



Upute za izradu vježbi/zadataka

Vježbe je potrebno izraditi pomoću razvojne okoline MS Visual Studio. Prije dolaska na vježbu potrebno je izraditi pripremu. Bez napravljene pripreme nije moguće pristupiti izradi vježbi.

Prilikom završetka izrade vježbe, nakon što je dežurni asistent sve pregledao:

1. Potrebno pokrenuti i osigurati da se izvodi bez pogrešaka, izrađene vježbe koje u sebi sadrže programske pogreške NEĆE biti priznate kao odrađene!
2. Sve napravljene projekte u razvojnoj okolini MS Visual Studio potrebno je arhivirati s nazivom u slijedećem obliku – *JMBAG_GRPUGGGG-MM-DD_VJEŽBE#_ZADATAK#.zip* te unijeti u sustav Merlin.

Primjer ispravnog naziva ZIP datoteke: 0135123456_A_2011-10-25_VJEZBA01.zip

Vrijeme za izradu navedenih zadataka iznosi 90 minuta.

Cilj vježbe: Upoznavanje s naredbama za unos i ispis različitih tipova podataka. Prilagođavanje formata ispisa unutar tekstualnog sučelja.

Opis vježbe

Aplikacije u računalima obrađuju podatke. Svi podaci u računalima su prikazani u binarnom brojevnom sustavu pri čemu se značenje određene kombinacije znamenki „0“ i „1“ definira kôdom za taj pripadni tip podatka. Prilikom unosa podataka u računalo sve se prvo prikazuje podatkom tipa „string“ odnosno nizom znakova ili podatkom tipa „int“ odnosno cijelim brojem ako se radi o učitavanju samo jednog znaka. Za učitavanje podataka je tako moguće koristiti sljedeće metode (sastavni dijelovi sistemske metode „Console“ namijenjene za rad s tekstualnim sučeljem odnosno konzolom):

- „Read“ -> čita znak po znak i vraća podatak tipa „int“;
- „ReadLine“ -> čita niz znakova i vraća podatak tipa „string“.

Prilikom čitanja znak po znak potrebno je obratiti pažnju da osim znakova koje je unio korisnik postoje i sistemski znakovi za kraj unosa te novi red. Navedene znakove je također potrebno učitati. Za pretvorbu učitano podatka u željeni format podatka potrebno je koristiti metodu „Convert“ za pretvorbu podataka. Radi se o sistemske metodi iz imeničkog prostora „System“. Metoda „Convert“ podržava sljedeće bitne pretvorbe formata podataka (početni tip podatka je „string“):

- „ToBoolean“ -> u logičku varijablu;
- „ToInt16“ -> u cijeli broj duljine 2 byte;
- „ToInt32“ -> u cijeli broj duljine 4 byte;
- „ToInt64“ -> u cijeli broj duljine 8 byte;

- „ToString“ -> u niz znakova.
- „ToChar“ -> u znak;
- „ToDecimal“ -> u broj s plivajućim zarezom većeg opsega;
- „ToDouble“ -> u broj s plivajućim zarezom;

Kao ulaznu varijablu za metodu „Convert“ moguće je iskoristiti rezultat koji vraćaju metode za čitanje s konzole. Situacija je vidljiva u slučaju programskog kôda za čitanje podatka tipa „int“:

```
// korištenje metode ReadLine
int cijeliBroj;
cijeliBroj = Convert.ToInt32(Console.ReadLine());
```

Metoda „Console.ReadLine“ dohvaća niz znakova s konzole i prosljeđuje ga dok metoda „Convert.ToInt32“ prihvaćeni učitani niz znakova pretvori u cijeli broj. Pri tome je bitno poštivati pravila za unos cijelih brojeva da ne dođe do pogreške odnosno generiranje iznimke (engl. „exception“) te mogućeg „plavog ekrana smrti“. To znači da se prilikom unosa cijelog broja smiju koristiti samo znakovi za prikaz znamenki od 0 do 9.

