SVEUČILIŠTE U ZAGREBU
Fakultet prometnih znanosti
Zavod za inteligentne transportne sustave
Vukelićeva 4, Zagreb, HRVATSKA

Računalstvo

Operatori, pisanje izraza i osnove pseudokôda

Izv. prof. dr. sc. Edouard Ivanjko, dipl.ing.

Sadržaj

- Uvod
- Trigonometrijske funkcije
- Logaritamske funkcije
- Logički operatori
- Pisanje izraza
- Osnove pseudokôda

Uvod

- Za složenije proračune potrebno korištenje trigonometrijskih i logaritamskih funkcija te logičkih operatora
  - Složeniji programi uključuju zaključivanje i donošenje odluke
- U programskim jezicima postoje matematičke biblioteke koje uključuju složenije funkcije te konstante
  - Trigonometrijske i logaritamske funkcije, izračun najmanje/najveće vrijednosti između dva broja ili niza brojeva

Trigonometrijske funkcije

- Odnose se na trigonometrijske funkcije i njihov inverz
  - Sinus, cosinus, tanges i cotanges
  - Arkus sinus, arkus cosinus, arkus tanges i arkus cotanges
- Ulazna vrijednost je kut izražen u radijanima
  - Mi ljudi izražavamo kut u stupnjevima
  - Potrebna pretvorba iz stupnjeva u radijane
- Inverzna trigonometrijska funkcija daje rezultat u radijanima
  - Potrebna pretvorba iz radijana u stupnjeve

Trigonometrijske funkcije

- Naredbe za izračun trigonometrijskih funkcija

Funkcija od x	Pseudokôd	Raptor	C♯
sinus	sin(x)	sin(x)	Math.Sin(x)
kosinus	cos(x)	cos(x)	Math.Cos(x)
tangens	tan(x)	tan(x)	Math.Tan(x)
kotangens	cot(x)	cot(x)	1/Math.Tan(x)
sekans	1/cos(x)	1/cos(x)	1/Math.Cos(x)
kosekans	1/sin(x)	1/sin(x)	1/Math.Sin(x)

  - Programski jezik C# sadrži imenički prostor „Math” s pomoćnim metodama za matematičke funkcije
  - Neke matematičke funkcije se izračunavaju korištenjem definicije preko osnovnih funkcija

Trigonometrijske funkcije

- Naredbe za izračun inverza trigonometrijskih funkcija

Funkcija od x	Pseudokôd	Raptor	C♯
arkus sinus	arcsin(x)	arcsin(x)	Math.Asin(x)
arkus kosinus	arccos(x)	arccos(x)	Math.Acos(x)
arkus tangens	arctan(x)	arctan(x)	Math.Atan(x)
arkus kotangens	arccot(x)	arccot(x)	Math.Atan(1/x)

  - Funkcija arkus tanges je problematična jer je potrebno uzeti u obzir i kvadrant koordinatnog sustava
    - Neki programski jezici u tu svrhu imaju definiranu i dodatnu funkciju atan2
    - Programski jezik C# posjeduje pomoćnu metodu **Math.Atan2(y, x)**

Logaritamske funkcije

- Logaritamska funkcija je inverzija potenciranja

$$10^x = 100 \qquad x = \log_{10} 100$$

- Najčešće se koriste logaritmi prirodne baze (e = 2,718) i po bazi 10
  - Moguća promjena baze logaritma korištenjem ovih osnovnih logaritama

Logaritamska funkcija	Pseudokôd	Raptor	C♯
$log_{(10)}(a)$	Log(a)	Log(a)	Math.Log10(a)
$log_{(2)}(a)$	Log(a)/Log(2)	Log(a)/Log(2)	Math.Log(a)/Math.Log(2)
$ln(a)$	Ln(a)	Ln(a)	Math.Log(x)

Logički operatori

- Obrađuju logičke (engl. „Boolean”) varijable
  - Također je moguće koristiti logičke izraze te kombinacije logičkih i matematičkih izraza
- Logičke varijable mogu imati samo dvije vrijednosti
  - Logička „0” (neistina ili laž, engl. „false”)
  - Logička „1” (istina, engl. „true”)
- Numerička vrijednost se može pretvoriti u logičku vrijednost
  - Numerička vrijednost 0 predstavlja neistinu (laž)
  - Svaka druga numerička vrijednost predstavlja istinu

