



Programiranjem do igre, igrom do programiranja

Informatički klub FUTURA
Dubrovnik, studeni 2013.



Creative Commons



☐ slobodno smijete:

- **dijeliti** — umnožavati, distribuirati i javnosti priopćavati djelo
- **remiksirati** — prerađivati djelo



☐ pod sljedećim uvjetima:

- **imenovanje.** Morate priznati i označiti autorstvo djela na način kako je specificirao autor ili davatelj licence (ali ne način koji bi sugerirao da Vi ili Vaše korištenje njegova djela imate njegovu izravnu podršku).
- **nekomercijalno.** Ovo djelo ne smijete koristiti u komercijalne svrhe.
- **dijeli pod istim uvjetima.** Ako ovo djelo izmijenite, preoblikujete ili stvarate koristeći ga, preradu možete distribuirati samo pod licencom koja je ista ili slična ovoj.



U slučaju daljnjeg korištenja ili distribuiranja morate drugima jasno dati do znanja licencne uvjete ovog djela. Najbolji način da to učinite je linkom na ovu internetsku stranicu.

Od svakog od gornjih uvjeta moguće je odstupiti, ako dobijete dopuštenje nositelja autorskog prava.

Ništa u ovoj licenci ne narušava ili ograničava autorova moralna prava.

Programski jezik Scratch

<http://scratch.mit.edu>

Create stories, games, and animations
Share with others around the world



A creative learning community with **4,158,693** projects shared

Programski jezik Scratch

- U programskom jeziku SCRATCH moguće je programirati i pokretati programe koristeći samo web preglednik
 - Međutim, za razvoj programa bolje je instalirati *offline editor*:
<http://scratch.mit.edu/scratch2download/>
- Potrebno je pritisnuti INSTALL NOW i slijediti upute.

Napomena: na računalima u ovoj učionici već je instaliran *offline editor* - zato je ovo uputa kako ćete to napraviti kod kuće!

Zadatak!

- U programskom jeziku SCRATCH napraviti igru za pogađanje zamišljenog broja između 1 i 10.
 - Broj “zamišlja” računalo, a mi pogađamo. Kod svakog pokušaja računalo nam javi da li je pokušaj bio iznad ili ispod zamišljenog broja. Kad pogodimo “zamišljeni” broj, potrebno je ispisati broj pokušaja



Pristup rješenju

→ 1. Pročitaj zadatak i shvati ga

☐ **Ne idi dalje dok nisi shvatio zadatak!**

2. Skiciraj rješenje

- ☐ Blok dijagramom,
- ☐ Pseudokodom,
- ☐ Slobodnim tekstom,...

