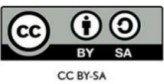


PREDLOG UČNEGA SCENARIJA

01 PREDSTAVITEV

Naslov učnega scenarija Oblikujte kratek, privlačen naslov učnega scenarija.	Matematika in programiranje z roko v roki
Tema (glede na področje RIN) Označite ustrezno.	Algoritmi in programiranje
Povzetek učnega scenarija Kratka predstavitev učne aktivnosti	Uporaba pogojnega stavka. Računanje ničle linearne funkcije. Preverjanje ali dana točka leži na grafu funkcije. Kolinearnost danih točk. Računanje presečišča dveh linearnih funkcij – reševanje sistema dveh enačb z dvema neznankama.
Ključne besede	Pogojni stavek, linearna funkcija
Licenca dostopnosti in uporabe učnega scenarija	 <small>CC BY-SA</small>
Avtorji učnega scenarija na VIZ V abecednem vrstnem redu.	Avtorja učnega scenarija Mateja Valjavec in Mateja Žepič.

02 KONTEKST IZVEDBE IN PRIPRAVA

Starost dijakov	16 let
Predznanje Potrebno/pričakovano predznanje.	Matematično predznanje – linearna funkcija, ki jo obravnavajo v prvem letniku, zadnja četrtina šolskega leta.
Trajanje izvedbe Trajanje izvedbe aktivnosti (pedagoške ure).	3 ure
Viri za oblikovanje priprave Npr. spletne strani, e-knjige in članki, zvočni posnetki, videoposnetki, interaktivni spletni viri, fizični viri (npr. monografije, učbeniki). Bodite pozorni na kvaliteto podanih virov.	Učbenik RIN: https://lusy.fri.uni-lj.si/ucbenik/ Programiranje z delčki Bon Klanjšček, Dvoržak, Felda: Matematika 1, učbenik za gimnazije, DZS, Ljubljana 2009

03 NAMEN IN UČNI CILJI (OPERATIVNI)

Namen Splošni cilji učnega scenarija.	Dijaki bodo utrdili znanja uporabe pogojnega stavka. Ponovijo znanje matematičnih vsebin: definicija linearne funkcije, pomen smernega koeficienta, ničla funkcije, kolinearnost, reševanje sistema dveh linearnih enačb za dvema neznankama.
---	---

Učni cilji (operativni):	Dijaki bodo <ul style="list-style-type: none"> – znali uporabiti pogojni stavek – sposobni predvideti rezultate in preveriti svoja predvidevanja s praktičnim izvajanjem programa. – znali izračunati ali dana točka leži na grafu funkcije, v kateri točki se sekata grafa dveh funkcij, ali so dane točke kolinearne.
---------------------------------	--

04 AKTIVNOST

Didaktični pristop	Projektno učenje, sodelovalno učenje, programiranje v parih
Material	Računalnik ali tablica, spletna stran Programiranje z delčki Matematika 1, učbenik za gimnazije
Koraki izvedbe aktivnosti	
<p>Uvodna frontalna razlaga pogojnega stavka: Ponovimo/utrdimo znanje o pogojnih stavkih, ki smo jih spoznali v eni od prejšnjih ur.</p> <p>Motivacija: Kje v vsakdanjem življenju uporabljamo linearno funkcijo? Dijaki navedejo nekaj primerov.</p> <p>Pomoč oz. debata primerov:</p> <p>a) Mama je vključila pečico, ki se vsako minuto segreje za nekaj stopinj, segrevanje je enakomerno. b) Sosedje večkrat med letom napolnijo bazen do določene višine in se gladina vode enakomerno zvišuje vsako minuto.</p> <p>Uvodna frontalna razlaga matematičnih vsebin:</p> <ol style="list-style-type: none"> 1. Definicija linearne funkcije. Kdaj točka leži na grafu dane funkcije. 2. Kako računamo ničlo funkcije in začetno vrednost. 3. Kdaj so tri točke kolinearne. 4. Kako izračunamo presečišče grafov dveh linearnih funkcij. <p>Primer 1: Ali dana točka $A(x, y)$ leži na grafu linearne funkcije $f(x) = kx + n$?</p> <p>Če točka leži na grafu dane funkcije, zadošča njeni enačbi. Torej mora biti funkcijska vrednost enaka drugi koordinati dane točke. Dijak si bo izbral poljubno funkcijo in točko ter zapisal algoritem, ki mu poda rešitev v obliki, npr.:</p> <p>Točka (3.0,3.0) ne leži na grafu funkcije $f(x) = 3.0x + 3.0$ Točka (1.0,1.0) ne leži na grafu funkcije $f(x) = 1.0x + 1.0$</p> <p>Programiranje z delčki:</p>	