Logički operatori

- Za obradu logičkih varijabli se koriste logički operatori
- Svaki operator ima svoj simbol i tablicu istinitosti kao definiciju

Operator	Pseudokôd	Raptor	C#
Logičko NE	**-, ¯**	**NOT**	**!**
Logičko I	**I, •**	**AND**	**&** ili **&&**
Logičko uključivo ILI	**ILI, +**	**OR**	**|** ili **||**
Logičko isključivo ILI	**XOR**	**XOR**	**^**

Logički operatori

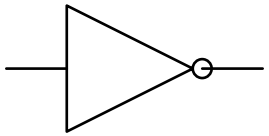
- Za izračun logičkih izraza s više varijabli potrebno u obzir uzeti prioritet operatora
- Redoslijed izračuna
  - Izraz u zagradi
  - Unarni operator logičko NE
  - Logičko I
  - Logičko ILI
- Ako postoji uzastopno više operatora istog prioriteta izračun se radi slijedno po dvije varijable u paru uz korištenje međurezultata prethodnog para varijabli

Logički operatori – Logičko NE

- Engleski naziv je NOT
- Djeluje samo na jednu logičku varijablu
  - Unarni operator
- Radi se o funkciji inverzije
  - Logička 0 postaje logička 1
  - Logička 1 postaje logička 0

$$Z = -A = \bar{A}$$

A	Z
LAŽ	ISTINA
ISTINA	LAŽ



NE-sklop

Logički operatori – Logičko NE

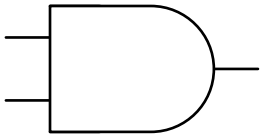
- Koristi se za prilagodbu vrijednosti logičke varijable kod kreiranja uvjeta odlučivanja
  - Npr. potrebno je reagirati kada neki uvjet nije ispunjen (logička nula), a koristimo logički operator koji reagira na vrijednost logičke jedinice
- Prometna analogija
  - Ako prijeđem cestu, neću ostati na istoj strani
    - Dolazim na drugu stranu ceste
  - Ako ne prijeđem cestu, neću doći na drugu stranu
    - Ostajem na istoj (početnoj) strani ceste

Logički operatori – Logičko I

- Naziva se još i konjukcija
- Engleski naziv je AND
- Služi kao funkcija za sigurnost
  - Izlaz će biti logička jedinica samo ako su svi ulazi jednaki logičkoj jedinici