3. Izaberi alat (programski jezik)

4. Programiraj u **malim koracima**

→ 1. Isprogramiraj mali komadić koda

2. Itestiraj napisani komadić koda

→ 3. Ako je do tada napisani kod u redu, dodaj novi



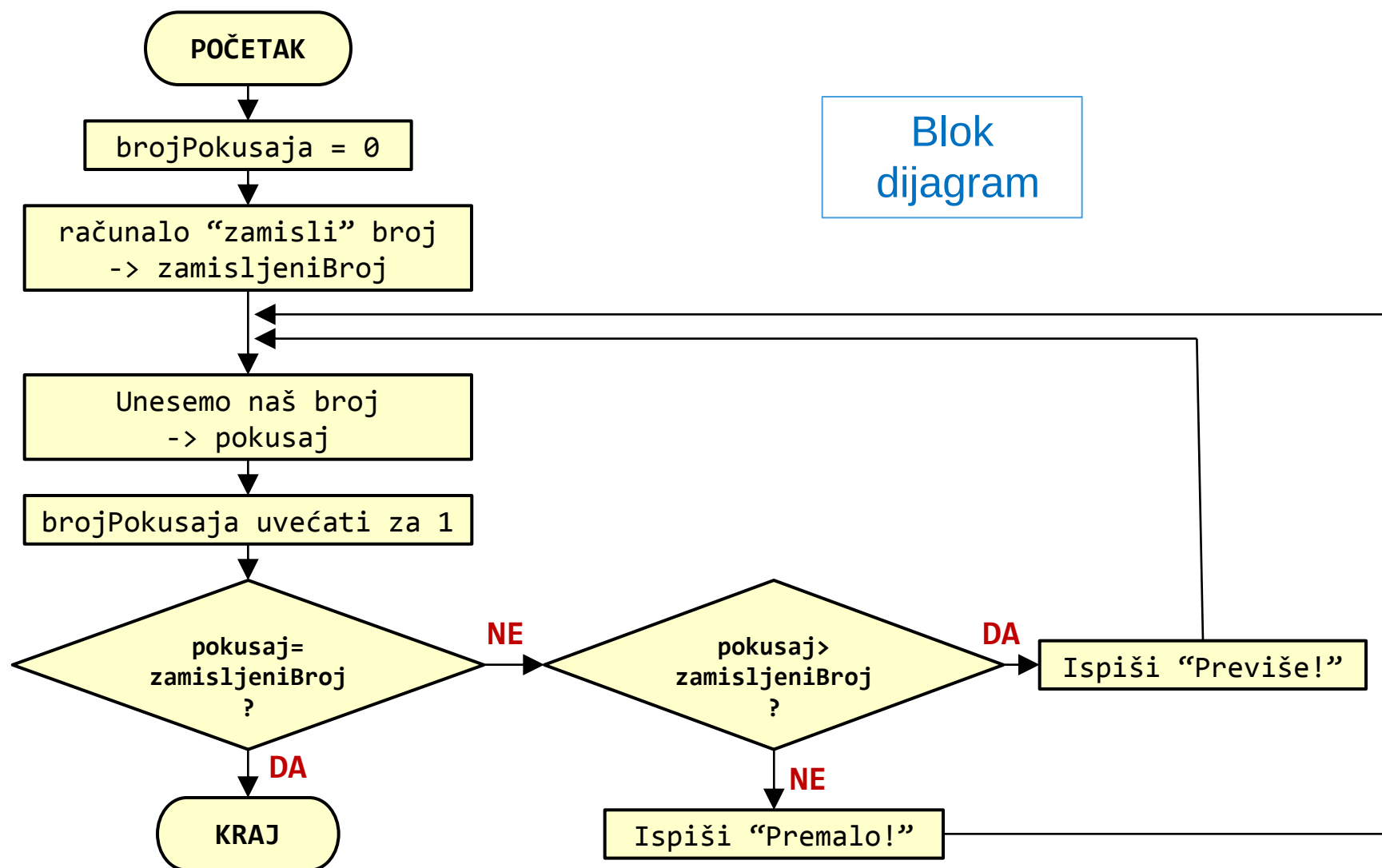
Plan rješavanja zadatka

1. Računalo “zamisli” broj između 1 i 10
2. Broj pokušaja = 0
3. Upisati broj (naš pokušaj pogađanja zamišljenog broja)
4. Broj pokušaja uvećati za 1
5. Ako smo pogodili broj ispisati poruku “Bravo!” i broj pokušaja. Prekinuti daljnje izvođenje programa.
6. Ako broj nije pogodan onda ispisati “Premalo!” ako je naš broj manji od zamišljenog, ili “Previše!” ako je naš broj veći od zamišljenog.
7. Ponavljati korake 3.-6.

Plan rješavanja
zadatka napisan
slobodnim tekstom



Plan rješavanja zadatka



Variable

□ **varijabla = promjenjivica** 😊

- **varijabla** ima **ime** i **vrijednost**
- to je **memorijska lokacija** kojoj pristupamo preko **imena**, a na kojoj se nalazi **podatak** koji se može mijenjati

Ime varijable	Podatak
brojPokusaja	5
prezime	Perić
prosjecna_ocjena	4.6
datumRod	20.1.2000.

Napomena: provjeriti za konkretni programski jezik pravila za nazive varijabli! Ne koristiti “naša” slova u nazivima varijabli!

