



Mladi raziskovalci Slovenije 2025

59. državno srečanje

Raziskovalna naloga

PROGRAMIRANJE V SCRATCHU S POMOČJO UI

Raziskovalno področje: Računalništvo ali telekomunikacije

Šola: **OŠ Angela Besednjaka Maribor**

Avtor: **Jurij Černetič**

Mentor: **Dejan Žmavc, prof.**

Maribor, april 2025

KAZALO VSEBINE

1	UVOD	1
1.1	IZHODIŠČE IN NAMEN DELA	1
1.2	CILIJ NALOGE	1
1.3	HIPOTEZE	1
1.4	METODE DELA	1
2	TEORETIČNI DEL	2
2.1	KAJ JE SCRATCH?	2
2.1.1	<i>Pomen</i>	3
2.1.2	<i>Uporabnost</i>	4
2.1.3	<i>Grafični vmesnik</i>	4
2.1.4	<i>Sprite ali figura</i>	5
2.2	TEKMOVANJE V SCRATCHU V SLOVENIJI	6
2.2.1	<i>Uspešnost naše šole na tekmovanju</i>	7
2.3	UMETNA INTELIGENCA.....	8
2.3.1	<i>ChatGPT</i>	9
2.3.2	<i>Gemini</i>	10
2.3.3	<i>Copilot</i>	12
3	RAZISKOVALNI DEL	14
3.1	PRIMER REŠITVE Z UPORABO CHATGPT	15
3.1.1	<i>Odgovor (predlog rešitve) aplikacije UI – ChatGPT</i>	15
3.1.2	<i>Ali mi lahko narediš barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3? ...</i>	17
3.1.3	<i>Moje ugotovitve</i>	17
3.2	PRIMER REŠITVE Z UPORABO GEMINI	19
3.2.1	<i>Odgovor (predlog rešitve) aplikacije UI – Gemini</i>	19
3.2.2	<i>Ali mi lahko narediš barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3? ...</i>	20
3.2.3	<i>Moje ugotovitve</i>	21
3.3	PRIMER REŠITVE Z UPORABO MICROSOFT 365 COPILOT	23
3.3.1	<i>Odgovor (predlog rešitve) aplikacije UI – Copilot</i>	23
3.3.2	<i>Ali mi lahko narediš barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3? ...</i>	24
3.3.3	<i>Moje ugotovitve</i>	24
3.4	REŠITEV ZOTKS.....	26
4	REZULTATI IN UGOTOVITVE	27
4.1	MNENJA UČITELJEV	28
4.2	HIPOTEZA 1.....	29
4.3	HIPOTEZA 2.....	29
4.4	HIPOTEZA 3.....	30
4.5	HIPOTEZA 4.....	30
4.6	HIPOTEZA 5.....	30
5	DRUŽBENA ODGOVORNOST, TRAJNOST, NAPREDEK	30

5.1	KORISTI UMETNE INTELIGENCE	30
5.2	NEVARNOSTI IN POMISLEKI	31
5.3	KRITIČNA PRESOJA IN ODGOVORNA UPORABA	31
6	ZAKLJUČEK	31
7	VIRI IN LITERATURA	33
7.1	LITERATURA	33
7.2	SPLETNI VIRI	33
7.3	VIRI SLIK	34

KAZALO SLIK

Slika 1:	Logotip Scratcha in glavna figura (vir: splet)	3
Slika 2:	Grafični vmesnik Scratcha (lasten vir)	4
Slika 3:	Knjižnica vgrajenih figur (lasten vir)	6
Slika 4:	Potek dela na raziskovalnem taboru (lasten vir)	14
Slika 5:	Naloga, ki sem jo izbral za svoje raziskovanje (vir: splet, ZOTKS)	15
Slika 6:	Fotografija kode (blokovi), ki jo je predlagal ChatGPT (lasten vir)	18
Slika 7:	Fotografija kode (blokovi), ki jo je bilo potrebno dodati (lasten vir)	18
Slika 8:	Fotografija kode (blokovi), ki jo je predlagal Gemini (lasten vir)	22
Slika 9:	Fotografija kode (blokovi), ki jo je bilo potrebno dodati (lasten vir)	22
Slika 10:	Fotografija kode (blokovi), ki jo je predlagal Copilot (lasten vir)	25
Slika 11:	Fotografija kode (blokovi), ki jo je bilo potrebno dodati (lasten vir)	25
Slika 12:	Koda (bloki) rešitve naloge (vir: ZOTKS)	26
Slika 13:	Merjenje časa generiranja odgovora UI s štoparico (lasten vir)	27