A	B	Z
LAŽ	LAŽ	LAŽ
LAŽ	ISTINA	LAŽ
ISTINA	LAŽ	LAŽ
ISTINA	ISTINA	ISTINA

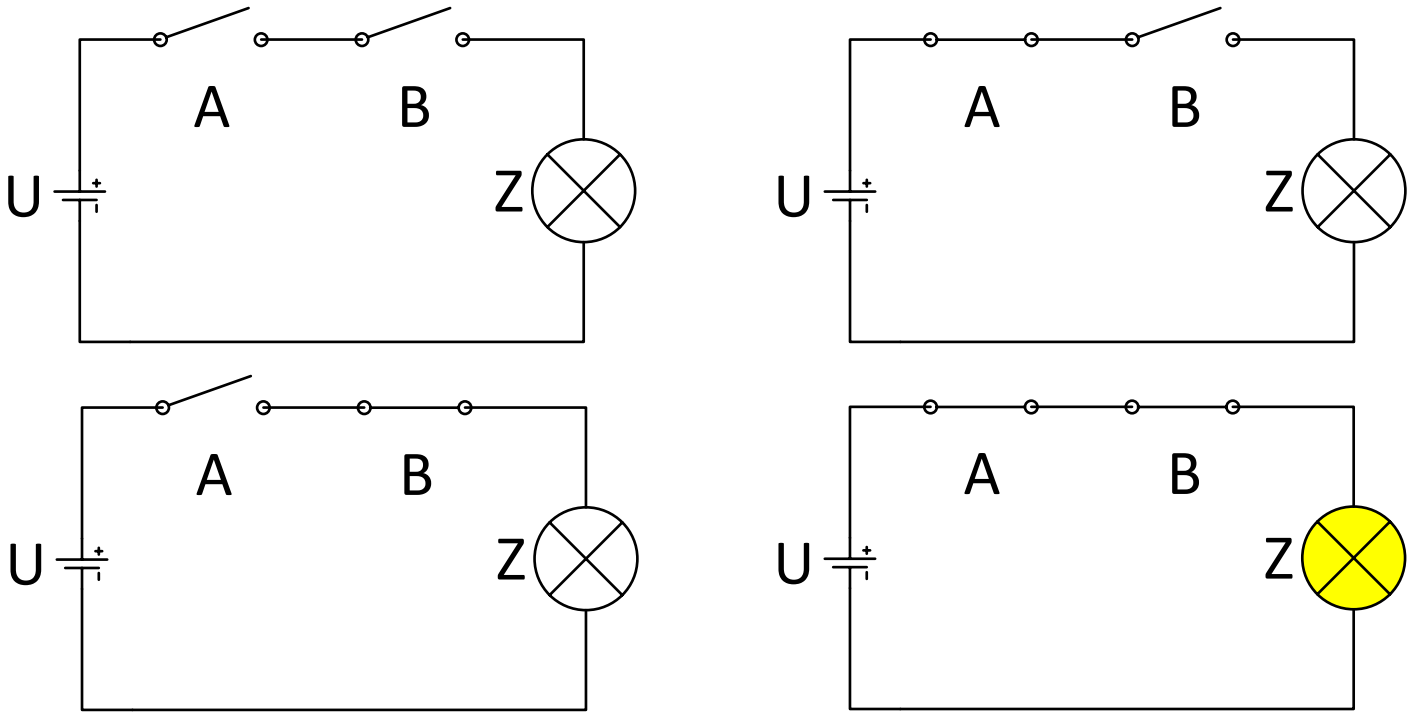
$$Z = A\ I\ B = A \bullet B = AB$$



I-sklop

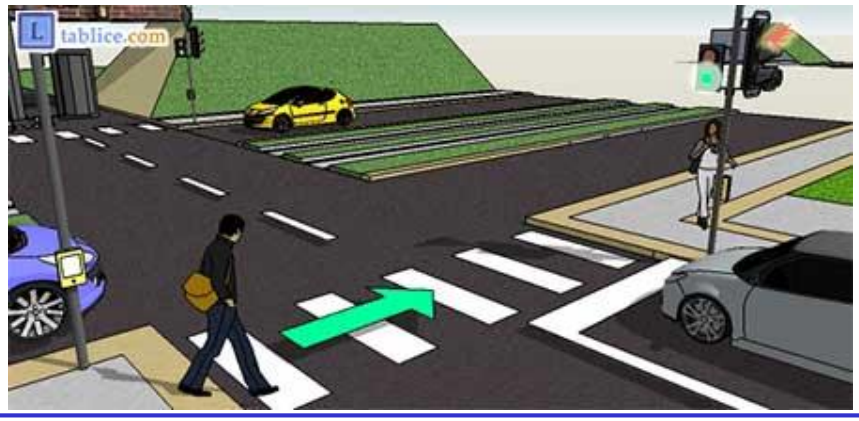
Logički operatori – Logičko I

- Primjer rada logičkog I korištenjem serijskog spoja sklopki
  - Sklopka otvorena -> logička 0
  - Sklopka zatvorena -> logička 1



Logički operatori – Logičko I

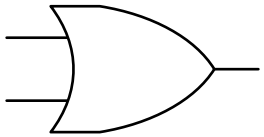
- Prometna analogija
  - Siguran prijelaz preko cestovne prometnice sa semaforom i pješačkim prijelazom
    - Dva uvjeta je potrebno ispuniti za siguran prijelaz preko pješačkog prijelaza
      - Zeleno svjetlo na semaforu za pješake
      - Svi automobili na prometnici su se zaustavili
    - Ako je na pješačkom semaforu zeleno i automobili su zaustavljeni, prijeći ću cestu