Varijable

brojPokusaja = 0

korak = 3

Ime varijable	Podatak
brojPokusaja	0
korak	3

brojPokusaja = brojPokusaja + 2

Ime varijable	Podatak
brojPokusaja	2
korak	3

brojPokusaja = brojPokusaja * korak

Ime varijable	Podatak
brojPokusaja	6
korak	3

Ne davati ovakve
nazive varijablama:

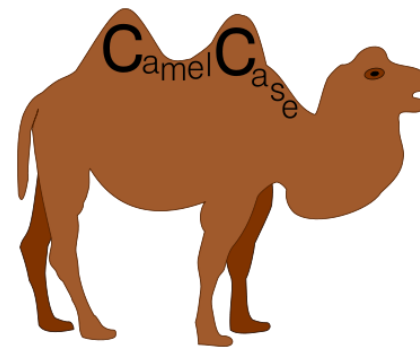
aaaa

hkgftuizf

x456

Zar nije jasnije:

brojPokusaja
zamisljeniBroj
ime_ucenika
imeNajPrij



Programiranje u Scratch-u

Scratch 2 Offline Editor

File Edit Tips

Offline Editor (beta)

pogodi-broj

pokusaj 0 odgovor "??"

brojPokusaja 1

Scripts

Costumes

Sounds

Motion

Looks

Sound

Pen

Data

Events

Control

Sensing

Operators

More Blocks

Make a Variable

brojPokusaja

odgovor

pokusaj

zamisljeniBroj

set odgovor to 0

change odgovor by 1

show variable odgovor

hide variable odgovor

Make a List

when this sprite clicked

play sound meow

hide variable zamisljeniBroj

set odgovor to "??"

set pokusaj to 0

set zamisljeniBroj to pick random 1 to 10

set brojPokusaja to 0

repeat until zamisljeniBroj = 0

set brojPokusaja to brojPokusaja + 1

ask Reci broj između 1 i 10 and wait

set pokusaj to answer

if pokusaj = zamisljeniBroj then

change color effect by 25

set odgovor to "Bravo, pogodio si!"

say odgovor

show variable zamisljeniBroj

repeat 10

play drum 8 for 0.1 beats

set zamisljeniBroj to 0

else

if pokusaj > zamisljeniBroj then

set odgovor to "Preeviše"

play drum 1 for 0.25 beats

say odgovor for 2 secs

else

set odgovor to "Premalo"

play drum 2 for 0.25 beats

say odgovor for 2 secs

Sprites

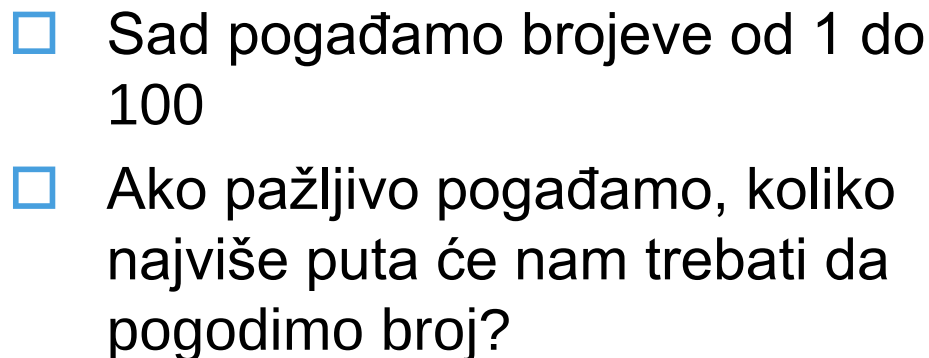
New sprite:

Stage

2 backdrops

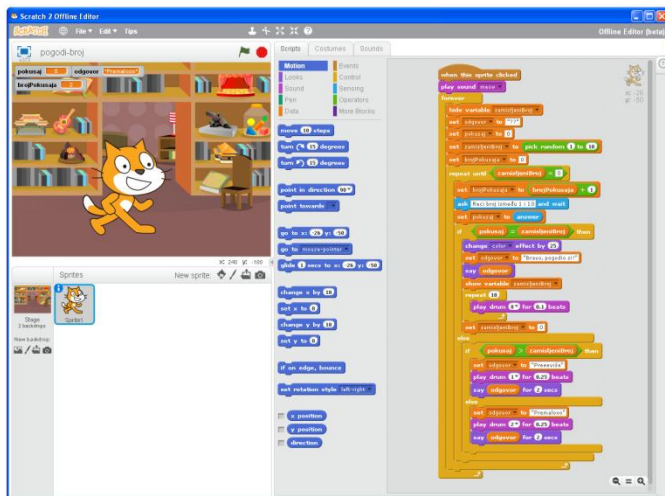
New backdrop:

Sprite1



1-5 pokušaja -> “Baš imaš sreće”
6-7 pokušaja -> “Jako dobro”
8-9 pokušaja -> “Može to bolje”
više od 9 pokušaja -> “Ajme meni”

Malo zabave za doma!



- Vidjeli smo da nam treba najviše 7 pokušaja za pogoditi zamišljeni broj **između 1 i 100**
- Kako bi pogađanje išlo računalu? Na temelju prethodnog programa napisati novi program u kojem će računalo pogađati **broj koji smo mi zamislili**.
- Kad računalo ponudi svoj broj, naš odgovor može biti:

Rješenja („sb2” datoteke)
šaljite na mail adresu:
unidu.prog@gmail.com

- -> premalo
- + -> previše
- 0** -> broj je pogođen

A sad malo drugog programskog jezika ☹️

- ❑ Nažalost na školskim natjecanjima još uvijek se ne može koristiti Scratch

www.infokup.hr



Infokup > Kategorije > Algoritmi

Algoritmi

Računalni algoritam je niz naredbi koji efikasno rješava neki problem. U ovoj kategoriji cilj je upravo osmisliti algoritam koji rješava zadani problem.

Međutim, zadani problemi su jako često složeni, pa je potrebno kombinacijom postojećih i osmišljavanjem novih algoritama doći do efikasnog rješenja.

Natjecanja

Rezultate, zadatke, rješenja i službene propozicije natjecanja iz kategorije Algoritmi možeš pregledati koristeći sljedeće linkove:

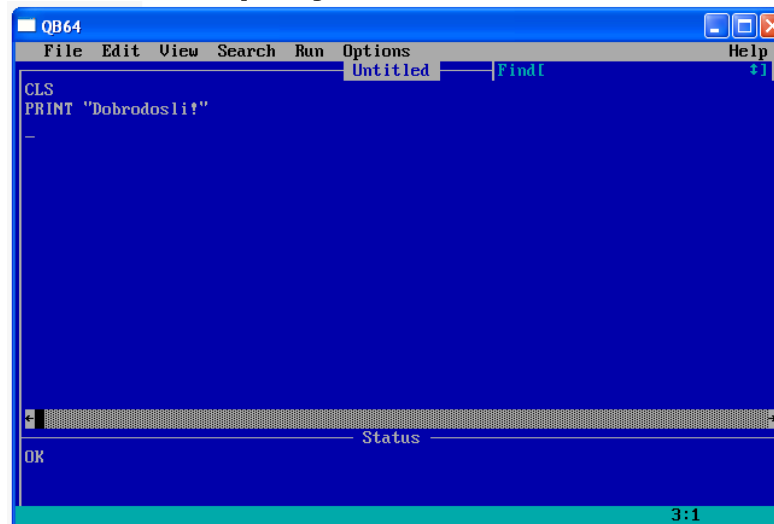
Osnovna škola LOGO Basic/C/C++/Pascal	Srednja škola C/C++/Pascal (1.) C/C++/Pascal (2.) C/C++/Pascal (3.) C/C++/Pascal (4.)	Propozicije Službene propozicije (1.8 MB, .pdf)
--	--	--

A sad malo drugog programskog jezika ☹️

- ❑ Zato ćemo se malo baviti i **Basic**-om
- ❑ Koristiti ćemo **QB64** inačicu:

www.qb64.net/qb64v0980-win.7z

Nakon što se arhiva raspakira, nema posebne instalacije, već je samo potrebno napraviti prečac (*shortcut*) do datoteke **qb64.exe** (koja se nalazi u mapi **qb64**).



