

# Možnosti uporabe programa Scratch pri matematiki

Milan Hlade  
Osnovna šola Koroška Bela, Jesenice

## Povzetek

Medpredmetno povezovanje oziroma povezovanje znanj je čedalje pomembnejše v izobraževanju. Če k temu dodamo še uporabo sodobne komunikacijske tehnologije in aktivno vlogo učencev, medpredmetnemu povezovanju damo dodatno težo. Povezali smo matematiko in računalniške izbirne predmete v osnovni šoli – sklop programiranja. V prispevku predstavimo možnosti in načine izdelave preprostih aplikacij za utrjevanje ali učenje osnovnih računskih operacij pri matematiki s pomočjo programa Scratch. Aplikacije lahko izdelujejo učitelji sami ali skupaj z učenci osmega in devetega razreda. Namenjene so utrjevanju seštevanja, odštevanja, množenja in deljenja v množici naravnih števil do 10, 20, 100 in celo v množici celih števil. Možnosti programa so velike, zato bi lahko obdelali še druga področja pri matematiki: funkcija, enačbe in neenačbe, zbiranje in predstavitev podatkov, geometrija. Pri izdelavi aplikacije prikažemo različne metode reševanja iste naloge, istega problema. Obenem prikažemo sistematičen, urejen pristop k sestavljanju aplikacije, kar je pogoj za reševanje kompleksnejših problemov. Tako izdelane aplikacije so zelo uporabne pri diferenciaciji pouka, pri pouku dodatne strokovne pomoči, pri dopolnilnem in dodatnem pouku matematike ter pri medsebojni pomoči med sošolci.

**Ključne besede:** medpredmetna povezava, aplikacija, matematika, računalništvo

## Use of Scratch Application in Mathematics

## Abstract

Interdisciplinary integration or knowledge collaboration is of increasing importance in education. By adding modern communication technology and an active role of pupils we can greatly enrich this area. We connected Mathematics and elective Computer Science subjects in primary school – the programming unit. This paper presents variations and methods of programming simple applications for training or learning basic arithmetic operations in Mathematics with the Scratch program. Applications can be made by teachers alone or together with pupils of the eighth or ninth grade. They are intended for pupils to train addition, subtraction, multiplication and division up to 10, 20, 100 and even in the set of whole numbers. The variations of this programme are vast, so it could be used for discussing other mathematical fields. By making the application we can show different methods for solving the same task or problem. At the same time, we show a systematic, organised approach to compiling this application, which is a basic condition for solving more complex problems. Such applications are helpful in differentiating instruction, providing additional professional support, teaching Mathematics to children with learning disabilities, teaching mathematically-talented pupils, etc.

**Keywords:** interdisciplinary integration, application, Mathematics, Computer Science

## Uvod

Matematika in računalniško programiranje sta v zadnjem času neločljivo povezana. Vendar danes v osnovni šoli redko najdemo primer medpredmetne povezave med omenjenima področjema. Razlog za to delno leži v majhnem deležu ur, namenjenih programiranju v učnih na-

črtih računalniških izbirnih predmetov. Vendar so znanja, ki jih učenci pridobijo z medpredmetnim učenjem globlja, uporabnejša (vir 7). Mnogi menijo, da je izdelava programov, aplikacij, s katerimi računamo, za osnovnošolce prevelik zalogaj. S prihodom uporabnikom prijaznejših okolij za programiranje (Light-Bot, Scratch, Alice, Etoys ...) je tudi to

področje postalo zanimivejše, prijaznejše, dostopnejše.

Pri računalniških izbirnih predmetih v osmem in devetem razredu smo za učenje programiranja uporabili program Scratch. Ker so se nekateri učenci s tem programom že srečali, so ga že precej dobro poznali. Na koncu sklopa programiranje naj bi izdelali neko svojo aplikacijo,

»igrico«. Uspešnejšim učencem se je to zdelo prelahko, nezanimivo. Skupaj smo se odločili za reševanje izbranega matematičnega problema, saj so bili učenci uspešni matematiki, program pa je poleg drugih ukazov vseboval sklopa ukazov operatorji in podatki, ki sta bila za učence še posebej zanimiva. Aplikacijo naj bi predstavili mlajšim bratom ali sestram in sošolcem, ki jim je matematika povzročala težave. Izkazalo se je, da so bile izdelane aplikacije uporabne tudi pri podaljšanem bivanju in za utrjevanje znanja računskih operacij.

## Programski jezik Scratch

Programski jezik Scratch je prosto dostopen program, namenjen zgodnjemu učenju programiranja. Lahko rečemo, da je osnovnošolcem pisan na kožo, saj lahko sprogramirajo že kar zahtevne igrice ali zahtevnejši program. Scratch je projekt skupine Lifelong Kindergarten Group pri MIT (Massachusetts Institute of Technology) Media Lab in je namenjen otrokom od 8. do 16. leta. Z njim z veseljem programirajo tudi starejši.

