

# 4. PROGRAMIRANJE

## 4.4 Potprogrami

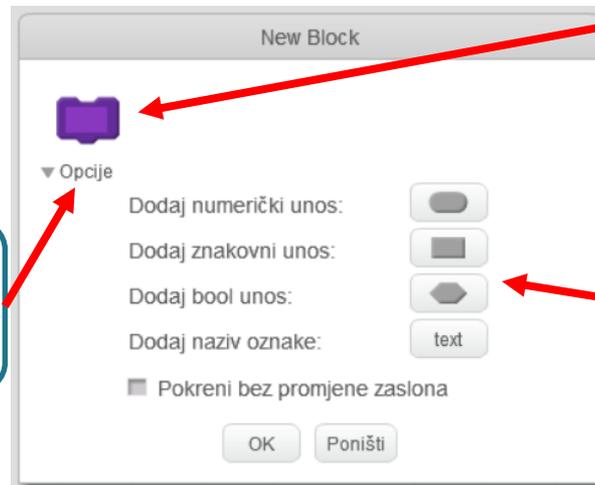
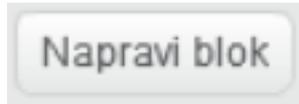


# POTPROGRAM

- Potprogram je dio programskog kôda koji je neovisan o glavnom programu.
- On se ne izvršava dok ga ne pozovemo.
- Uobičajen naziv za potprogram je **funkcija**.

# KAKO IZRADITI POTPROGRAM U SCRATCHU

- U Scratchu za izradu potprograma koristimo blok **definiraj** koji dobijemo klikom na **Napravi blok** u kategoriji **Više Blokova**.
- Nakon toga otvara se okvir:



1.

Unos naziva novog bloka

2.

Dodavanje ulaznih parametara

3.

Dodatne mogućnosti

# KAKO PROVJERITI JE LI UNESENI BROJ DJELJIV S 3 – bez parametra

1.

Glavni program – unosi se vrijednost varijable, a zatim poziva potprogram



2.

Pozivanje potprograma

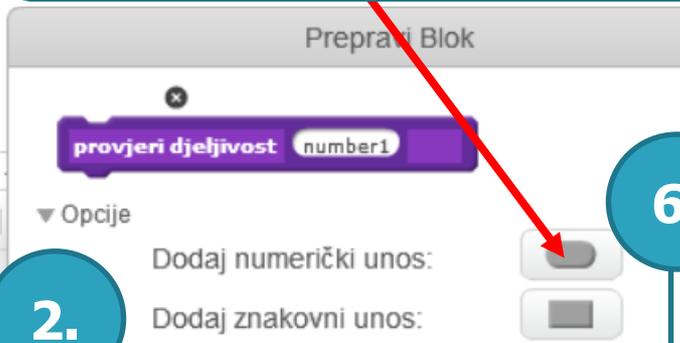
3.

Naziv potprograma



# KAKO PROVJERITI JE LI UNESENI BROJ DJELJIV S 3 – s parametrom

1. Dodavanje brojčanog parametara (ulaznog podatka)



2. Pozivanje potprograma



3. Prenesena vrijednost

4. Definiranje potprograma s parametrom



6. Blokovi naredbi potprograma

5. Prenesena će vrijednost "sjesti" u vrijednost pomoćne varijable number1, bez obzira na to što nemaju isti naziv

# IZRADA KALKULATORA ZA ZBRAJANJE I ODUZIMANJE UPORABOM POTPROGRAMA

1.

Potprogram za zbrajanje



2.

Potprogram za oduzimanje



3.

Unos brojeva nad kojima se vrši matematička operacija



4.

Unos matematičke operacije



5.

## Provjera odabira operacije



Ako je upisana operacija "zbrajanje", pozovi potprogram zbrajanje

Ako je upisana operacija "oduzimanje", pozovi potprogram oduzimanje

Ako unos nije ispravan, ispiši poruku

AL"  
mpF  
key  
(va  
lif(  
temp  
lif  
uff  
nes:  
oup  
eV+

```
kada je  kliknut  
pitaj Upiši prvi broj: i čekaj  
postavi a na odgovor  
pitaj Upiši drugi broj: i čekaj  
postavi b na odgovor  
pitaj Upiši željenu operaciju: i čekaj  
postavi operacija na odgovor  
ako operacija = zbrajanje onda  
  zbrajanje  
inače  
  ako operacija = oduzimanje onda  
    oduzimanje  
  inače  
    reci Niste unijeli ispravnu operaciju.
```

```
definiraj zbrajanje  
postavi zbroj na a + b  
reci spoji Zbroj iznosi: zbroj
```

```
definiraj oduzimanje  
postavi zbroj na a - b  
reci spoji Razlika iznosi: zbroj
```



# ZADATAK



1. Pokreni i testiraj skriptu.
2. Proširi program tako da pomoću potprograma dodaš mogućnosti množenja i dijeljenja.

# ZADATAK



- Sastavi skriptu koja omogućuje mijenjanje liste tako da ti daje izbor dodavanja ili uklanjanja elemenata na listu, a odabirom jedne od navedenih operacija poziva se odgovarajući potprogram koji izvršava odabranu operaciju.