Za prikaz rezultata izvršavanja aplikacije na zaslonu monitora potrebno je napraviti ispis podataka. U tu svrhu je moguće koristiti metode (sastavni dijelovi systemske metode „Console“ namijenjene za rad s tekstualnim sučeljem odnosno konzolom):

- „Write“ -> ispisuje prosljeđeni niz znakova;
- „WriteLine“ -> ispisuje prosljeđeni niz znakova uz automatski ispis znaka za novi redak.

Moguće je napraviti neformatirani i formatirani ispis. Neformatirani ispis izravno konzoli prosljeđuje dani niz znakova bez zadavanja opcija ispisa dok je u slučaju formatiranog ispisa moguće zadati opcije izgleda ispisa. Primjer neformatiranog ispisa učitanih cijelih brojeva dan je sljedećim kôdom:

```
// ispis podataka
Console.WriteLine("Učitani cijeli broj je " + cijeliBroj);
```

Potrebno je primijetiti da je cijeli broj potrebno pretvoriti u niz znakova pri čemu današnji prevodioci to rade automatski tako da u ovom slučaju nije potrebno raditi ranije spomenuti pretvorbu. Ukoliko je potrebno napraviti pretvorbu, u tu je svrhu moguće iskoristiti pomoćnu metodu „ToString“ koja je sastavni dio svake varijable. Za dohvatanje navedene pomoćne metode se koristi operator točka nakon imena varijable. Uz korištenje te pomoćne metode gornji programski kôd se može napisati kao:

```
// ispis podataka
Console.WriteLine("Učitani cijeli broj je " + cijeliBroj.ToString());
```

Kod formatiranog ispisa podataka moguće je dodavati opcije izgleda ispisa. Opcije uključuju broj decimala za ispis, poravnanja u tekstu, razmak prema prethodnom tekstu i sl. Primjer formatiranog ispisa broja PI na tri decimale dan je sljedećim kôdom:

```
// ispis podataka
Console.WriteLine("Broj PI: {0,0:0.###}", Math.PI);
```

Osim ručnog podešavanja formata ispisa moguće je koristiti već predefinirane formate za ispis postotka, ispis novčane jedinice, heksadecimalni format i sl. U tom slučaju potrebno je u definiciji formata ispisa iza dvotočke staviti oznaku željenog formata prema popisu danom u predavanju broj 11.

Priprema za vježbu

- Proučiti predavanja broj 10 i 11 vezane za prikaz podataka te unos i ispis podataka.
- Popuniti sljedeću tablicu i pokazati dežurnom asistentu na početku vježbe.
- Riješiti zadatak u nastavku (napisati programski kôd) i pokazati ga dežurnom asistentu na početku vježbe.

Napomena:

Pripremu za laboratorijsku vježbu potrebno je napisati vlastoručno na ovom papiru. Student koji prilikom ulaska u dvoranu nema napisanu pripremu nema pravo pristupa laboratoriju i smatra se da student nije odradio laboratorijsku vježbu.

Tablica za objašnjenje pojedinih pomoćnih metoda

Ime metode	Namjena metode	Primjer kôda
Console.Write		
Console.WriteLine		
Console.ReadLine		
Console.ReadKey		
Convert.ToInt32		
Convert.ToChar		
Convert.ToDouble		

Program za unos i ispis različitih numeričkih vrijednosti

Potrebno je napisati program u programskom jeziku C# koji će učitati po jednu vrijednost za svaki od sljedećih tipova podataka: short, int, long, char, string, float, double i decimal. Prilikom unosa je za svaki podatak potrebno ispisati pripadnu poruku operateru. Za izvođenje programa je potrebno pripremiti testne podatke za svaki tip. Za cijele brojeve je potrebno predložiti što veće vrijednosti koje pripadni tip može prikazati, za brojeve s plivajućim zarezom je potrebno predložiti što veće vrijednosti koje pripadni tip može prikazati uz najmanje 5 decimala, a za znak te niz znakova predložiti proizvoljne vrijednosti. Predložene vrijednosti potrebno je unijeti u tablicu dolje. Učitane podatke je potrebno ispisati u pripadnoj poruci u formatiranom obliku. Cijele brojeve je prilikom ispisa potrebno poravnati pri čemu svaku vrijednost ispisati u zasebnom retku. Brojeve s plivajućim zarezom je potrebno ispisati poravnate te sa samo dvije decimale. Znak i niz znakova ispisati po volji.