The Scratch code consists of the following blocks:

- Five 'ask for number' blocks for variables `k`, `n`, `x`, `y`, and `f` with prompts: "Zapiši smerni koeficient", "Zapiši začetno vrednost", "Koliko je prva koordinata točke", "Koliko je druga koordinata točke", and a blank prompt.
- A calculation block: `f = k * x + n`.
- An 'if' block: `if f == y`.
- Inside the 'if' block: 'set answer to' "da".
- Outside the 'if' block: 'set answer to' "ne".
- 'set output to' "Točka (".
- 'add text' blocks: `x`, `)`, `odg`, "leži na grafu funkcije f(x) = ", `k`, `x +`, and `n`.
- 'create speech bubble' blocks for `y` and `)`.
- 'print' block: `izpisi`.

Koda v Pythonu:

```

k = float(input('Zapiši smerni koeficient'))
n = float(input('Zapiši začetno vrednost'))
x = float(input('Koliko je prva koordinata točke'))
y = float(input('Koliko je druga koordinata točke'))
f = k * x + n
if f == y:
    odg = "
else:
    odg = 'ne '
izpis = 'Točka ('
izpis = str(izpis) + str(x)
izpis = str(izpis) + str(',') + str(y) + ')' + str(odg) + 'leži na grafu funkcije f(x) ='
izpis = str(izpis) + str(str(k) + 'x +' + str(n))
print(izpis)

```

Nato z dijaki pogledamo skupaj, kaj jim je uspelo in na kakšen način. Za tiste dijake, ki težje sami zapišejo celoten postopek, le to naredimo skupaj.

Primer 2: Izračunaj ničlo linearne funkcije $f(x) = kx + n$.

Dijaki na list papirja naprej matematično zapišejo izračun ničle dane funkcije. Torej funkcijo izenačijo z 0 in izrazijo spremenljivko x . Nato skupaj premislimo korake, ki so potrebni za zapis programa. Sledi samostojno delo v parih.

Po krajšem samostojnem delu skupaj zapišemo program

```

nastavi k na vprašaj za število s sporočilom " Zapiši smerni koeficient "
nastavi n na vprašaj za število s sporočilom " Zapiši začetno vrednost "
če
  n = 0
izvedi
  nastavi x na 0
sicer
  nastavi x na k ÷ n × -1
izpiši x

```

```

k = float(input('Zapiši smerni koeficient '))
n = float(input('Zapiši začetno vrednost '))
if n == 0:
    x = 0
else:
    x = (k / n) * -1
print(x)

```

Spomnimo se, da je premica lahko tudi vzporedna abscisni osi, torej ničla ne obstaja. Želim, da program izpiše v takem primeru: Graf linearne funkcije je vodoravna premica, ki ne seka abscisne osi. Dijaki program dopolnijo.

```

nastavi k na vprašaj za število s sporočilom " Zapiši smerni koeficient linearne funkcije "
nastavi n na vprašaj za število s sporočilom " Zapiši začetno vrednost linearne funkcije "
če
  k = 0
izvedi
  izpiši ustvari besedilo iz " Graf linearne funkcije f(x) = "
  " je vodoravna premica, ki ne seka abscisne osi. "
sicer
  če
    n = 0
  izvedi
    nastavi x na 0
    izpiši ustvari besedilo iz " Graf linearne funkcije f(x) = "
    " x ima ničlo pri x = 0. "
  sicer
    nastavi x na k ÷ n × -1
    izpiši ustvari besedilo iz " Graf linearne funkcije f(x) = "
    " x + ( "
    " ) ima ničlo pri x = "
    " x "

```

Izpis:

```
Graf linearne funkcije f(x) = 4.0x + (5.0) ima ničlo pri x = -0.8
Graf linearne funkcije f(x) = 7.0 je vodoravna premica, ki ne seka abscisne osi.
Graf linearne funkcije f(x) = 4.0x ima ničlo pri x = 0.
```

```
k = float(input('Zapiši smerni koeficient linearne funkcije'))
n = float(input('Zapiši začetno vrednost linearne funkcije'))
if k == 0:
    print('Graf linearne funkcije f(x) = ' + str(n) + ' je vodoravna premica, ki ne seka abscisne osi.')
else:
    if n == 0:
        x = 0
        print('Graf linearne funkcije f(x) = ' + str(k) + 'x ima ničlo pri x = 0.')
    else:
        x = (k / n) * -1
        print('Graf linearne funkcije f(x) = ' + str(k) + 'x + (' + str(n) + ') ima ničlo pri x = ' + str(x))
```

Primer 3: Ali so dane tri točke kolinearne?

