

## Čas br.3

Pojmovi:

- Događaji u Processingu - funkcije
- funkcije *mousePressed()* i *keyPressed()*
- varijable

Primjer 1:

```
void setup() {  
  size(200,200);  
  background(255);  
}  
void draw() {  
}  
void mousePressed() {  
  stroke(0);  
  fill(175);  
  rectMode(CENTER);  
  rect(mouseX,mouseY,16,16);  
}  
void keyPressed() {  
  background(255);  
}
```

U ovom primjeru pored funkcija **setup()** i **draw()** pojavljuju se dvije dodatne: **mousePressed()** i **keyPressed()**. Program starta tako što se funkcijom **setup()** inicijaliziraju veličina i pozadina prozora. Dalje, funkcija **draw()** koja ne sadrži nikakav kod, dakle prozor ostaje prazan. (Podsjetimo se da funkcija **draw()** beskonačno se ponavlja.)

Kod koji se nalazi u funkcijama **mousePressed()** i **keyPressed()** čeka na upotrebu. Kada korisnik klikne miš (ili pritisne neku tipku) izvrši se blok koda unutar ovih funkcija (samo jednom).

**Vježba 1:**

Dodajte „**background(255);**“ unutar funkcije **draw()**. Zašto program prestaje raditi?

## Varijable

Kako da shvatimo pojam varijable..

Komputer posjeduje memoriju. Zašto to zovemo memorija? Zato što je to ono što računar koristi kako bi zapamtio one stvari koje su mu potrebne.

Tehnički govoreći, varijabla je imenovani pokazivač na lokaciju u kompjuterskoj memoriji („memorijska adresa“) gdje su sačuvani podaci. Kako računari mogu procesirati samo jednu po jednu instrukciju, varijabla dopušta programer da sačuva informaciju iz jednog trenutka i pozove se na nju kasnije. To je itekako korisno jer varijable mogu da sačuvaju informacije vezane za oblike: boja, veličina, lokacija. Varijable su upravo ono šta nam je potrebno kako bismo napravili da se trougao

mijenja iz plave boje u ljubičastu, da krug „leti“ po ekranu, ili pravougaonik smanjuje do trenutka nestanka.

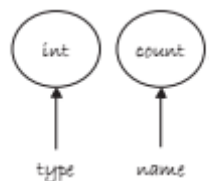
### Deklaracija i inicijalizacija varijable

Varijable mogu čuvati *primitivne* vrijednosti ili *reference na objekte ili nizove*. Za sada ćemo se samo brinuti o primitivima. Primitivne vrijednosti su gradivni blokovi podataka na kompjuteru i tipično uključuju pojedinačni komad informacije, kao što su broj ili karakter.

Varijable se deklariraju tako što se prije svega izražava *tip* a zatim *ime*. Ime varijable se mora sastojati od jedne riječi (bez razmaka) i mora počinjati s slovom (mogu uključivati brojeve, ali ne i započeti s njima). Ne mogu uključivati znakove interpunkcije ili specijalne karaktere, osim donje crtice „\_“.

Tip varijable određuje kakvi se to podaci čuvaju unutar te varijable. Ovo bi mogao biti cijeli broj, decimalni ili karakter. Ovo su podatkovni tipovi koji se najčešće koriste:

- **Cijeli brojevi**, kao što su 0, 1, 2, 3, -1, -2, i tako dalje i oni se pohranjuju kao „*integers*“ a ključna riječ je *int*.
- **Decimalni brojevi**, kao što su 3.14159, 2.5 i -9.95 čuvaju se kao *float*.
- **Karakter**, kao što su slova ‚a‘, ‚b‘, ‚c‘ i tako dalje čuvaju se kao tip *char* te se deklariraju ka slovo unutar jednostrukih navodnika, tj, ‚a‘. Karakteri su korisni kada se treba odrediti koje slovo na tipkovnici je pritisnuto i u druge svrhe kada se koriste stringovi, ili nizovi karaktera.



1 Deklaracija varijable

Na slici 1 deklarirali smo varijablu pod imenom „count“ koja je tipa „int“.

#### Svi primitivni tipovi

- *boolean*: true, false
- *char*: karakter ‚a‘, ‚b‘, ...
- *byte*: mali broj, -128 do 127
- *short*: nešto veći broj, -32768 do 32767
- *int*: veliki broj, -2147483648 do 2147483647
- *long*: stvarno veeelik broj 😊
- *float*: decimalni broj, kao što je 3.14159
- *double*: decimalni broj s mnogo više decimalnih mjesta (samo potreban za napredne programe koji zahtijevaju matematički preciznost).