Programski jezik Basic



- ❑ Zadatak je isti - treba napraviti program za pogađanje brojeva između 1 i 100
- ❑ Već smo govorili da je najbolje programirati u malim koracima. Kako bi to izgledalo na ovom primjeru?

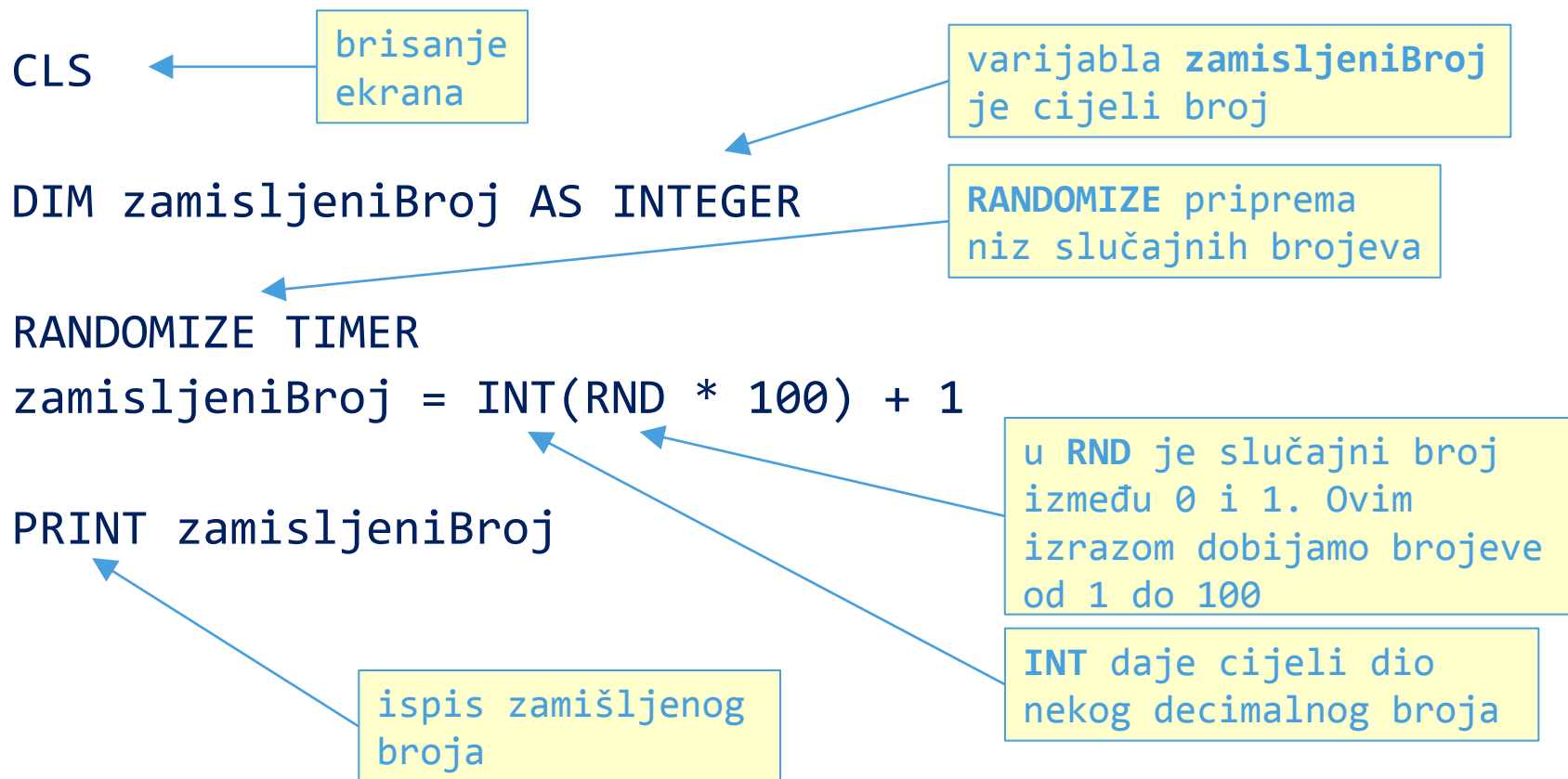
1. Isprogramiraj mali komadić koda
2. Istestiraj napisani komadić koda
3. Ako je do tada napisani kod u redu, dodaj novi