Ponovimo kaj vemo o kolinearnosti točk. Možnih načinov reševanja danega problema je več. Mi bomo izbrali računanje ploščine trikotnika, ki ga dane tri točke določajo. Če bi imeli več točk, pa je smiselno zapisati enačbo premice, ki poteka skozi dve točki in nato preverimo, tako kot smo zapisali program v prvi nalogi, ali preostale točke ležijo na premici.

Ponovimo teorijo:

Za dane točke $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$ izračunamo ploščino trikotnika po formuli:

$S = \frac{1}{2} \cdot o \cdot ((x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1))$, kjer je o orientacija trikotnika. Če si oglišča trikotnika sledijo v nasprotni smeri urinega kazalca, je orientacija pozitivna (+1), v nasprotnem primeru je negativna (-1). Iz formule vidimo, da je pomemben le člen v oklepaju $(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)$. Če je enak 0, potem je ploščina enaka 0, kar pomeni da dane točke ležijo na isti premici in so kolinearne.

Dijaki v parih sodelujejo in zapišejo program.

```

nastavi x1 na vprašaj za število s sporočilom "Zapiši prvo koordinato točke A"
nastavi y1 na vprašaj za število s sporočilom "Zapiši drugo koordinato točke A"
nastavi x2 na vprašaj za število s sporočilom "Zapiši prvo koordinato točke B"
nastavi y2 na vprašaj za število s sporočilom "Zapiši drugo koordinato točke B"
nastavi x3 na vprašaj za število s sporočilom "Zapiši prvo koordinato točke C"
nastavi y3 na vprašaj za število s sporočilom "Zapiši drugo koordinato točke C"
nastavi rezultat na (x2 - x1) * (y3 - y1) - (x3 - x1) * (y2 - y1)
če rezultat = 0
    izvedi izpiši "Točke A, B in C so kolinearne"
sicer izvedi izpiši "Točke A, B in C niso kolinearne"
  
```

Program v Pythonu:



```

x1 = float(input('Zapiši prvo koordinato točke A'))
y1 = float(input('Zapiši drugo koordinato točke A'))
x2 = float(input('Zapiši prvo koordinato točke B'))
y2 = float(input('Zapiši drugo koordinato točke B'))
x3 = float(input('Zapiši prvo koordinato točke C'))
y3 = float(input('Zapiši drugo koordinato točke C'))
rezultat = (x2 - x1) * (y3 - y1) - (x3 - x1) * (y2 - y1)
if rezultat == 0:
    print('Točke A, B in C so kolinearne')
else:
    print('Točke A, B in C niso kolinearne')

```

Primer 4: Izračunaj presečišče grafov dveh linearnih funkcij.

Ponovimo teorijo: Na list papirja najprej zapišemo matematični postopek za izračun presečišča dveh premic. Ponovimo metode reševanja in nato se dijaki sami odločijo s katero metodo bodo zapisali program, mi bomo uporabili primerjalni način način. [10 minut]

Dani sta funkciji $f_1(x) = k_1x + n_1$ in $f_2(x) = k_2x + n_2$. Njuna grafa sta premici podani z enačbama $y_1 = k_1x + n_1$ in $y_2 = k_2x + n_2$.

Reševane tega sistema na primerjalni način:

$$\begin{aligned}
 y_1 &= y_2 \\
 k_1x + n_1 &= k_2x + n_2 \\
 x &= \frac{n_2 - n_1}{k_1 - k_2}
 \end{aligned}$$

Iz enačbe izrazimo neznanke x :

Ko imamo spremenljivo x je vstavimo v eno izmed danih enačb premic in izračunamo še y : $y = k_1 \cdot \frac{n_2 - n_1}{k_1 - k_2} + n_1$.

Rešitev je točka $T(x, y)$. Spomnimo dijake še na množico rešitev sistema dveh enačb, ki ni samo točka. Lahko je prazna množica ali cela premica. Tekom zapisa programa morajo upoštevati vse tri možne rešitve.

Dijaki se lotijo samostojnega dela.