Logički operatori – Logičko ILI

- Naziva se još i uključivo ILI odnosno disjunkcija
- Engleski naziv je OR
- Predstavlja funkciju odabira
  - Izlaz je istinit ako je najmanje jedna ulazna vrijednost istinita

$$Z = A\ ILI\ B = A + B$$

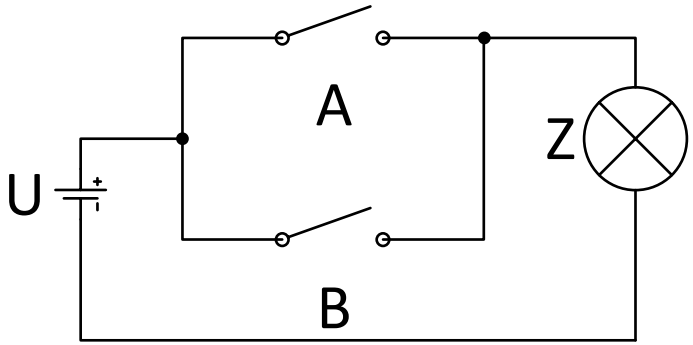


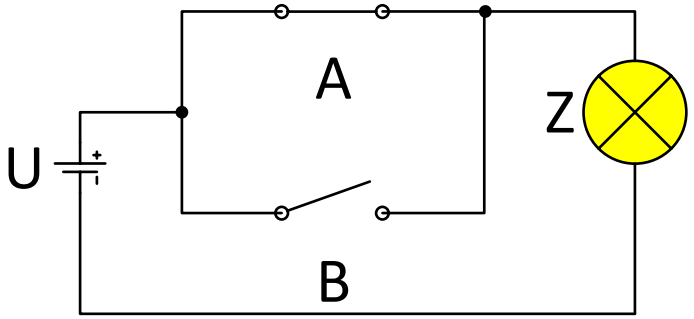
ILI-sklop

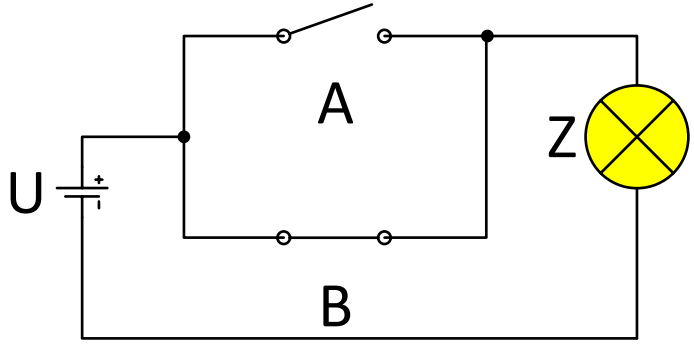
A	B	Z
LAŽ	LAŽ	LAŽ
LAŽ	ISTINA	ISTINA
ISTINA	LAŽ	ISTINA
ISTINA	ISTINA	ISTINA

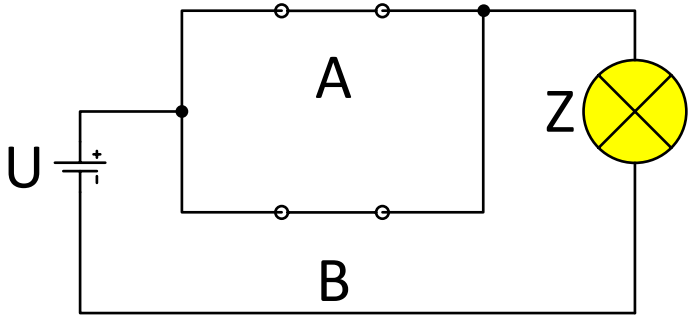
Logički operatori – Logičko ILI

- Primjer rada logičkog ILI korištenjem paralelnog spoja sklopki









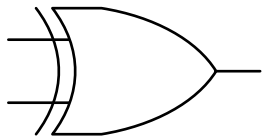
Logički operatori – Logičko ILI

- Prometna analogija
  - Korištenje dva moda transporta koje je moguće kombinirati istovremeno
  - Na predavanje mogu stići ako idem pješke ili ako idem biciklom
    - Cijelu rutu je moguće propješačiti
    - Cijelu rutu je moguće prijeći biciklom
    - Ruta sadrži dionice gdje je potrebno gurati bicikl ili se koristi bicikl bez pedala
      - Pješački prijelaz bez biciklističke staze
      - Istovremeno se pješači i vozi bicikl