Na sliki 1 vidimo uporabniški vmesnik programa, ki je preveden v slovenščino. Vmesnik je razdeljen na več področij. Omenimo le področje na levi strani, kjer lahko vidimo, kaj se med programom dogaja z različnimi elementi, figurami. To področje se pri uporabi aplikacije v celoti pokaže na ekranu. Na sredini vmesnika je polje z gradniki, bloki programa, ki jih preprosto sestavljamo v področju, ki leži

na desni strani vmesnika. Funkcij vmesnika je še veliko, omenimo le spletno galerijo projektov uporabnikov, ki je zelo podobna galeriji v Geogebri.

Gradniki programskega jezika so smiselno razvrščeni v sklopih: premikanje, videz, zvok, svinčnik, podatki, dogodki, krmiljenje, zaznavanje, operatorji in ostali gradniki. Podrobneje si bomo ogledali le dva sklopa, najpomembnejša za matematiko - to sta podatki in operatorji. V sklopu podatki lahko ustvarimo in poimenujemo različne spremenljivke, ki skozi program spreminjajo svoje vrednosti. V naših primerih bodo to: seštevanec, odštevanec, faktor, delitelj, rezultat, število točk, napake ... V tem sklopu lahko ustvarimo tudi seznam spremenljivk. Učenci vidijo lep primer uporabe spremenljivk, saj se pri matematiki velikokrat postavi vprašanje: Kje bomo pa to potrebovali? Za matematiko še bolj zanimiv pa je sklop operatorji, kjer najdemo gradnike za seštevanje, odštevanje, množenje in deljenje dveh števil ali spremenljivk. Poleg teh gradnikov najdemo še gradnike za primerjanje dveh števil, logični IN, ALI in NE, gradnik, ki nam da ostanek pri deljenju, gradnik za zaokroževanje in gradnik za računanje kvadratnih korenov. Nekateri gradniki (kvadratni koren) so bili dodani šele z zadnjimi verzijami, zato lahko pričakujemo še nove gradnike, namenjene matematiki.

## Izdelava aplikacije

Pri izdelavi aplikacij smo se z učenci odločili uporabiti vse možnosti progra-

ma Scratch, ne le računanje, ampak tudi komunikacijo z uporabnikom, štetje pravih in napačnih izračunov in beleženje časa računanja. Uporabili nismo le možnosti zvoka pa še to le zaradi časovne stiske.

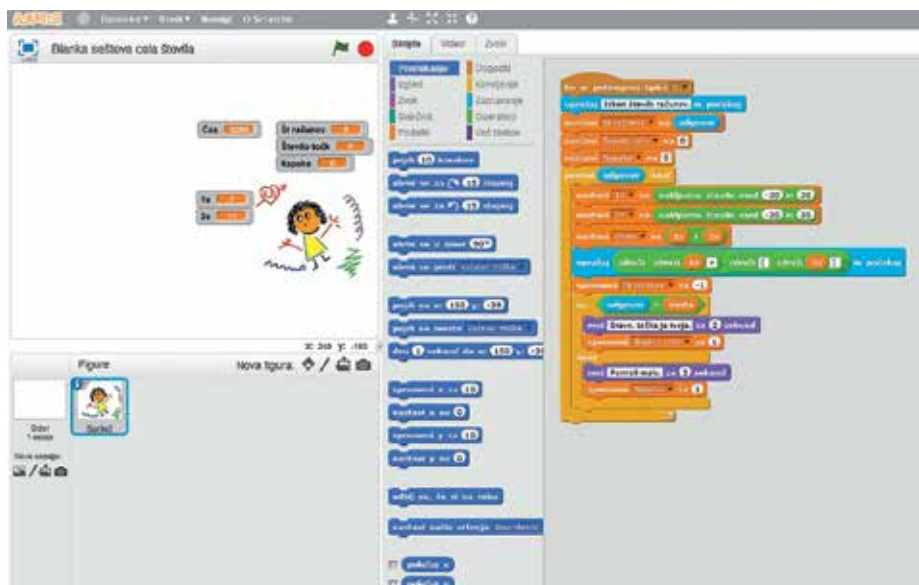
Na začetku smo predvideli računanje v množici naravnih števil do 10, 20, 100 in računanje s celimi števili, zato v nadaljevanju opišemo izdelavo aplikacije za seštevanje celih števil. Pred začetkom dela smo morali nalogo sistematično razčleniti na smiselno zaključene dele. Brez takega sistematičnega pristopa se lahko hitro prikradejo napake ali pa ne najdemo rešitve problema. Za komunikacijo z učencem uporabnikom aplikacije lahko uporabimo katerikoli lik, figurico, sliko ... Na OŠ Koroska Bela smo se odločili za deklco Blanko, ki je tudi maskota naše šole. Na sliki 2 vidimo zaslonko sliko z Blanko in spremenljivkami, ki jih vidi učenec na ekranu.