The Scratch code performs the following steps:

- Asks for the slope (k_1) and y-intercept (n_1) of the first function.
- Asks for the slope (k_2) and y-intercept (n_2) of the second function.
- Checks if $k_1 = k_2$ and $n_1 = n_2$. If true, it outputs: "Grafa linearnih funkcij se prekrivata, torej je ..."
- Checks if $k_1 = k_2$ but $n_1 \neq n_2$. If true, it outputs: "Grafa linearnih funkcij sta vzporedni premici, t..."
- Otherwise, it calculates the intersection point $T(x, y)$ using the formulas:

$$x = \frac{n_2 - n_1}{k_1 - k_2}$$

$$y = k_1 \cdot x + n_1$$
 and outputs: "Presečišče je točka T()"

Program zapisan v Pythonu:

```

k1 = float(input('Zapiši smerni koeficient prve funkcije'))
n1 = float(input('Zapiši začetno vrednost prve funkcije'))
k2 = float(input('Zapiši smerni koeficient druge funkcije'))
n2 = float(input('Zapiši začetno vrednost druge funkcije'))
if k1 == k2 and n1 == n2:
    odg = 'Grafa linearnih funkcij se prekrivata, torej je rešitev premica z enačbo ' + 'y = ' + str(k1) + 'x + (' + str(n1) + ')'
    print(odg)
elif k1 == k2:
    odg = 'Grafa linearnih funkcij sta vzporedni premici, torej nimata skupne točke.'
    print(odg)
else:
    x = (n2 - n1) / (k1 - k2)
    y = k1 * x + n1
    izpis = 'Presečišče je točka T(' + str(x) + ', ' + str(y) + ')'
    print(izpis)

```

Ko dijaki zaključijo s samostojnim delom, vse rešitve komentiramo in skupaj razložimo še tistim dijakom, ki sami niso prišli do končnega rezultata. Ob koncu ure ponovimo, kaj smo se naučili o pogojnem stavku.

Prilagoditev aktivnosti	Dejavnost prilagodimo glede na odziv dijakov, skupini, ki hitro opravi nalogo postavimo vprašanja za razmislek o robnih primerih oziroma o zapisu programa, če bi rešili nalogo po kakšnem drugem načinu, kot smo ga mi predpostavili.
-------------------------	--

05 VREDNOTENJE, EVALVACIJA, REFLEKSIJA

<p>Spremljanje in evalvacija</p> <p>Opišite, kako bo potekalo formativno spremljanje napredka dijakov in evalvacija izvedene aktivnosti. Vir: https://www.zrss.si/izdelek/formativno-spremljanje-v-podporo-ucenju/</p> <p>Pripravite vprašanja, zapišite oblikovane namene učenja in kriterije uspešnosti za vrednotenje znanja in spremljanje napredka dijakov za načrtovane aktivnosti</p>	
<p>Refleksija</p> <p><i>Refleksija izvedbe dejavnosti skupaj z otroki/ učenci (izhodišča pogovora z otroki)</i></p> <p>Kaj je bilo dijakom všeč in zakaj – zapis izjav dijakov.</p> <p>Kaj so dijaki po njihovem mnenju spoznali, ugotovili?</p> <p>S kom in kako so sodelovali?</p> <p>Kako so se počutili, kaj do doživljali?</p> <p><i>Profesionalna refleksija načrtovanje in izvedbe:</i></p> <p>Na podlagi izvedbe ocenite ustreznost načrtovanja in izvedbe procesa učenja in poučevanja za dijake v razredu (kaj ocenjujete kot uspešno načrtovanje in kaj bi spremenili/ dopolnili)?</p> <p>Opišite situacije, na podlagi katerih lahko sklepate, da so se v procesu učenja začeli realizirati zastavljeni cilji.</p> <p>Pri katerih elementih dejavnosti so bili dijaki uspešni/ delno ali neuspešni?</p> <p>Katere cilje so dijaki dosegli in katerih mogoče ne, zakaj?</p> <p>Zapišite pobude in predloge ter komentarje dijakov, ki so jih dali oziroma izrazili v procesu izvedbe in evalvacije.</p> <p>Zakaj je izvedba dejavnosti lahko primer dobre prakse? Kje ste imeli največ težav in katere predloge izboljšav predlagate?</p>	
Drugi komentarji	

06 KURIKULUM

Medpredmetno povezovanje	Linearna funkcija, smerni koeficient, začetna vrednost, izračun ničle da ne funkcije, kolinearnost, presečišče.
Povezanost s skupnimi cilji učnih načrtov	Digitalne kompetence