Logički operatori – Logičko isključivo ILI

- Engleski naziv je XOR
- Predstavlja funkciju detekcije razlike
  - Izlaz je istinit ako su ulazne vrijednosti različite

$$Z = A \; XOR \; B$$



Isključivo-ILI-sklop

A	B	Z
LAŽ	LAŽ	LAŽ
LAŽ	ISTINA	ISTINA
ISTINA	LAŽ	ISTINA
ISTINA	ISTINA	LAŽ

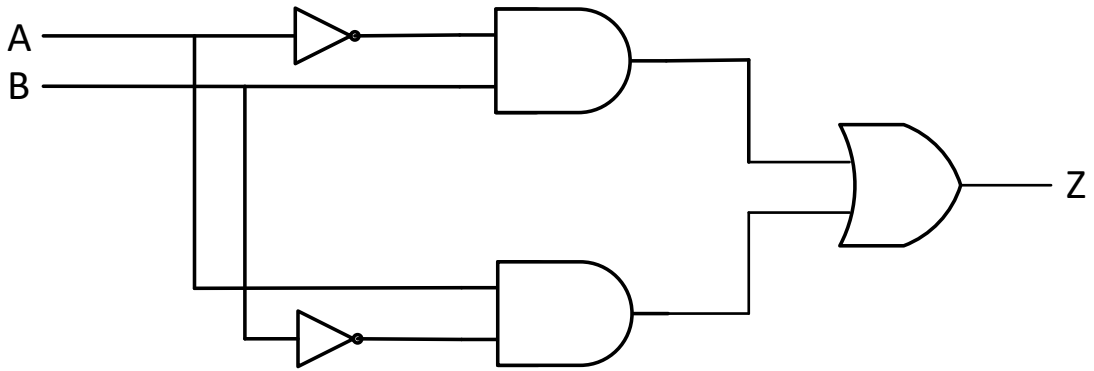
Logički operatori – Logičko isključivo ILI

- Prometna analogija
  - Korištenje dva moda transporta koje nije moguće kombinirati istovremeno
  - Put između dva obalna grada na različitim kontinentima s aerodromima mogu prijeći avionom ili brodom
    - Cijelu rutu je moguće preletjeti
    - Cijelu rutu je moguće otploviti
    - Ruta ne sadrži dionice gdje je moguće biti istovremeno u avionu i u brodu

Logički operatori – Logičko isključivo ILI

- Logička funkcija isključivo ILI se može realizirati i pomoću osnovnih funkcija I, ILI i NE
  - Prvi pristup
    - Jednostavniji za objašnjenje, složeniji za implementaciju
    - Koristi se prilagodba ulaza logičkog sklopa I korištenjem logičkog sklopa NE

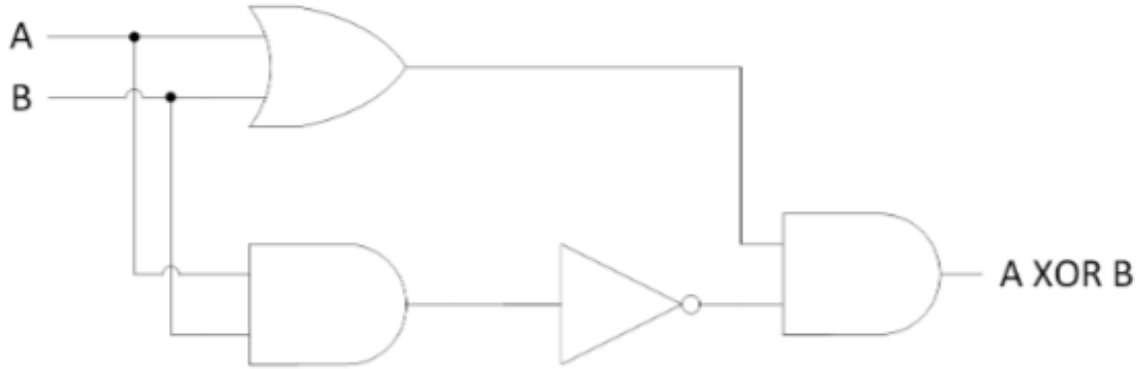
$$Z = (\bar{A}\ I\ B)\ OR\ (A\ I\ \bar{B})$$



