



IRIM - Institut za razvoj
i inovativnost mladih

micro:bit u nastavi

Dijeljenje cijelih brojeva

6. razred



MATEMATIKA

Dijeljenje cijelih brojeva

Izradite program na micro:bitu pomoću kojeg učenici mogu vježbati dijeljenje cijelih brojeva. Pritiskom na tipku A, ispisuje se djeljenik, npr. -54. Pritiskom na tipku B, ispisuje se djelitelj, npr. 9. Tada trebate razmisliti koliki je količnik tih brojeva tj. koliko je $-54 : 9$. Nakon što ste samostalno izračunali, pritiskom lijeve i desne tipke zajedno, rješenje se ispisuje na ekranu te možete provjeriti jeste li dobro izračunali. Ako rezultat na ekranu nije jednak onome do kojeg ste sami došli, ponovo možete pogledati koji su vam brojevi bili zadani, pritiskom na tipku A i na tipku B, odnosno koji je rezultat pritiskom na obje tipke zajedno. U ovom programu djelitelj je broj između -10 i 10, a isto tako i količnik. U slučaju dijeljenja s nulom, ispisuje se tužni smješko. Novi zadatak se generira nakon što protresetete micro:bit.

Ovu lekciju možete pronaći na edukacijskom portalu Izradi! na poveznici:

<https://bit.ly/dijeljenje-cijelih-brojeva-2>.

Korištenje s učenicima

Predmet	Matematika
Razred	6., osnovna škola
Polugodište	2.
Nastavna cjelina	Cijeli brojevi
Tema	Dijeljenje cijelih brojeva
Ključne riječi	cijeli brojevi, dijeljenje
Ideja i izrada materijala	Antonija Horvatek

Učenici pomoću ponuđene datoteke mogu vježbati na razne načine. Jedan je da vježbaju samostalno kod kuće. Na isti način mogu vježbati i na dopunskoj nastavi. Učenici koji imaju poteškoća s ovakvim računom, jednostavno i samostalno mogu vježbati pomoću micro:bita. Osim što vježbaju gradivo matematike, time ih ujedno navikavate na samoprocjenu svog znanja (kroz povratnu informaciju koju micro:bit daje o točnosti rezultata) i na uočavanje da i samostalnim radom (bez učitelja) mogu napredovati.

Na redovnoj nastavi također možete koristiti micro:bit s danom datotekom. Jedan je način da na kraju sata, na kojem ste zajedno vježbali ovakve zadatke, napravite igru/natjecanje između učenika. Nakon što ste s učenicima prošli ovo gradivo, na redovnoj nastavi, na početku bilo kojeg kasnijeg sata možete učenicima reći da, dok učitelj upisuje sat, na brzinu trebaju uključiti micro:bitove i ponovo provježbati/ponoviti si ovo gradivo. Micro:bit možete iskoristiti i za zadavanje domaće zadaće. Npr. učenicima možete reći da si pomoću micro:bita zadaju npr. 10 zadataka, zapišu ih u bilježnicu i riješe. Pritom učenik prvo treba s micro:bita prepisati prvi zadatak u bilježnicu, samostalno ga riješiti u bilježnici, te nakon toga pomoću micro:bita provjeriti rješenje. Na isti način nastaviti s do desetog zadatka. Također učenike treba upozoriti da kod zapisivanja zadataka, ako je drugi faktor negativan, moraju sami napisati zgradu, iako se ona na micro:bitu ne pojavljuje, npr. u zadatku $32 : (-8)$. Kroz takvu zadaću vježbaju i taj detalj.

Opis programa

Izradite program pomoću kojeg možete vježbati dijeljenje cijelih brojeva, npr. $-54 : 9$, $24 : (-8)$ i sl., pri čemu će djeljitelj i količnik biti brojevi od -10 do 10 . Usput naučite “trik” kako generirati dvije varijable takve da je prva djeljiva s drugom i uočite nijanse na što sve trebate paziti ako se generira zadatak u kojem dijelite s nulom (a znate da se ne može dijeliti s nulom).

Program treba raditi tako da:

Kad protresetete micro:bit, generira se novi zadatak tj. novi brojevi (ali se tad ne ispisuju).

Pritiskom tipke A ispisuje se djeljjenik.