Programski jezik Basic

□ 1. korak - računalo “zamišlja broj:





Programski jezik Basic

□ 2. korak - unosimo naš pokušaj:

CLS

DIM zamisljeniBroj AS INTEGER

DIM pokusaj AS INTEGER

varijabla **pokusaj** je cijeli broj

RANDOMIZE TIMER

zamisljeniBroj = INT(RND * 100) + 1

PRINT zamisljeniBroj

naredbom **INPUT** unosimo podatak s tipkovnice, i to u varijablu **pokusaj**

INPUT "Pogodi broj (1-100)! ", pokusaj

IF naredbom ispitujemo da li je neki izraz istinit

IF pokusaj = zamisljeniBroj THEN

PRINT "Bravo - pogodio si!"

END IF

ovaj **PRINT** se izvodi samo ako je istina da je **pokusaj = zamisljeniBroj**



Programski jezik Basic

□ 2. korak - unosimo naš pokušaj:

```
CLS
```

```
DIM zamisljeniBroj AS INTEGER
```

```
DIM pokusaj AS INTEGER
```

```
RANDOMIZE TIMER
```

```
zamisljeniBroj = INT(RND * 100) + 1
```

```
PRINT zamisljeniBroj
```

naravno da u konačnoj verziji programa
nećemo ostaviti prikaz zamišljenog broja ;)

```
INPUT "Pogodi broj (1-100)! ", pokusaj
```

```
IF pokusaj = zamisljeniBroj THEN
```

```
    PRINT "Bravo - pogodio si!"
```

```
END IF
```



Programski jezik Basic

□ 3. korak - da li je naš pokušaj iznad ili ispod:

CLS

DIM zamisljeniBroj AS INTEGER

DIM pokusaj AS INTEGER

RANDOMIZE TIMER

zamisljeniBroj = INT(RND * 100) + 1

PRINT zamisljeniBroj

INPUT "Pogodi broj (1-100)! ", pokusaj

IF pokusaj = zamisljeniBroj THEN

PRINT "Bravo - pogodio si!"

ELSEIF pokusaj > zamisljeniBroj THEN

PRINT "Previše!"

ELSE

PRINT "Premalo!"

END IF

```
IF prvi_uvjet THEN
    ....
    izvodi se ako je
    ispunjen prvi_uvjet, i
    zatim izlaz iza END IF
    ....
ELSEIF drugi_uvjet THEN
    ....
    izvodi se ako je
    ispunjen drugi_uvjet, i
    zatim izlaz iza END IF
    ....
ELSE
    ....
    izvodi se jedino ako
    nisu ispunjeni
    prethodni uvjeti
```



Programski jezik Basic

- 4. korak - moramo u petlji ponavljati unos broja - dok ne pogodimo:

CLS

DIM zamisljeniBroj AS INTEGER

DIM pokusaj AS INTEGER

RANDOMIZE TIMER

zamisljeniBroj = INT(RND * 100) + 1

PRINT zamisljeniBroj

INPUT "Pogodi broj (1-100)! ", pokusaj

IF pokusaj = zamisljeniBroj THEN

 PRINT "Bravo - pogodio si!"

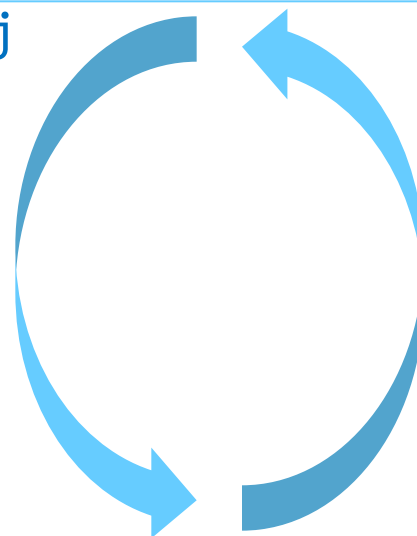
ELSEIF pokusaj > zamisljeniBroj THEN

 PRINT "Previše!"