KAZALO TABEL

Tabela 1:	Število udeležencev na šolskem in državnem tekmovanju (Scratch) ter osvojena priznanja v zadnjih petih letih (vir: šolska spletna stran)	8
Tabela 2:	Podatki merjenja hitrosti generiranja odgovora na vprašanje (poziv UI)	28

KAZALO GRAFOV

Graf 1:	Osvojena priznanja učencev na šolskem in državnem tekmovanju (Scratch) v zadnjih petih letih (vir: šolska spletna stran)	8
Graf 2:	Kako hitro je aplikacija UI ponudila rešitev naloge (čas v sekundah)	28

POVZETEK

V raziskovalni nalogi sem raziskoval, kako se lahko programiranje v Scratchu – vizualnem programskem jeziku, ki je namenjen začetnikom, še izboljša s pomočjo umetne inteligence. Scratch je odlično orodje za otroke in mladostnike, saj s preprostim sestavljanjem blokov omogoča učenje logike in reševanja problemov. Poleg raziskovanja osnov programiranja skozi Scratch sem se osredotočil tudi na vlogo orodij umetne inteligence (UI), kot so ChatGPT, Gemini in Microsoft 365 Copilot, ki nam lahko pomagajo pri ustvarjanju projektov, reševanju programerskih izzivov ter iskanju napak. Ta orodja delujejo kot mentorji, ki te lahko vodijo skozi proces programiranja. Primerjal sem njihove odgovore oz. predloge za reševanje nalog in jih testiral, tako da sem sestavil projekt v Scratchu.

Prav tako sem se posvetil povezavi med programiranjem v Scratchu in računalniškimi tekmovanji, kjer lahko učenci svoje znanje preizkusimo v praksi. Tekmovanja, kot so Scratch izzivi ali naloge na tekmovanjih iz programiranja za osnovnošolce, nam ponujajo priložnost, da svoje spretnosti izpopolnimo in se pomerimo z drugimi v reševanju kreativnih problemov. S pomočjo umetne inteligence lahko še bolje razumem programerske koncepte in se pripravljam na tekmovanja ter s tem izboljšujem svoje rezultate.

Ključne besede: Scratch, umetna inteligenca, programiranje, reševanje problemov, tekmovanja

SUMMARY

In my research project, I explored how programming in Scratch – a visual programming language designed for beginners – can be further enhanced with the help of artificial intelligence. Scratch is an excellent tool for children and young people, as it allows them to learn logic and problem-solving through simple block-based coding. In addition to exploring the basics of programming with Scratch, I also focused on the role of AI tools, such as ChatGPT, Gemini, and Microsoft 365 Copilot, which can assist us in creating projects, solving programming challenges, and debugging. These tools act as mentors, guiding users through the programming process. I compared their responses and suggestions for solving tasks and tested them by creating a project in Scratch.

I also examined the connection between programming in Scratch and computer science competitions, where students can put their knowledge into practice. Competitions, such as Scratch challenges or tasks in programming contests for primary school students, provide opportunities to improve our skills and compete with others in solving creative problems. With the help of artificial intelligence, I can better understand programming concepts, prepare for competitions, and improve my performance.

Keywords: Scratch, artificial intelligence, programming, problem-solving, competitions

ZAHVALA

Zahvaljujem se mentorju, ki me je tako pri tekmovanjih kot tudi pri raziskovalni nalogi zelo podpiral in mi pomagal.

1 UVOD

1.1 Izhodišče in namen dela

Namen moje raziskovalne naloge je, da najprej predstavim nastanek, razvoj in pomen programskega orodja Scratch. Na kratko sem orisal tudi potek računalniških tekmovanj iz programiranja v Scratchu. Sledi poglavje o umetni inteligenci (kratica: UI) in nato raziskovalni del, kjer sem raziskoval ali nam različne aplikacije UI lahko pomagajo pri programiranju v Scratchu.

V sklepnem delu sem analiziral rezultate in zapisal moje ugotovitve. Posvetil sem se tudi družbeni odgovornosti, trajnosti ter napredku s pomočjo UI in v zaključku povzel opravljeno delo.