Programski kôd

Tip podatka	Predložena vrijednost	Tip podataka	Predložena vrijednost
short		float	
int		double	
long		decimal	
char		string	

Ime i prezime studenta:

Potpis studenta

Potpis dežurnog asistenta

Rad na vježbi

Rad na vježbi se sastoji od unosa programskog kôda danog dolje radi testiranja tijeka unosa podataka u računalo te prilagodbe formata ispisa. Nakon testiranja programskog kôda danog dolje potrebno je unijeti vlastiti programski kôd i testirati ga.

Unos različitih tipova podataka u računalo

Radi testiranja unosa podataka različitog formata u računalo potrebno je prvo kreirati novi projekt za izradu aplikacije s tekstualnim sučeljem imena „UnosIspisPodataka“. U predložak glavne metode potrebno je zatim unijeti sljedeći programski kôd:

```
// deklaracija varijabli
int unos, cijeliBroj;
char znak, znakKraj, znakNoviRedak;
double plivajuciZarez;
string ulazKonzola;

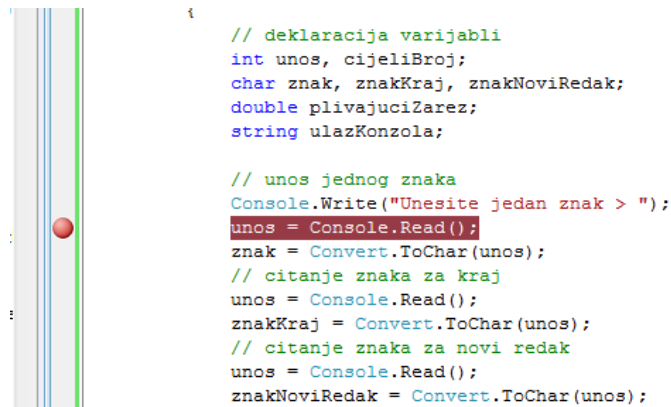
// unos jednog znaka
Console.Write("Unesite jedan znak > ");
unos = Console.Read();
znak = Convert.ToChar(unos);
// citanje znaka za kraj
unos = Console.Read();
znakKraj = Convert.ToChar(unos);
// citanje znaka za novi redak
unos = Console.Read();
znakNoviRedak = Convert.ToChar(unos);

// unos cijelog broja
Console.Write("Unesite cijeli broj > ");
ulazKonzola = Console.ReadLine();
cijeliBroj = Convert.ToInt32(ulazKonzola);


// unos broja sa plivajucim zarezom
Console.Write("Unesite broj s plivajucim zarezom > ");
ulazKonzola = Console.ReadLine();
plivajuciZarez = Convert.ToDouble(ulazKonzola);

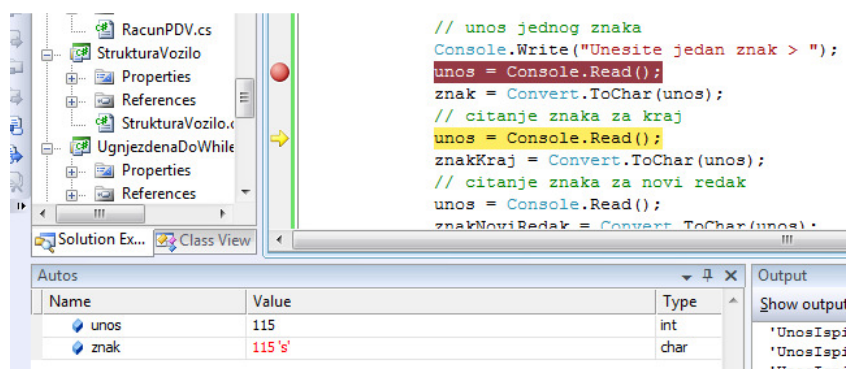
// unos niza znakova
Console.Write("Unesite niz znakova > ");
ulazKonzola = Console.ReadLine();
```

Uneseni programski kôd potrebno je provjeriti te generirati izvršnu datoteku. U slučaju postojanja pogrešaka potrebno je primijeniti postupak ispravljanja opisan u laboratorijskoj vježbi 10 koja se odnosi na osnove korištenja razvojne okoline MS Visual Studio. Nakon što su sve pogreške ispravljene potrebno je postaviti točku prekida u liniju kôda označenoj na slici 11.1.



