {0} i {1} je {2}", 2, 3, 2+3)
```

U ovom primjeru će se za vrijeme izvršavanja programa u ovaj tekst koji je napisan `{0}` zamijeniti s vrijednošću prvog parametra koji je naveden iza teksta (2), `{1}` će se zamijeniti s drugonavedenim (3), a `{2}` s trećim (tj. s rezultatom izraza 3+2), itd. Brojanje počinje od 0, a unutar teksta ih nije potrebno navoditi po redu pojavljivanja, npr.

VB

```
1 Console.WriteLine("{1} {0} {2}", 10, 20, 30)
```

će dati ispit 20 10 30.

Drugi, po nekima i lakši način za formatiranje ispisa je direktno umetanje vrijednosti određenih varijabli ili izraza u tekst u vitičastim zagradama, a da bi takav tekst bio ispravno *interpoliran*, potrebno ga je označiti znakom `$` **ispred** navodnika, npr.

VB

```
1 Console.WriteLine($"Suma brojeva {2} i {3} je {2+3}")
```

Ovaj vrsta ispisa se zove *interpolirani string* (string je niz znakova). Svaka varijabla, konstanta ili izraz unutar vitičastih zagrada se izračuna i umetne na to mjesto unutar niza znakova, a zatim se niz znakova ispisuje.

Unos podataka iz konzole

Za učitavanje podataka koje unosi korisnik iz konzole koriste se postupci `VB Read()` i `VB ReadLine()`. Prva naredba čita točno jedan (sljedeći) znak s konzole koji unosi korisnik, dok druga čita sve znakove s konzole, sve dok korisnik ne pritisne enter.

Jednostavna konzolna aplikacija

Navedeno ćemo ilustrirati jednostavnom konzolnom aplikacijom u kojoj će aplikacija tražiti od korisnika da unese dva cijela broja, te će ispisati njihovu sumu i produkt.

VB

```
1 Dim a As Integer
2 Dim b As Integer
3 Dim suma As Integer
4 Dim produkt As Integer
5
6 ' Unos prvog broja
7 Console.Write("Unesi prvi broj: ")
8 a = Console.ReadLine()
9
10 ' Unos drugog broja
11 Console.Write("Unesi drugi broj: ")
12 b = Console.ReadLine()
13
14 ' Radimo izračun
15 suma = a + b
16 produkt = a * b
17
18 ' Ispis rezultata
19 Console.WriteLine($"Suma brojeva {a} i {b} je {suma}")
20 Console.WriteLine("Produkt brojeva {0} i {1} je {2}", a, b, produkt)
```

Konstante

Kao konstante pohranjujemo vrijednosti koje se ne mijenjaju za vrijeme izvršavanja programa. Uobičajeno je definirati konstante za neke vrijednosti koje se često koriste u programu, kako bismo povećali čitljivost koda i olakšali buduće održavanje programa. VB.NET poznaje dva tipa konstanti:

- konstantne vrijednosti
- simboličke konstante

Konstante vrijednosti su nepromjenjive vrijednosti koje se koriste u programu, te imaju svoj tip podataka. Npr.

VB

```
1 Dim a as Integer = 23
2 Dim b as Integer
3 b = a + 5
```

U navedenom primjeru, 5 je konstantna vrijednost. Ona će biti predstavljena tipom koji se određuje iz konteksta u kojem se ta konstanta koristi. U navedenom primjeru, konstanta 5 će biti tipa `Integer` jer su `a` i `b` tipa `Integer`, iako se mogla predstaviti i nekim manjim tipom, primjerice `Byte`.

Simboličke konstante su slične varijablama te također sadrže tip. Deklariraju se na isti način osim što se umjesto ključne riječi `Dim` koristi ključna riječ `Const`. Npr.

VB

```
1 Const PI as Double = 3.14159
```

Simboličke konstante

Simboličke konstante ne mogu mijenjati vrijednost. Ako trebate nešto što mijenja vrijednost, deklarirajte varijablu umjesto konstante.

Pretvorba tipova podataka

Podatke je moguće u VB.NET-u pretvoriti u druge tipove, a proces se naziva pretvorba tipova (engl. *type casting*).

Kada neki tip podataka pretvaramo u neki drugi tip koji je veće memorijske veličine, ta operacija nikada ne rezultira gubitkom podataka. Primjer je pretvorba iz tipa `Integer` u tip `Long`. Oba tipa su cjelobrojni, a kako je raspon tipa `Integer` manji od raspona tipa `Long`, do gubitka ne dolazi. Ovakva pretvorba se naziva **proširena pretvorba**.

Do gubitka podataka može doći kod pretvaranja nekog tipa podataka u drugi tip koji je manje memorijske veličine ili ako se radi o varijabli različitih vrsta. Primjerice, ako varijablu tipa s pomičnim zarezom želimo pretvoriti u cjelobrojnu, izgubit ćemo decimalni dio.

Drugi tip operacije kod koje može doći do gubitka podataka je tzv. **sužena pretvorba**, primjerice iz `Integer` u `Short`, jer je tip `Short` manji od tipa `integer`. Ipak, do gubitka neće doći uvijek, već samo ako je vrijednost originalne varijable izvan raspona koji se može pohraniti u novi tip varijable. S obzirom da je sužena pretvorba relativno „opasna” operacija koja ponekad rezultira gubitkom podataka, u VB.NET-u postoji opcija koja takve operacije onemogućuje. Ako želimo uključiti ovu opciju, navodimo `Option Strict On` na vrhu programa. Ako se koristi ova opcija, za operacije sužene pretvorbe se koriste postupci i funkcije iz klase `Convert`, koje vode računa dolazi li do gubitka podataka, te obavljaju konverziju ako je konverzija uredna (nema gubitka), a inače podižu iznimke. Korištenje ovih postupaka je poželjno i kada nismo uključili opciju striktno pretvorbe, tj. kada je `Option Strict Off` (što je defaultni slučaj).

Zadaci

1. Napišite program u kojem korisnik putem konzole unosi vrijednost u milimetrima, te se ta vrijednost ispisuje u centimetrima i metrima.
2. Napišite aplikaciju u kojoj korisnik unosi bazu i eksponent putem konzole. Ispišite vrijednost potencije. Za potenciranje koristite matematičku funkciju `Math.Pow()`.
3. Korisnik unosi dva broja putem konzole. Pohranite te vrijednosti u varijable, te im zamijenite vrijednosti, bez korištenja treće varijable.
4. Napišite aplikaciju koja računa drugi korijen broja koji je korisnik unio putem konzole. Zadatak riješite bez eksplicitnog deklariranja varijabli!
5. Napišite program u kojem korisnik unosi radijus, a program ispisuje oplošje i volumen kugle tog radijusa.
6. Napišite program koji će u konzoli ispisati koliko byteova ima jedan kibi, Mebi i Gibi.