1.2 Cilij naloge

Glavni cilji moje raziskovalne naloge:

- predstavil bom razvoj in pomen programskega orodja Scratch,
- petletna analiza sodelovanja in rezultatov tekmovanja na naši OŠ,
- testiranje predlaganih rešitev s strani UI,
- izdelava predlogov izboljšav po testiranju,
- primerjava odgovorov UI z rešitvijo ZOTKS,
- ovrednotenje mojih predlogov izboljšav z mnenjem učiteljev računalništva.

1.3 Hipoteze

Pri svojem delu sem si postavil naslednje hipoteze:

Hipoteza 1: ChatGPT mi ne bo podal prave rešitve naloge.

Hipoteza 2: Gemini mi bo podal popolnoma ustrezno rešitev naloge.

Hipoteza 3: Microsoft 365 Copilot bo znal ustvariti .sb3 datoteko.

Hipoteza 4: Orodja UI nam lahko pomagajo pri programiranju v Scratchu.

Hipoteza 5: Učitelji računalništva se bodo strinjali z mojimi predlogi izboljšav.

1.4 Metode dela

Na začetku sem predstavil programsko orodje Scratch, njegov pomen ter uporabnost.

Takoj za tem sem izbral tri aplikacije umetne inteligence, ki se v zadnjem času najbolj uporabljajo. Nato sem poiskal primerno nalogo iz starih šolskih tekmovanj.

Vsem trem aplikacijam sem podal enako nalogo oz. poziv, torej naj mi pomaga rešiti nalogo v Scratchu. Opazoval sem proces izdelave rešitev ter koliko časa so aplikacije za to potrebovale. Za predlagane rešitve je bilo potem potrebno ustvariti projekt v Scratchu, ga testirati in po potrebi dopolniti. Na koncu sem primerjal in analiziral vse odgovore UI ter zapisal moje rezultate in ugotovitve.

V sklepnem delu naloge pa sem se posvetil še ozaveščanju o uporabi UI, njeni uporabnosti, prednostih ter da je potrebna velika mera kritične presoje in preverjanje resničnosti podatkov.

Pri pripravi raziskovalne naloge so bila uporabljena tri različna orodja: ChatGPT, Gemini in Microsoft 365 Copilot. Podatki, besedila (odgovori) in predlagane rešitve so bili pregledani, prilagojeni ter verificirani s strani avtorja in mentorja.

2 TEORETIČNI DEL

2.1 Kaj je Scratch?

Scratch je najpopularnejši začetniški vizualni programski jezik oz. orodje ter odličen način, da se otroci in mladi na zabaven ter interaktivni način naučijo programiranja. Namesto pisanja zapletenih kodnih vrstic, Scratch uporablja grafične bloke, ki jih lahko med seboj povezuješ in tako ustvarjaš svoje interaktivne zgodbe, igre, animacije in še veliko več. Bil je razvit v Inštitutu Media Lab na Tehnološkem inštitutu Massachusettsa (MIT) v ZDA. (Scratch Foundation, 2025) [10]

Prototip Scratch 0.1: Nastal je oktobra 2003.

Scratch 1.0: Uradna različica je bila izdana maja 2007.

Scratch 2.0: Izšla je maja 2013 in prinesla številne nove funkcije, kot so podpora za več brskalnikov in boljše orodje za ustvarjanje slik.

Scratch 3.0: Trenutno najnovejša različica, ki je bila izdana januarja 2019. Prinaša še boljše uporabniško izkušnjo, nove knjižnice in podporo za sodobne spletne tehnologije. (Scratch Foundation, 2025) [10]



Slika 1: Logotip Scratcha in glavna figura (vir: splet)

Datoteke za odpiranje v Scratchu so v naslednjih formatih:

- **.sb3** – je format projektnih datotek za **Scratch 3.0** (aktualna različica),
- **.sb2** – format je za **Scratch 2.0**,
- **.sb** – pa je stari format za **Scratch 1.x**.