Slika 11.1 Smještanje točke prekida

Pokretanjem programa pokreće se procedura ispravljanja i moguće je pratiti vrijednosti korištenih varijabli. Izvođenje programa staje kada dođe na red naredba u liniji kôda sa točkom prekida. U tom slučaju potrebno je dalje izvoditi program naredbu po naredbu korištenjem funkcije „Step Into“ na traci sa alatima (ikonica ). Pri tome je bitno uočiti da se ne izvodi linija kôda označena ispunjenim žutim pravokutnikom, već linija kôda prije označene linije. Slika 11.2 prikazuje takvu situaciju uz ispunjenu tablicu vrijednosti varijabli za primjer učitavanja slova „s“. Vidi se da pripadno slovo ima numeričku vrijednost 115 u UNICODE tablici i upravo je ta vrijednost učitana u računalo korištenjem metode „Read“.



Slika 11.2 Prikaz vrijednosti trenutno korištenih varijabli tijekom izvođenja programa

Izvršavanjem naredbe po naredbu aktivira se konzola i moguće je unijeti traženu vrijednost. Učitana vrijednost se prikazuje u tablici vrijednosti varijabli u razvojnoj okolini MS Visual Studio. Pripadne vrijednosti za odabrani znak potrebno je unijeti u tablicu 11.1 prema redosljedju izvođenju programa.

Ime varijable	Uneseni znak	Numerička vrijednost	UNICODE znak	Tip
unos				
znak				
unos				
znakKraj				
unos				
znakNoviRedak				

Tablica 11.1 Vrijednosti varijabli kod unosa jednog znaka

Nakon što dio programa, koji se odnosi na unos jednog znaka završi, počinje dio koji se odnosi na unos brojeva te niza znakova na kraju. Izvođenje programa potrebno je i dalje raditi naredbu po naredbu te za korištene ulazne podatke ispuniti tablicu 11.2.

Ime varijable	Uneseni znakovi	Numerička vrijednost	UNICODE znak	Tip
ulazKonzola				
cijeliBroj				
ulazKonzola				
plivajuciZarez				
ulazKonzola				

Tablica 11.2 Vrijednosti varijabli kod unosa numeričkih podataka i niza znakova

Ispis podataka

Za testiranje formata ispisa podataka potrebno je zatvoriti projekt korišten u prvom dijelu vježbe te kreirati nova imena „Formatiranispis“. U glavnu metodu potrebno je zatim unijeti programski kôd prikazan na slici 11.3.

```

// deklaracija varijabli
double brojVozilaJutro, brojVozilaPodne, brojVozilaVecer;
double razlikaPodne, razlikaVecer, razlikaPoslijepodne;

// inicijalizacija varijabli
brojVozilaJutro = 148.0;
brojVozilaPodne = 124.0;
brojVozilaVecer = 78.0;

// obrada podataka
razlikaPodne = (brojVozilaPodne - brojVozilaJutro) / brojVozilaJutro;
razlikaVecer = (brojVozilaVecer - brojVozilaJutro) / brojVozilaJutro;
razlikaPoslijepodne = (brojVozilaVecer - brojVozilaPodne) / brojVozilaPodne;

// ispis podataka
Console.WriteLine("Broj vozila se u podne prema jutru promijenio za " + razlikaPodne);
Console.WriteLine("Broj vozila se u vecer prema jutru promijenio za {0,0:P1}" , razlikaVecer);
Console.WriteLine("Broj vozila se u navecer prema podnevu promijenio za {0,0:P2}", razlikaPoslijepodne);

// ispis podataka
Console.WriteLine("Broj vozila ujutro je bio \t{0,5:0000}", brojVozilaJutro);
Console.WriteLine("Broj vozila u podne je bio \t{0,5:####}", brojVozilaPodne);
Console.WriteLine("Broj vozila navecer je bio \t" + brojVozilaVecer);

// da se vidi ispis konzole
Console.ReadLine();