Pritiskom tipke B ispisuje se djeljitelj.

Pritiskom obje tipke zajedno, ispisuje se količnik tih brojeva (rezultat kod dijeljenja).

Program se izrađuje u MakeCode editoru kojem možete pristupiti preko poveznice: <https://makecode.microbit.org/>.

Uvodna pojašnjenja ideje izrade programa:

Ovdje ne želite dijeljenje s ostatkom, pa djeljjenik ne može biti bilo koji broj (npr. ne želite zadatak poput $13:4$), već djeljjenik mora biti djeljiv s djeljiteljem. Da biste to postigli, u pozadini se neće slučajnim odabirom generirati djeljjenik i djeljitelj, već dva faktora, a djeljjenik ćete dobiti njihovim množenjem. Dakle, kad protresetete micro:bit, generirat će se dva broja između -10 i 10 (uzmite za primjer -4 i 10), pomnožit ćete ih ($-4 \cdot 10 = -40$), te će se pritiskom na tipka A ispisati njihov umnožak (u ovom slučaju broj -40), koji zapravo “glumi djeljjenika”, a pritiskom na tipkom B prvi faktor (u ovom slučaju -4), koji “glumi djeljitelja”. Pritiskom na tipka B prikazat će se drugi faktor (u ovom slučaju broj 10), koji predstavlja količnik brojeva -40 i -4 . Na taj način će na micro:bitu biti predstavljen zadatak $-40:(-4)=10$.

Ovdje treba uočiti jedan poseban slučaj, a to je kad je djeljitelj nula. Znamo da se ne može dijeliti s nulom! Budući da se djeljitelj generira kao prvi faktor (u programu to će biti varijabla a), treba provjeriti je li slučajno $a=0$. Ako jest, onda:

Djeljjenik nećete generirati na gore opisani način, umnoškom generiranih faktora, jer bi on u tom slučaju uvijek bio 0 , čime biste eliminirali zadatke poput $-4:0$, a to ne želite. Dakle, u slučaju $a=0$, djeljjenik ćete generirati slučajnim odabirom bilo kojeg broja od -10 do 10 .

Budući da u tom slučaju rezultat ne postoji, umjesto njega će se istovremenim pritiskom tipka A i B, prikazati plačka, koji nas podsjeća da ne možete dijeliti s nulom.

Izrada programa

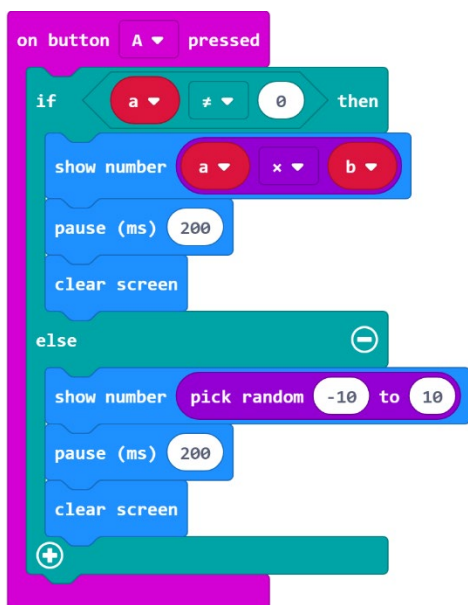
1. korak

Prvo kodirajte da se prilikom trešnje micro:bita u pozadini slučajnim odabirom generiraju dvije varijable **a** i **b** s vrijednostima između -10 i 10. Varijable kreirajte u kategoriji **Variables** pritiskom na opciju Make a variable. U prozor koji se pritom otvori upišite ime varijable.



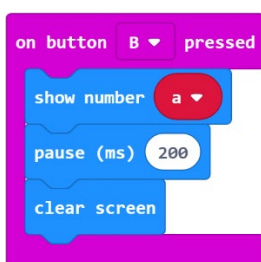
2. korak

Sad definirajte da se pritiskom tipke A prvo provjeri je li broj s kojim dijelite 0, tj. je li varijabla **a=0**. Ako nije, onda se, pritiskom na tipku A, treba ispisati umnožak brojeva **a** i **b**, a ako je **a=0**, onda se ispisuje bilo koji slučajno izabrani broj od -10 do 10 (u skladu s idejom opisanom pod “Uvodna pojašnjenja ideje izrade programa”). Pripadni kôd prikazan je na slici.