Logički operatori – Logičko isključivo ILI

- Logička funkcija isključivo ILI se može realizirati i pomoću osnovnih funkcija I, ILI i NE
  - Drugi pristup
    - Složeniji za objašnjenje, jednostavniji za implementaciju kao elektronički sklop
      - Jedan invertor (sklop logičko NE) manje

$$Z = (A\ ILI\ B)\ I\ \overline{(A\ I\ B)}$$



Logički izrazi

- Primjeri

(1 XOR 1) AND 1 = 0 AND 1 = 0

(1 XOR 1) AND 1 OR 1 = 0 AND 1 OR 1 = 1

(0 XOR 1) AND 1 OR 1 = 1 AND 1 OR 1 = 1

(1 XOR 1) AND 1 OR 0 = 0 AND 1 OR 0 = 0

NOT 1 AND (1 XOR 0) = 0 AND 1 = 0

Pisanje izraza

- Računala naredbe izvršavaju slijedno i izraze izračunavaju u skupini po dva operanda (varijable)
  - Prioritet operatora utječe na redoslijed izvršavanja
  - Zagrade mijenjaju redoslijed izvršavanja
- Prioritet izračuna
  - Prvo se izračuna izraz u zagradi
  - Zatim unarni operatori pa množenje i dijeljenje
  - Nakon toga zbrajanje i oduzimanje pa operatori usporedbe
  - Na kraju se izvrše logički operatori
    - Najviši prioritet ima negacija (NE), zatim logički I, te na kraju logičko ILI

Pisanje izraza – Prioriteti u C#

- Prioritet izvršavanja definiran za svaki programski jezik
  - Većinom slijede matematički definiran prioritet

Opis	Sintaksa C♯
Uvećaj nakon, umanji nakon	$i++$, $i--$
Unarni operatori (+, -, !)	i += 5
Uvećaj prije, umanji prije	$++i$, $--i$
Množenje, dijeljenje (* / %)	a * b, a / b
Zbrajanje, oduzimanje	a + b, a - b
Operatori usporedbe (<, >, <=, >=)	a > b
Operatori jednakosti (==, !=)	a == b, a != b
Logički I	a && b
Logički ILI	a || b

Pisanje izraza

- Implementacija razlomaka
  - Razlomak po definiciji znači dijeljenje brojnika s nazivnikom
  - U računalu se koristi operator dijeljenja
    - Općenito se razlomak implementira tako da se brojnik i nazivnik stave u zagrade

$$\frac{a+b}{c+d} \rightarrow (a+b)/(c+d)$$

    - Česta pogreška

$$\frac{a+b}{c+d} \rightarrow a+b/c+d \rightarrow a+\frac{b}{c}+d$$

Pisanje izraza

- Implementacija potencija
  - Koristi se naredba potenciranja
  - U programskom jeziku C# je to pomoćna metoda „Math.Pow(baza, eksponent)”
    - Ulazne varijable su tipa double

$$a^b \rightarrow Math.Pow(a,b)$$

    - Varijable mogu biti napisane u obliku izraza

$$a^{\frac{b+c}{d+e}} \rightarrow Math.Pow\left(a,\left(b+c\right)/\left(d+e\right)\right)$$

    - Moguće i korjenovanje

$$\sqrt[a]{\frac{b+c}{d+e}} \rightarrow Math.Pow\left(\left(b+c\right)/\left(d+e\right),1/a\right)$$

Pisanje izraza

- Implementacija trigonometrijskih funkcija
  - Koriste se pomoćne metode za trigonometrijske funkcije
  - Ulazna varijabla je u radijanima
    - Potrebna pretvorba iz stupnjeva u radijane
      - Koristi se već ugrađena konstanta pi

$$\sin(a) \rightarrow Math.\mathrm{Sin}(a * Math.PI \,/\, 180.0)$$

  - Izlazna vrijednost kod arkus funkcija u radijanima
    - Potrebna pretvorba iz radijana u stupnjeve

$$arc\sin(a) \rightarrow Math.\mathrm{Asin}(a) * 180.0 \,/\, Math.PI$$

Pisanje izraza - Primjer

- Potrebno je izračunati duljinu hipotenuze pravokutnog trokuta ako su poznate obje katete