ELSE

 PRINT "Premalo!"

END IF





Programski jezik Basic

- 4. korak - moramo u petlji ponavljati unos broja - dok ne pogodimo:

CLS

DIM zamisljeniBroj AS INTEGER

DIM pokusaj AS INTEGER

RANDOMIZE TIMER

zamisljeniBroj = INT(RND * 100) + 1

DO

INPUT "Pogodi broj (1-100)! ", pokusaj

IF pokusaj = zamisljeniBroj THEN

PRINT "Bravo - pogodio si!"

EXIT DO

ELSEIF pokusaj > zamisljeniBroj THEN

PRINT "Previše!"

ELSE

PRINT "Premalo!"

END IF

LOOP

DO

....

LOOP

"beskonačna" petlja

EXIT DO

izlaz iz "beskonačne"
petlje



Programski jezik Basic

- 5. korak - još dodamo brojač pokušaja:

CLS

```
DIM zamisljeniBroj AS INTEGER
```

```
DIM pokusaj AS INTEGER
```

```
DIM brojPokusaja AS INTEGER
```

```
brojPokusaja = 0
```

```
RANDOMIZE TIMER
```

```
zamisljeniBroj = INT(RND * 100) + 1
```

```
DO
```

```
    INPUT "Pogodi broj (1-100)! ", pokusaj
```

```
    brojPokusaja = brojPokusaja + 1
```

```
    PRINT "Ovo je "; brojPokusaja; ". pokusaj"
```

```
    IF pokusaj = zamisljeniBroj THEN
```

```
        PRINT "Bravo - pogodio si broj iz "; brojPokusaja; ". pokusaja!"
```

```
        EXIT DO
```

```
    ELSEIF pokusaj > zamisljeniBroj THEN
```

```
        PRINT "Previše!"
```

```
    ELSE
```

```
        PRINT "Premalo!"
```

```
    END IF
```

```
LOOP
```



Još malo zabave za doma!

- Isto kao za programski jezik Scratch:

Na temelju prethodnog programa, u Basicu napisati novi program u kojem će računalo pogađati **broj koji smo mi zamislili**.

```
CLS
DIM zamisljeniBroj AS INTEGER
DIM pokusaj AS INTEGER
DIM brojPokusaja AS INTEGER
brojPokusaja = 0

RANDOMIZE TIMER
zamisljeniBroj = INT(RND * 100) + 1

DO
    INPUT "Pogodi broj (1-100)! ", pokusaj
    brojPokusaja = brojPokusaja + 1
    PRINT "Ovo je "; brojPokusaja; ". pokusaj"
    IF pokusaj = zamisljeniBroj THEN
        PRINT "Bravo - pogodio si broj iz "; brojPokusaja; ". pokusaja!"
    ELSEIF pokusaj > zamisljeniBroj THEN
        PRINT "Preveliko!"
    ELSEIF pokusaj < zamisljeniBroj THEN
        PRINT "Premalo!"
    END IF
    EXIT DO
LOOP UNTIL pokusaj = zamisljeniBroj

PRINT "Status"
```

Rješenja („bas“ datoteke)
šaljite na mail adresu:
unidu.prog@gmail.com

Zadaci na natjecanjima

- Zadaci na natjecanjima najčešće su zadani u obliku priče koja bi sudionicima natjecanja trebala biti zabavna, zanimljiva i poticajna
 - Priča opisuje problemsko područje i postavlja okvir za rješavanja zadatka
 - Naglašeni su specifični uvjet i ograničenja
 - Detaljno je opisan oblik ulaznih podataka i izlaznih rezultata
 - Obavezno je navedeno nekoliko ulaznih podataka i očekivanih izlaza
 - **OPREZ!** Navedeni podaci za testiranje često ne pokrivaju SVE moguće situacije

Primjer zadatka (6.razred, 2.zadatak)

Da vidite da se iza puno teksta krije lagan zadatak ...