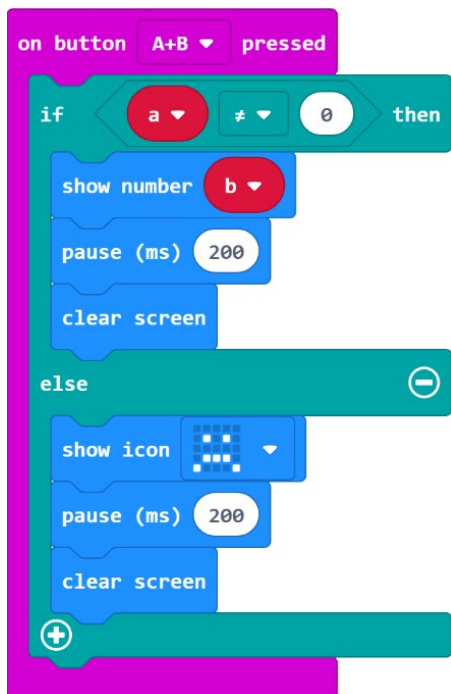
3. korak

Pritiskom na tipku B, treba se ispisati djelitelj, a to je upravo varijabla **a**. Kodirate kao što je prikazano na slici.



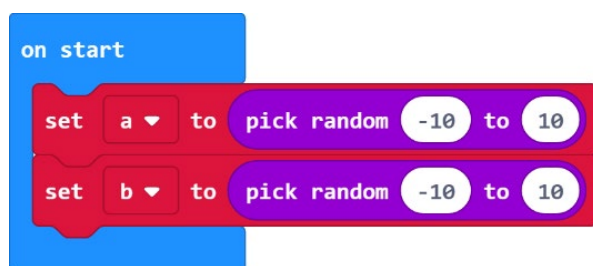
4. korak

Istovremenim pritiskom tipki A i B treba se pojaviti rezultat. U skladu s opisanim pod "Uvodna pojašnjenja ideje izrade programa", u slučaju kad varijabla **a** nije 0, kao rezultat dijeljenja se ispisiuje varijabla **b**, a u slučaju kad je **a=0**, budući da ne možete dijeliti s nulom, ispisiuje se plačko. Pripadni kôd prikazan je na slici.



5. korak

Na kraju još možete definirati da se prilikom samog uključenja micro:bita također generira novi zadatak, tj. da nije potrebno tresti micro:bit odmah nakon uključenja. Budući da se u tu svrhu trebaju pokrenuti iste naredbe koje se nalaze unutar **on shake** bloka, upravo te naredbe iz bloka **on shake** kopirajte i ugnijezdite ih unutar **on start** bloka, kao što je prikazano na slici.



Gotov program

```
on button A pressed
  if a ≠ 0 then
    show number a x b
    pause (ms) 200
    clear screen
  else
    show number pick random -10 to 10
    pause (ms) 200
    clear screen

on button B pressed
  show number a
  pause (ms) 200
  clear screen

on start
  set a to pick random -10 to 10
  set b to pick random -10 to 10

on button A+B pressed
  if a ≠ 0 then
    show number b
    pause (ms) 200
    clear screen
  else
    show icon [grid]
    pause (ms) 200
    clear screen

on shake
  set a to pick random -10 to 10
  set b to pick random -10 to 10
```

The image displays a Scratch script for a Micro:bit program. The script is organized into several event-driven blocks:

- on button A pressed:** An if-then-else block. If variable `a` is not equal to 0, it shows the product of `a` and `b`, pauses for 200ms, and clears the screen. Otherwise, it shows a random number between -10 and 10, pauses for 200ms, and clears the screen.
- on button B pressed:** Shows the value of variable `a`, pauses for 200ms, and clears the screen.
- on start:** Initializes variables `a` and `b` with random values between -10 and 10.
- on button A+B pressed:** An if-then-else block. If `a` is not 0, it shows the value of variable `b`, pauses for 200ms, and clears the screen. Otherwise, it shows a grid icon, pauses for 200ms, and clears the screen.
- on shake:** Resets variables `a` and `b` to random values between -10 and 10.