- Rješenje
  - Pitagorin poučak

$$c = \sqrt{a^2 + b^2}$$

    - Raptor

$$c \leftarrow SQRT\left(a \wedge 2 + b**2\right)$$

    - C#

$$c = Math.Sqrt\left(Math.Pow(a, 2.0) + b*b\right);$$

Pisanje izraza - Primjer

- Potrebno je izračunati površinu trokuta ako su poznate duljine njegovih stranica

- Rješenje
  - Heronova formula

$$s = \frac{a+b+c}{2}$$

$$P = \sqrt{s(s-a)(s-b)(s-c)}$$

  - Raptor

$$s \leftarrow (a+b+c)/2 \quad P \leftarrow SQRT(s*(s-a)*(s-b)*(s-c))$$

  - C#

$$s = (a+b+c)/2;$$

$$P = Math.Sqrt(s*(s-a)*(s-b)*(s-c));$$

Pisanje izraza - Primjer

- Potrebno je izračunati duljinu treće stranice trokuta ako su poznate duljine dviju stranica i kut između njih u stupnjevima

- Rješenje
  - Kosinusov poučak

$$c = \sqrt{a^2 + b^2 - 2ab\cos\gamma}$$

  - Raptor

$$c \leftarrow SQRT\left(a\,\hat{}\,2 + b**2 - 2*a*b*\cos\left(\gamma * pi/180\right)\right)$$

  - C#

$$c = Math.Sqrt\left(Math.Pow(a, 2.0) + b*b - 2.0*a*b*Math.Cos\left(\gamma * Math.PI/180.0\right)\right);$$

Osnove pseudokôda

- Prije pisanja programa potrebna je skica ideje rješenja
  - Naročito kod izrade većih i složenijih programa
- Pretpostavka je prethodno proučen problem te sastavljen popis varijabli
  - Varijabla nam služi za pohranu vrijednosti u memoriju
  - Predstavlja zamjenu za fizičku adresu memorijske lokacije
- Skica idejnog rješenja se radi pomoću pseudokôda i dijagrama toka

Osnove pseudokôda

- Pseudokôd je program ili algoritam napisan lako razumljivim riječima
  - Pseudo -> grčki za laž, lažni, nadri, nazovi, tobožnji
  - Kôd -> skup dogovorenih znakova za oblikovanje poruke odnosno ideje programa
- Moguće napisati slijed naredbi kao koncept programa
  - Vrijedi za bilo koji programski jezik
  - Oponašanje redoslijeda stvarnih naredbi nekog programskog jezika
  - Prilikom programiranja se koncept prepisuje u pripadne naredbe (grafičke ili tekstualne)

Osnove pseudokôda

- Napisani pseudokôd se jednostavno može prenositi za implementaciju
  - U multinacionalnom okruženju se preporuča korištenje engleskog jezika
- Koncepti algoritama se prikazuju pseudokôdom
- Za označavanje pojedinih operacija koriste se jednostavne ključne riječi
  - Unos, ispis, ako je, inače, onda, ponavljati, do, za, …
- Za pojedine standardne strukture programa postoje preporučeni koncept pisanja
  - Grananja, skretnice, petlje, unos i ispis varijabli

Osnove pseudokôda

- Radi lakšeg praćenja napisanog pseudokôda koriste se uvlake
- Uvlakama se označava pripadnost pojedinih naredbi ili varijabli određenoj ključnoj riječi ili nadređenoj naredbi
  - Prvo se navede ključna riječ, a zatim u novom redu uz uvlaku dolaze daljnje naredbe ili varijable
- Primjer

```
Ključna riječ
        Varijabla
Ključna riječ
        Naredba
```