Če delamo v aktualni spletni različici Scratcha (<https://scratch.mit.edu>) ali v aplikaciji za računalnik, potem uporabljamo **.sb3** datoteke. Torej, če si recimo shranimo projekt na računalnik, bo ime datoteke videti takole: **moj-projekt.sb3**

2.1.1 Pomen

Scratch ima velik vpliv na:

- **izobraževanje** – omogoča bolj interaktivno in učinkovito učenje programiranja,
- **družbo** – spodbuja digitalno pismenost in pripravlja mlade na prihodnost,
- **ekonomijo** – ustvarja nove možnosti za razvoj inovativnih izdelkov in storitev,
- **kultura** – omogoča ustvarjanje novih oblik umetnosti in izražanja.

Scratch je več kot le programski jezik, je orodje, ki oblikuje prihodnost in omogoča novim generacijam, da aktivno sodelujejo pri ustvarjanju digitalnega sveta.

Z njim je programiranje postalo dostopno vsem, ne glede na predznanje in starost.

(Gemini, 2024) [7]

2.1.2 Uporabnost

Scratch je odlično orodje za:

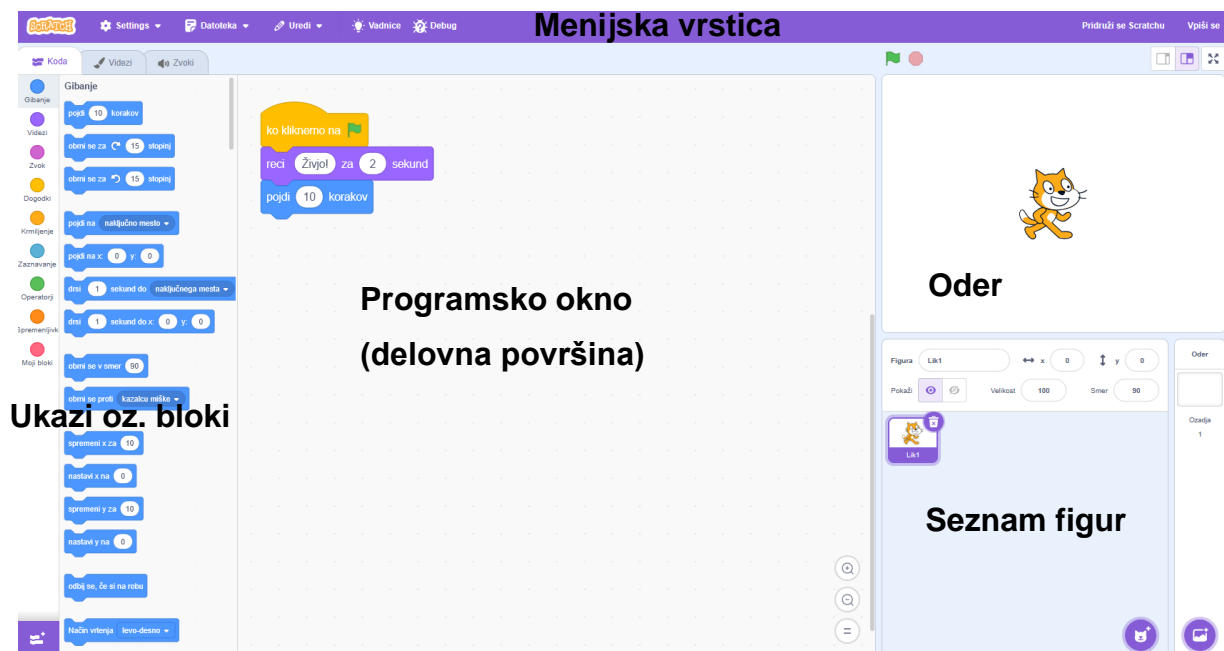
- **učenje** – osnov programiranja, računalniškega mišljenja in reševanja problemov,
- **ustvarjanje** – iger, animacij, zgodb in drugih interaktivnih vsebin,
- **razvoj** – kreativnosti, logičnega mišljenja in sodelovanja. (Gemini, 2024) [7]

Scratch je enostaven, saj lahko skozi igro spoznavamo osnove programiranja.

2.1.3 Grafični vmesnik

Programsko orodje Scratch si je moč namestiti na računalnik, deluje pa tudi v brskalnikih (npr. Mozilla Firefox, Google Chrome, Microsoft Edge). Najdemo ga na portalu <https://scratch.mit.edu/>, kjer si lahko ustvarimo svoj profil, na katerem imamo nato shranjene vse naše projekte.