Slika 2: Zaslonko sliko

## Komunikacija z učencem in merjenje časa

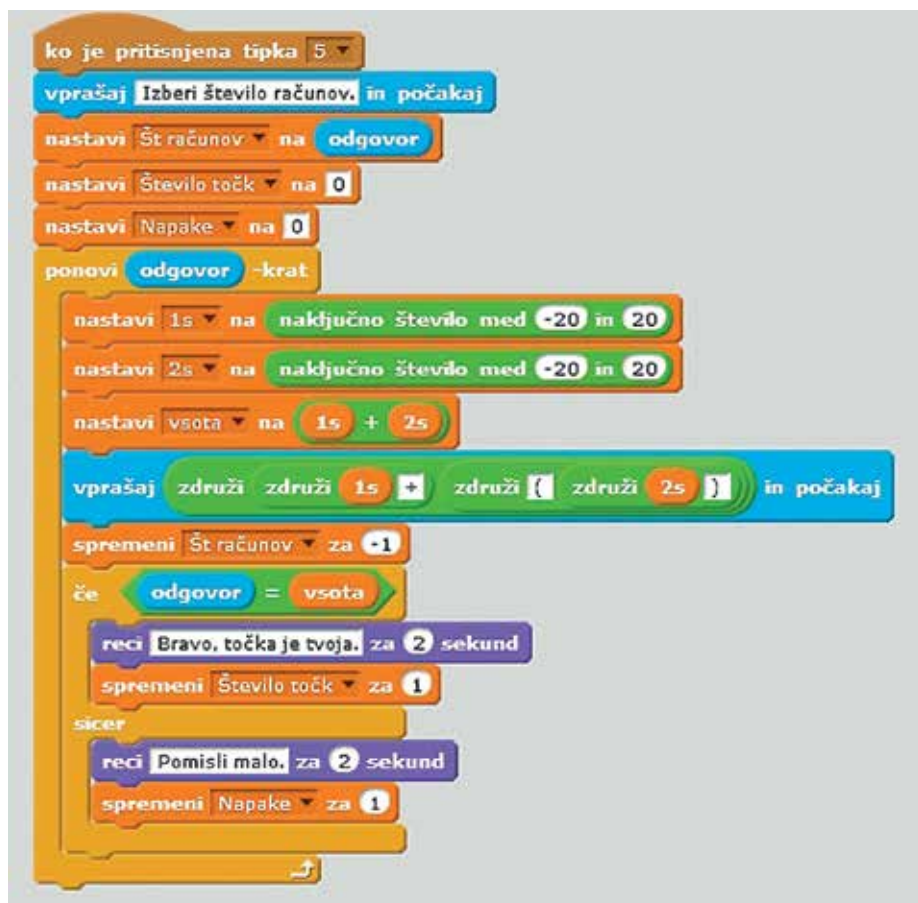
Na sliki 3 vidimo v zgornjem delu podprogram, kjer Blanka komunicira z učencem. Taka oblika komunikacije je pomembna za učence s težavami pri računanju. Opazil sem, da je ob taki komunikaciji trema in strah pred neuspehom pri učencih manjša. Strahu pred sodobno komunikacijsko tehnologijo običajno nimajo. Kasneje lahko s spodbudnimi namigi pomagamo učencem pri računanju; lahko bi tudi za določen čas prikazali številsko premico. Obenem učenci, ki izdelujejo aplikacijo, spoznajo podprogram, ki ga lahko uporabijo v več aplikacijah. Drugi podprogram je namenjen merjenju časa. Še pred tem moramo ustvariti spremen-



Slika 1: Uporabniški vmesnik s programom za seštevanje celih števil



Slika 3: Podprogram za komunikacijo z učencem in za merjenje časa.



Slika 4: Program za seštevanje celih števil

ljivko čas, ki se prikaže na zaslonu. Čas začnemo meriti na začetku računanja, vrednost spremenljivke čas pa se mora ob novem računanju nastaviti na nič. Vidimo, da merjenje časa izvedemo s preprosto zanko, kjer se vrednost spremenljivke poveča za 1 vsako sekundo. Seveda se mora merjenje časa ustaviti, ko je izračunan zadnji primer.