# KAKO POSTAVITI UVJET



1.

Definiranje potprograma

2.

Određivanje broja okusa koje želiš dodati

3.

Dodavanje novih okusa na listu

# ZADATAK



- Izradi kviz s 3 različita područja (*Povijest, Geografija, Fizika*) koji svaki sadrži 3 nasumična pitanja.
- Napiši glavni program koji pomoću potprograma omogućuje izbor područja iz kojeg će biti postavljeno pitanje.

## RJEŠENJE

AL"  
mpF  
key  
(va  
lif(  
temp  
lif  
uff  
nes:  
oup  
eV+

kada je  kliknut

govori a)Geografija b)Povijest c)Fizika 2 sekundi

pitaj Odaberi svoju kategoriju: i čekaj

ako odgovor = a onda

Geografija

ako odgovor = b onda

Povijest

ako odgovor = c onda

Fizika

definiraj Geografija

postavi pitanje na slučajni broj od 1 do 3

ako pitanje = 1 onda

pitaj Koji je najviši vrh svijeta? i čekaj

ako odgovor = Everest onda

reci Točno

inače

reci Probaj ponovno!



# ZADATAK RB



- Napravi skriptu koja traži unos broja  $n$ , a zatim pomoću potprograma ***provjeri parnost*** provjerava je li broj paran ili neparan te o tome ispisuje odgovarajuću poruku.

## RJEŠENJE

kada je  kliknut

pitaj Unesi broj: i čekaj

postavi broj na odgovor

provjera

definiraj provjera

ako broj mod 2 = 0 onda

govori Broj je paran 2 sekundi

inače

govori Broj nije paran 2 sekundi

# ZADATAK RB



- Napravi skriptu koja traži unos broja  $n$ , a zatim pomoću potprograma provjerava je li broj savršen te ispisuje odgovarajuću poruku.
- Broj je savršen ako je jednak zbroju svojih djelitelja osim samog sebe.

## RJEŠENJE

AL"  
mpF  
key  
(va  
lif(  
temp  
lif  
uff  
nes:  
oup(  
eV+)