Delo poteka s pomočjo grafičnega vmesnika, ki omogoča pisanje in preizkušanje programov, grafično oblikovanje in igranje igrice. Grafični vmesnik vsebuje več polj, ki so predstavljena na spodnji sliki. (Lajovic, 2011) [1]



Slika 2: Grafični vmesnik Scratcha (lasten vir)

Dodamo lahko tudi različne razširitve, kot so:

- glasba,
- svinčnik,
- video zaznavanje,
- besedilo v govor,
- prevedi,
- makey makey,
- micro:bit,
- Lego Mindstorms ev3,
- Lego Boost,
- Lego Education Wedo 2.0,
- go direct force & acceleration.

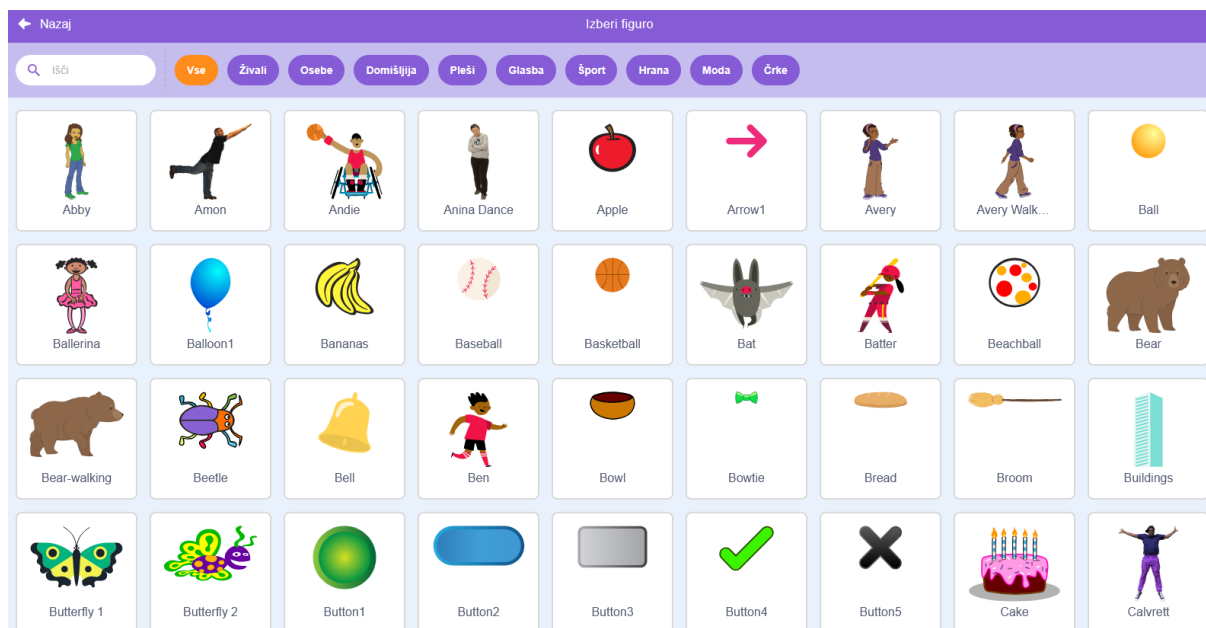
Ukazi oz. bloki so razdeljeni po različnih barvnih kategorijah ter po namenu uporabe.

2.1.4 Sprite ali figura

Glavni elementi Scratcha so bitja in predmeti, ki jim v angleščini pravimo *sprite*. V slovenščini jim pa pravimo *figure*.

Figuro lahko narišeš ali pa uporabiš katero koli sliko s svojega računalnika ali spleta.

- Figuri lahko daješ navodila, kako naj se premika, kakšno glasbo naj predvaja, kakšen bo njen videz. Zbirki navodil pravimo program ali skripta.
- Figuri lahko zamenjaš videz tako, da ji zamenjaš obleko.
- Vsaka figura ima lahko tudi svoj seznam zvokov, ki jih posnameš ali poiščeš na svojem računalniku ter svetovnem spletu. (Lajovic, 2011) [1]



Slika 3: Knjižnica vgrajenih figur (lasten vir)

2.2 Tekmovanje v Scratchu v Sloveniji

Kdo organizira tekmovanja iz Scratcha v Sloveniji?