Melkior, unuk profesora Baltazara je još početkom školske godine obećao svojim prijateljima iz razreda napraviti program koji će im na kraju polugodišta i školske godine olakšati određivanje ukupne zaključne ocjene. Zato je Melkior odlučio iskoristiti zimske praznike i u čarobnom laboratoriju svog djeda kreirati obećani program.

Program treba na osnovu zadanog broja zaključenih petica, četvorki, trojki, dvojki i jedinica iz svih nastavnih predmeta odrediti i ispisati **prosječnu vrijednost (prosjek) svih zaključenih ocjena** te dodatno ispisati odgovarajuću ukupnu zaključnu ocjenu u opisnom obliku.

... nastavlja se =>

Primjer zadatka (6.razred, 2.zadatak)

...

Znamo da se ukupna zaključna ocjena “**nedovoljan**” zaključuje kada je učenik imao barem jednu zaključenu jedinicu. Inače se ukupna zaključna ocjena zaključuju na sljedeći način: “**odličan**” se zaključuje kada je prosjek dobivenih zaključnih ocjena iz svih predmeta između 4.50 i 5.00, ocjena “**vrlo dobar**” za prosjek veći ili jednak od 3.50 i strogo manji od 4.50, ocjena “**dobar**” za prosjek veći ili jednak od 2.50 i strogo manji od 3.50 te ocjena “**dovoljan**” za prosjek veći ili jednak od 2.00 i strogo manji od 2.50.

□ ***Napomena: u Melkiorovom razredu će uvijek biti barem jedan nastavni predmet.***

... nastavlja se =>

Primjer zadatka (6.razred, 2.zadatak)

...

Ulaz

- ☐ prirodan broj **O1** ($0 \leq \mathbf{O1} \leq 10$), broj zaključenih jedinica;
- ☐ prirodan broj **O2** ($0 \leq \mathbf{O2} \leq 10$), broj zaključenih dvojki;
- ☐ prirodan broj **O3** ($0 \leq \mathbf{O3} \leq 10$), broj zaključenih trojki;
- ☐ prirodan broj **O4** ($0 \leq \mathbf{O4} \leq 10$), broj zaključenih četvorki;
- ☐ prirodan broj **O5** ($0 \leq \mathbf{O5} \leq 10$), broj zaključenih petica;

Izlaz

- ☐ u prvi redak izlaza treba ispisati prosječnu vrijednost svih zaključenih ocjena ispisanu na **dvije decimale**;
- ☐ u drugi redak treba ispisati jednu od sljedećih poruka: “odlican”, “vrlo dobar”, “dobar”, “dovoljan”, “nedovoljan”.

... nastavlja se =>

Primjer zadatka (6.razred, 2.zadatak)

... primjeri

Primjer 1

Ulaz

0

2

4

5

5

Izlaz

3.81

vrlo dobar

Primjer 2

Ulaz

2

3

4

5

5

Izlaz

3.42

nedovoljan

Primjer zadatka (6.razred, 2.zadatak)

... u zadatku su 2 primjera, ali će se kod ocjenjivanja program testirati s još podataka. Za očekivati je da će se neki od tih ulaznih podataka baviti graničnim slučajevima – npr. koja je zaključna ocjena ako je prosjek 3.50?

Zato je dobro ako sami smislite takav testni primjer:

Naš primjer

Ulaz

0

3

6

6

3

Izlaz

3.50

vrlo dobar

Tekst zadatka možete naći na:

<http://www.unidu.hr/odjeli.php?idizbornik=824>

Probajte samostalno riješiti!

Na dodatnoj grupi iz predmeta „Uvod u magiju“, Harry je pokušao riješiti jedan težak matematičko-magijski zadatak. On je dobio list papira na kome su bila napisana tri jednoznamenasta broja odvojena s po jednim razmakom.

Oko papira su letjela tri matematička znaka, znak za zbrajanje '+', znak za množenje '*' i znak '='. Harry mora jedan od dva matematička znaka (+ ili *) te znak =, uz pomoć čarobnog štapića postaviti u praznine između brojeva tako da dobije istinit matematički izraz.

Bla, bla

Tekst zadatka možete naći na:

<http://www.unidu.hr/odjeli.php?idizbornik=825>

=> Zadatak „Harry”