Jednom kada deklariramo varijablu, možemo joj dodijeliti vrijednost njenim postavljanjem da je jednaka nečemu. U većini slučajeva, ukoliko zaboravimo inicijalizirati varijablu, Processing će joj dodijeliti defaultnu vrijednost kao što je 0 za integere, 0.0 za float i tako dalje.

```
int count;  
count = 50;
```

ili

```
int count = 50;
```

### Primjer 2:

Deklarirajte i inicijalizirajte nekoliko varijabli različitog tipa.

### *Korištenje varijabli*

### Primjer 3:

Jednostavni program koji crta krug na ekranu.

```
void setup() {  
  size(200,200);  
}  
void draw() {  
  background(255);  
  stroke(0);  
  fill(175);  
  ellipse(100,100,50,50);  
}
```

U prethodnim vježbama naučili smo kako da ovaj jednostavni primjer unaprijedimo tako što smo za centar kruga koristili trenutnu lokaciju miša (***mouseX*** i ***mouseY***).

```
ellipse(mouseX,mouseY,50,50);
```

Kao što vidite ***mouseX*** i ***mouseY*** upravo predstavljaju varijable i to systemske, „ugrađene“ Processing varijable.

Ono što ćemo dalje jeste da napravimo naše varijable slijedeću sinaksu za deklaraciju i inicijalizaciju navedenu prethodno, te postavimo ih na vrh našeg koda.

### Primjer 4:

```
int circleX = 100;  
int circleY = 100;  
void setup() {  
  size(200,200);  
}  
void draw() {  
  background(100);  
  stroke(255);  
  fill(0);  
  ellipse(circleX,circleY,50,50);  
}
```

Pokretanjem ovog koda primjećujemo da smo dobili isti rezultat, ali to je i logično jer realno nismo vrijednosti za centar kružnice ni mijenjali. Da bismo zaista iskoristili moć upotrebe varijabli trebamo mijenjati njihovu vrijednost.

Varijable uvode i mogućnost upotrebe različitih operacija pored upotrebe samih funkcija. Na primjer:

```
x = 5;  
x = a + b;  
x = y - 10 * 20;  
x = x * 5;
```

### Vježba 2:

U našem primjeru recimo da želimo kreirati krug koji se pomjera. Primjetite da smo za koordinate centra koristili (100, 100). Šta ako želimo da se krug pomjera s lijeva na desno? Koju koordinatu variramo?

Kako bi izgledao kod?

### Vježba 3:

Ovaj put želimo da centar ostane fiksna ali da se veličina (prečnik) kruga povećava. Dopunite kod:

```
int circleSize = 0;  
int circleX = 100;  
int circleY = 100;  
void setup() {  
  size(200,200);  
}  
void draw() {  
  background(0);  
  stroke(255);  
  fill(175);  
  
  _____  
  _____  
}
```

### *Više varijabli*

Ovaj put želimo pokazati kako da mijenjamo više parametara elipse, dakle ne samo centar i prečnik.

### Primjer 5:

```
float circleX = 0;  
float circleY = 0;  
float circleW = 50;  
float circleH = 100;  
float circleStroke = 255;  
float circleFill = 0;  
float backgroundColor = 255;  
float change = 0.5;  
  
// Vaše osnovne postavke  
void setup() {  
  size(200,200);
```

```

smooth();
}

void draw() {
// Nacrtaj pozadinu ili elipsu
background(backgroundColor);
stroke(circleStroke);
fill(circleFill);
ellipse(circleX, circleY, circleW, circleH);

// Promijeni vrijednosti svih varijabli
circleX = circleX + change;
circleY = circleY + change;
circleW = circleW + change;
circleH = circleH - change;
circleStroke = circleStroke - change;
circleFill = circleFill + change;
}

```

### Zadatak za domaću zadaću:

**Korak 1:** Napišite kod koji crta sljedeće slike s „tvrdo-kodiranim“ vrijednostima.

**Korak 2:** Izmijenite sve „tvrdo-kodirane“ vrijednosti s varijablama.

**Korak 3:** Upišite operacije dodjeljivanja unutar funkcije **draw()** koje mijenjaju vrijednosti varijabli. Na primjer, „varijable1 = varijable1 + 2;“. Pokušajte različite ekspresije i vidite šta će se desiti!