V Sloveniji tekmovanja iz programiranja, vključno s tistimi v Scratchu, najpogosteje organizira **Zveza za tehnično kulturo Slovenije (ZOTKS)**. Ta organizacija spodbuja tehnično ustvarjalnost in znanje med mladimi ter organizira številna tekmovanja in dogodke na tem področju.

Katere ravni tekmovanja poznamo?

Tekmovanja iz Scratcha v Sloveniji se običajno izvajajo na dveh ravneh:

- **Šolsko tekmovanje:** Poteka na posameznih osnovnih šolah. Učenci se lahko prijavijo in tekmujejo med seboj.
- **Državno tekmovanje:** Poteka na centralni lokaciji, ki jo vsako leto določi ZOTKS. Na državno tekmovanje se uvrstijo najboljši tekmovalci s šolskih tekmovanj.

Kje potekajo tekmovanja?

- **Šolska tekmovanja:** Potekajo neposredno na šolah, v računalniških učilnicah ali v drugih primernih prostorih.

- **Državna tekmovanja:** Običajno potekajo na fakultetah, ki so povezane z računalništvom in informatiko, kot sta na primer Fakulteta za elektrotehniko, računalništvo in informatiko UM ter Fakulteta za računalništvo in informatiko UL.

Kaj je treba znati za tekmovanje iz Scratcha?

Za uspešno sodelovanje na tekmovanju iz Scratcha je potrebno dobro obvladati osnove programiranja v tem okolju. To vključuje:

- **Uporabo blokov:** Sposobnost izbiranja in povezovanja različnih blokov za ustvarjanje programov.
- **Osnovne koncepte programiranja:** Spremenljivke, pogoji, zanke, funkcije.
- **Ustvarjanje interaktivnih elementov:** Slike, zvok, animacije, ozadja.
- **Reševanje problemov:** Sposobnost razumevanja nalog in iskanja ustreznih rešitev. Problem razdeliti na manjše probleme.
- **Logično mišljenje:** Sposobnost razmišljanja v zaporedju in reševanje problemov korak za korakom. (Gemini, 2024) [7]

Zakaj se udeležiti tekmovanja iz Scratcha?

- **Pridobivanje znanja:** Tekmovanja so odlična priložnost za utrjevanje in nadgradnjo znanja iz programiranja.
- **Razvijanje veščin:** Tekmovanja spodbujajo razvoj logičnega mišljenja, kreativnosti in sposobnosti reševanja problemov.
- **Spoznavanje vrstnikov:** Tekmovanja omogočajo srečanje z drugimi mladimi, ki jih zanima programiranje.
- **Motivacija:** Tekmovanja so odličen način za motivacijo in spodbujanje nadaljnjega učenja. (Gemini, 2024) [7]