## Del programa za seštevanje celih števil

Glede na zahteve, ki smo si jih sami postavili, smo v sklopu podatki ustvarili potrebne spremenljivke in jih poimenovali (Slika 4):

- Število računov – izbere učenec na začetku računanja ter se zmanjša za 1 po izračunanem primeru
- Prvi seštevanec – 1 s – izbere računalnik naključno med  $-20$  in  $20$
- Drugi seštevanec – 2 s – izbere računalnik naključno med  $-20$  in  $20$
- Vsota – program sešteje oba seštevanca
- Število točk – poveča se za 1, če učenec pravilno izračuna račun
- Napake – poveča se za 1, če se učenec zmoti pri računanju

Prvi del aplikacije se začne s klikom na zastavico. Drugo del pa sprožimo s klikom na tipko 5 zaradi začetka merjenja časa računanja. Učenec najprej izbere število računov, ki jih želi izračunati. Vrednost se pripiše spremenljivki *Število računov*. Zaradi možnosti večkratne uporabe aplikacije nato nastavimo spremenljivki *Število točk* in *Napake* na nič. Nato izberemo zanko, ki se ponovi tolikokrat, kolikor računov želi izračunati učenec. V zanki najprej naključno izberemo vrednosti za spremenljivki 1 s in 2 s med  $-20$  in  $20$ . Obe meji lahko poljubno nastavljammo. Spremenljivki *Vsota* priredimo vsoto spremenljivk 1s in 2 s. V vprašanju združimo spremenljivko 1s, znak za seštevanje, oklepaj, spremenljivko 2 s in zaklepaj. Številski izraz je na zaslonu do vnosa rezultata. Pred nadaljevanjem moramo spremenljivko *Število računov* zmanjšati za 1. Sledi ugnjezdna zanka, kjer preverjamo spremenljivko *Vsota* z odgovorom. Če je odgovor pravilen, se spremenljivka *Število točk* poveča za 1, če pa je nepravilen, se tudi spremenljivka *Napake* poveča za 1. Vse skupaj opremimo s spodbudni-

mi komentarji. Ko se prva zanka ponovi zadnjič, se izvajanje aplikacije ustavi.

## Preverjanje aplikacije

Preverjanje delovanja aplikacije poteka že sprti, vendar pravo preverjanje sledi

na koncu. Po uspešnem delovanju se pri učencih običajno pojavi veliko novih idej. Naj naštejemo le nekatere, ki so se nam pojavile po izdelavi aplikacije:

- učenec lahko sam izbira velikost seštevancev,
- več računskih operacij,

- grafična predstavitev na številski premici,
- pomoč v obliki podobnega primera,
- računanje vrednosti zahtevnejših številskih izrazov ...

## Zaključek

Pri izdelavi opisanih aplikacij so učenci pokazali veliko ustvarjalnosti in matematičnega znanja. Rešitve posameznih korakov jim niso bile takoj na voljo, ampak so jih v večini primerov poiskali sami. Posamezno aktivnost npr. seštevanje naravnih števil do 20 sta ločeno »reševala« dva učenca, ki pa sta se lahko med seboj posvetovala. Včasih sta posamezna učenca delala skoraj enako, včasih pa sta do istega cilja prišla na različna načina. Zanimiva sta primera izdelave aplikacije deljenja naravnih števil v obsegu do 100. En učenec je izbral deljenca med 1 in 100 in delitelja med 1 in 10. Če se deljenje ni izšlo, je izbiro ponovil. Drugi učenec je najprej opravil množenje med dvema izbranimi številoma med 1 in 10 in nato opravil deljenje. Pri analizi aplikacij smo pregledali slabe in dobre programske in matematične rešitve aplikacije.

Menimo, da se lahko s programom Scratch izdela še veliko preprostih aplikacij, kot so: grafična predstavitev seštevanja celih števil, poštevanka, pisno računanje, računanje z racionalnimi števili, risanje geometrijskih likov, risanje pravilnih mnogokotnikov. Menim, da bi tak način programiranja lahko vključili v kakšno od računalniških (tekmovanje Bober) ali celo matematičnih tekmovanj. ■

## Viri

Bonaca, K. (2016). Bober – naloge iz matematike pri računalništvu. *Matematika v šoli*, XXII, št. 3-4, str. 63–72.

Hlade, M. (2016). *Izdelava aplikacij za računanje s programom Scratch*. Konferenca Sirikt 2016.

Lajovic, S. (2011). *Scratch*. Ljubljana: Založba Pasadena.

Repolusk, S. (2012). Priprava bodočih učiteljev matematike na povezovanje znanj pri pouku matematike, *Matematika v šoli*, XVIII, št. 1-2, str. 97–102.

<http://www.monitor.si/clanek/prvi-koraki-v-programiranje2/124892/?xURL=301> (oktober, 2017)

<https://scratch.mit.edu/> (oktober, 2017)

<http://pefprints.pef.uni-lj.si/2005/1/Matematika.pdf> (oktober, 2017)