kada je  kliknut

pitaj Unesi broj: i čekaj

postavi n na odgovor

provjera

definiraj provjera

postavi zbroj na 0

postavi brojač na 0

ponavljaj dok nije  $\text{brojač} = n$

ako  $n \bmod \text{brojač} = 0$

postavi zbroj na  $\text{zbroj} + \text{brojač}$

promijeni brojač za 1

ako  $\text{zbroj} = n$  onda

govori Broj je savršen 2 sekundi

inače

govori Broj nije savršen 2 sekundi



# ZADATAK RB



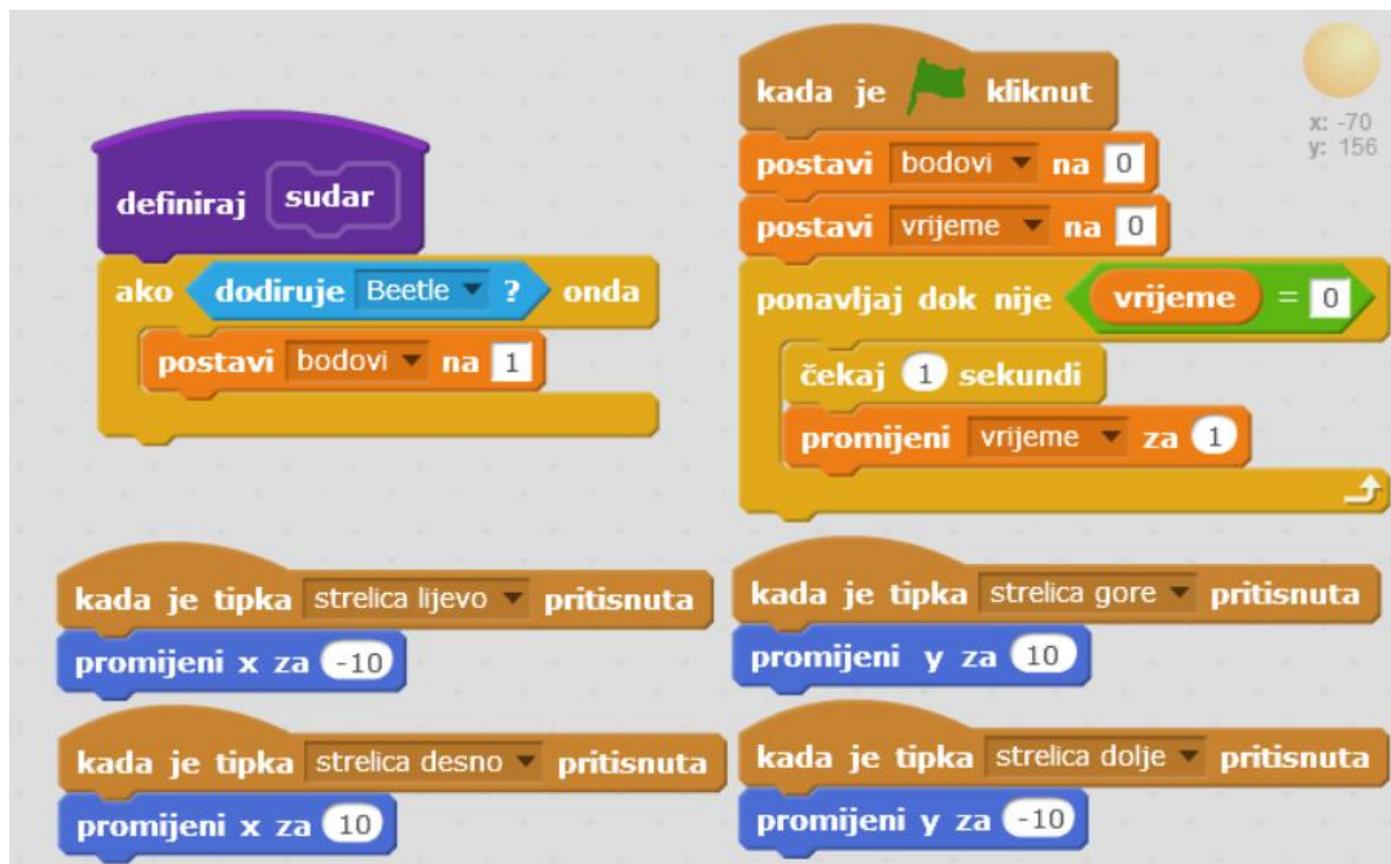
- Napravi igru u kojoj trebaš što više puta kuglom uloviti kukca do isteka vremena.
- Vrijeme unutar skripte postavi proizvoljno.
- Složi pozornicu kao na slici.
- Skripta omogućuje upravljanje likom kugle pomoću tipki za navigaciju te bilježi svaki (ulov) sudar likova.
- Nakon ulova kukac se automatski premješta na slučajnu poziciju na pozornici.
- Za provjeru sudara likova koristi potprogram *sudar*.

## RJEŠENJE

AL"  
mpF  
key  
(va  
lif(  
temp  
lif  
uff  
nes:  
oup  
eV+



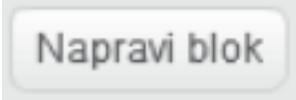
Scratch script for a green flag click event. The script starts with a 'kada je kliknut' (when green flag clicked) block, followed by a 'ponavljaj' (repeat) block. Inside the repeat block, there is an 'ako dodiruje Ball ? onda' (if Ball is touched then) block. Inside this 'if' block, there is an 'idi na x: slučajni broj od -240 do 240 y: slučajni broj od 180 do -180' (go to x: random number from -240 to 240 y: random number from 180 to -180) block.



Scratch script for a Beetle collision event. The script starts with a 'definiraj sudar' (define collision) block. Inside, there is an 'ako dodiruje Beetle ? onda' (if Beetle is touched then) block. Inside this 'if' block, there is a 'postavi bodovi na 1' (set score to 1) block. To the right of the 'if' block, there is a 'kada je kliknut' (when green flag clicked) block. This block is followed by 'postavi bodovi na 0' (set score to 0) and 'postavi vrijeme na 0' (set time to 0). Below these is a 'ponavljaj dok nije vrijeme = 0' (repeat while time is not 0) block. Inside this loop, there is a 'čekaj 1 sekundi' (wait 1 seconds) block and a 'promijeni vrijeme za 1' (change time by 1) block. Below the main script, there are four separate 'kada je tipka' (when key pressed) blocks. The first is 'kada je tipka strelica lijevo pritisnuta' (when left arrow key is pressed) followed by 'promijeni x za -10' (change x by -10). The second is 'kada je tipka strelica gore pritisnuta' (when up arrow key is pressed) followed by 'promijeni y za 10' (change y by 10). The third is 'kada je tipka strelica desno pritisnuta' (when right arrow key is pressed) followed by 'promijeni x za 10' (change x by 10). The fourth is 'kada je tipka strelica dolje pritisnuta' (when down arrow key is pressed) followed by 'promijeni y za -10' (change y by -10).



# SAŽETAK

- **Potprogram** je dio programskog koda koji je neovisan o glavnom programu. On se ne izvršava dok ga ne pozove glavni program.
- **definiraj** – blok naredba za definiranje potprograma.
- Dobijemo je klikom na  u kategoriji **Više Blokova**.
- Potprogramu nakon naredbe *definiraj* zadaješ naziv i naredbe koje izvršava nakon pozivanja.

# PONAVLJANJE

1. Kako pozivaš potprogram kada želiš da se izvede?
2. Zašto koristimo potprograme?
3. Kada se potprogram izvodi u glavnom programu (skripti)?