Osnove pseudokôda

- Uvlake automatski vizualno izdvajaju pojedine programske blokove pridružene pripadnim ključnim riječima
  - Pretpostavlja se da uvučeni pseudokôd pripada prvoj prethodnoj neuvučenoj ključnoj riječi
  - Programski blok može sadržavati jednu ili više naredbi
- Uvlake su naročito korisne kod implementacija struktura grananja i petlji
  - Tim strukturama je općenito pridruženo nekoliko programskih blokova

Osnove pseudokôda – Ispis

- Koriste se lako razumljive univerzalne naredbe u bilo kojem jeziku
  - Nema stroge norme
  - Za prosljeđivanje poruke na zaslon računala koristi se naredba „Ispis”
- Primjer osnovnog programa „Hej svijete!”
  - Ispis poruke „Hej svijete!” na zaslonu računala

```
Početak programa
    Ispis
        ”Hej svijete!”
Kraj programa
```

Osnove pseudokôda – Ispis

- Kod ispisa vrijednosti varijable dobro je generirati poruku za objašnjenje
  - Za povezivanje fiksnog dijela poruke i vrijednosti varijable koristi se operator nadovezivanja „+” radi spajanja niza znakova i vrijednosti varijable (tzv. proces konkatenacije)
- Primjer ispisa vrijednosti varijable
  - Ispis vrijednosti varijable na zaslonu računala uz pripadnu poruku

```
Početak programa
    Deklaracija i inicijalizacija varijabli
        ocjena := 5
    Ispis
        ”Moja ocjena iz predmeta Računalstvo je ” + ocjena
Kraj programa
```

Osnove pseudokôda – Unos

- Za unos podataka koristi se naredba „Unos”
  - Potrebno je definirati ime varijable u koju će se spremiti vrijednost
- Primjer programa za ispis poruke koju je operater unio

```
Početak programa
    Unos
        poruka
    Ispis
        poruka
Kraj programa
```

Osnove pseudokôda – Izračuni

- Za izvršavanje aritmetičko-logičkih operacija koristi se naredba „Izračunaj”
  - Rezultat izračuna se uvijek sprema u varijablu
  - U jednu varijablu se može spremiti samo jedna vrijednost
- Primjer programa za zbrajanje dva broja

```
Početak programa
    Unos
        prviPribrojnik
        drugiPribrojnik
    Izračunaj
        zbroj := prviPribrojnik + drugiPribrojnik
    Ispis
        zbroj
Kraj programa
```

Osnove pseudokôda – Izračuni

- Aritmetičko-logičke operacije se implementiraju kao izrazi korištenjem pripadnih operatora
- Za označavanje operatora u pseudokôdu postoje preporuke
- Kod pisanja izraza u pseudokôdu vrijede sva pripadna matematička pravila prioriteta i izvršavanja pojedinih funkcija
- Bitno je u svakom programu rezultat izračuna izraza spremiti u varijablu
  - Samo tako se rezultat izračuna može koristiti dalje u programu

Osnove pseudokôda – Izračuni

- Preporuke za aritmetičke operatore

Opis	Pseudo jezik	Raptor	C#
Zbrajanje	+	+	+
Oduzimanje	-	-	-
Množenje	*	*	*
Dijeljenje	/	/	/
Cjelobrojno dijeljenje	div	div	/
Ostatak cjelobrojnog dijeljenja	mod	mod	%

- Preporuke za relacijske operatore

Opis	Pseudo jezik	Raptor	C#
Manje	<	<	<
Manje ili jednako	<=	<=	<=
Veće	>	>	>
Veće ili jednako	>=	>=	>=
Jednako	=	=	==
Različito	<>	<>	!=

Osnove pseudokôda – Izračuni

- Preporuka za logičke operatore

Opis	Pseudo jezik	Raptor	C#
Logički NE	NE	NOT	!
Logički I	I	AND	&&
Logički ILI	ILI	OR	||

- Preporuka za standardne funkcije

Opis	Pseudo jezik	Raptor
Apsolutna vrijednost realnog broja	Abs(x)	Abs(x)
Kvadrat broja	Sqr(x)	Sqr(x)
Drugi korijen realnog broja	Sqrt(x)	Sqrt(x)
Zaokruživanje realnog broja na najbliži cijeli broj	Round(x)	Round(x)
Cijeli dio realnog broja x	Trunc(x)	Trunc(x)