```

Slika 11.3 Programski kôd za drugi dio vježbe

Nakon napravljene procedure provjere ispravnosti unesenog programskog kôda potrebno je pokrenuti program te ispuniti tablice 11.3 i 11.4. Tražene vrijednosti varijabli u programu mogu se izračunati ručno korištenjem zadanih konstanti odnosno postavljanjem prekidne točke te izvođenje programa naredbu po naredbu moguće je dohvatiti tražene vrijednosti tijekom izvođenja programa. Također navedite popis varijabli koje su bile poravnate prilikom ispisa. Prilikom unosa rezultata u tablicu pretpostavite da početak pojedinog retka tablice predstavlja početak retka za ispis unutar konzole.

Ime varijable	Vrijednost u programu	Ispisana vrijednost na konzoli	Format ispisa
razlikaPodne			
razlikaVecer			
razlikaPoslijepodne			

Tablica 11.3 Ispis prve skupine varijabli

Ime varijable	Vrijednost u programu	Ispisana vrijednost na konzoli	Format ispisa
brojVozilaJutro			
brojVozilaPodne			
brojVozilaVecer			

Tablica 11.4 Ispis druge skupine varijabli

Popis poravnatih varijabli:

Testiranje vlastitog programskog kôda

Program napisan u pripremi potrebno je iskoristiti za kreiranje novog projekta u programskom alatu MS Visual Studio pod imenom „UnoslspisPriprema“. Kao ulazne vrijednosti za testiranje programa iskoristiti vrijednosti predložene u pripremi. Rezultat izvođenja programa unijeti u tablicu dolje te pokazati dežurnom asistentu. Prilikom unosa rezultata u tablicu pretpostavite da početak pojedinog retka tablice predstavlja početak retka za ispis unutar konzole.

Tip podatka	Predložena vrijednost	Izgled ispisa
short		
int		
long		
char		
string		
float		
double		
decimal		

Napomena

Sve rezultate laboratorijske vježbe potrebno je na kraju laboratorijske vježbe pokazati dežurnom asistentu. U slučaju da student nije pokazao rezultate laboratorijske vježbe smatra se da istu nije odradio te ju je dužan nadoknaditi. Napravljene projekte u razvojnoj okolini MS Visual Studio te slike ekrana s rezultatima izvršavanja programa potrebno je spremi u obliku arhive (ZIP datoteke) te spremi u sustav Merlin.

Potpis studenta

Potpis dežurnog asistenta

Korisne napomene za rad na vježbi

- Vitičaste zagrade se dobivaju na HR tipkovnici kombinacijama tipki „Alt Gr“ + „b“ (lijeva vitičasta zagrada -> „{“) te „Alt Gr“ + „n“ (desna vitičasta zagrada -> „}“).
- Uglate zagrade se dobivaju na HR tipkovnici kombinacijama tipki „Alt Gr“ + „f“ (lijeva uglata zagrada -> „[“) te „Alt Gr“ + „g“ (desna uglata zagrada -> „]“).

- Tipkom „Delete“ ili „Del“ se briše znak ispred kursora.
- Tipkom „Backspace“ se briše znak iza kursora.
- Tipkom „Ins“ ili „Insert“ se aktivira opcija pisanja preko postojećeg teksta tako da novi znak prebriše postojeći znak. Aktivira se pritiskom na tipku „Ins“ ili „Insert“, a deaktivira ponovnim pritiskom.
- Potvrda unosa podatka u tekstualnom sučelje je pritisak na tipku „Enter“.
- Znak „|“ se dobiva na HR tipkovnici kombinacijom tipki „Alt Gr“ + „w“.
- Znakovi manje i veće se dobivaju na hrvatskoj tipkovnici kombinacijama tipki „Alt Gr“ + „,“ za znak „<“ te „Alt Gr“ + „.“ za znak „>“.