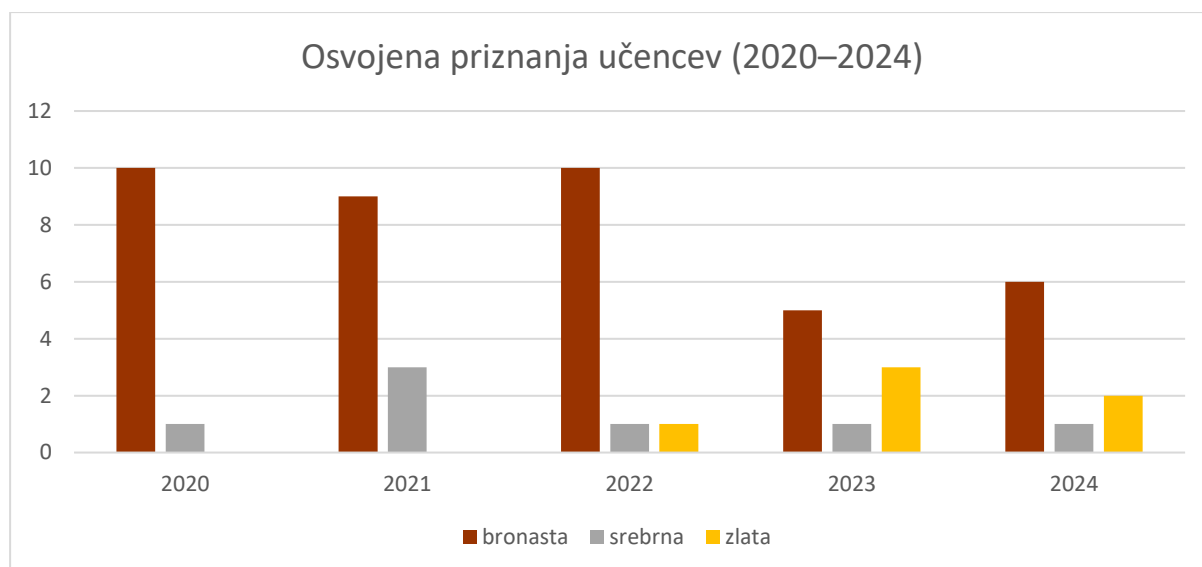
2.2.1 Uspešnost naše šole na tekmovanju

Naša šola je v zadnjih letih zelo uspešna na tekmovanju iz programiranja v Scratchu, zato prilagam nekaj statističnih podatkov.

Tabela 1: Število udeležencev na šolskem in državnem tekmovanju (Scratch) ter osvojena priznanja v zadnjih petih letih (vir: šolska spletna stran)

Število/leto	2020	2021	2022	2023	2024
število udeležencev na šolskem tekmovanju	20	10	11	13	10
število udeležencev na državnem tekmovanju	10	9	5	5	4
število osvojenih bronastih priznanj	10	9	10	5	6
število osvojenih srebrnih priznanj	1	3	1	1	1
število osvojenih zlatih priznanj	0	0	1	3	2

Graf 1: Osvojena priznanja učencev na šolskem in državnem tekmovanju (Scratch) v zadnjih petih letih (vir: šolska spletna stran)



2.3 Umetna inteligenca

Na zastavljeno vprašanje oz. poziv ["Kaj je umetna inteligenca?"] je orodje Gemini generiralo sledeč odgovor:

Umetna inteligenca (UI) je veja računalništva, ki se ukvarja z razvojem inteligentnih sistemov, torej sistemov, ki lahko simulirajo človeške kognitivne funkcije, kot so učenje, razumevanje, sklepanje, reševanje problemov in zaznavanje.

Primeri uporabe umetne inteligence:

- pametni pomočniki: Siri, Alexa, Google Assistant,

- samovozeči avtomobili,
- priporočilni sistemi (Netflix, Amazon),
- prevajanje jezikov,
- diagnostika bolezni,
- finančne napovedi,
- igre.

Prednosti umetne inteligence:

- **Avtomatizacija:** Poenostavi ponavljajoče se naloge.
- **Izboljšanje odločitev:** Omogoča boljše odločitve na podlagi podatkov.
- **Nove možnosti:** Odpira vrata novim inovacijam.

Izzivi umetne inteligence:

- **Etika:** Vprašanja zasebnosti, pristranskosti in odgovornosti.
- **Varnost:** Možnost zlorab.
- **Zaposlovanje:** Avtomatizacija lahko vpliva na trg dela.

Umetna inteligenca je hitro rastoče področje, ki bo v prihodnosti imelo velik vpliv na naše življenje. (Gemini, 2024) [7]

2.3.1 ChatGPT

Na zastavljena vprašanja oz. pozive je orodje ChatGPT generiralo sledeče odgovore.

Kaj je ChatGPT?

ChatGPT je napredni klepetalnik, ki temelji na velikem jezikovnem modelu in ga je razvilo podjetje OpenAI. Ta model je bil usposobljen na ogromni količini besedilnih podatkov, kar mu omogoča, da se pogovarja s teboj na način, ki je zelo podoben človeškemu. Lahko mu zastaviš vprašanja, mu daš navodila ali pa se z njim preprosto pogovarjaš o različnih temah.

Kdaj je nastal?

Prva različica ChatGPT je bila javnosti predstavljena konec leta 2022, vendar se razvoj tega modela nadaljuje in redno izhajajo nove ter izboljšane različice.

Kdo je avtor?

ChatGPT je razvilo podjetje OpenAI, ki je vodilna organizacija na področju razvoja umetne inteligence. OpenAI je znano po svojem delu na področju globokega učenja in razvoja velikih jezikovnih modelov.

Katere različice poznamo?

Obstaja več različic ChatGPT, ki se med seboj razlikujejo po zmogljivosti in funkcionalnosti. Najnovejše različice so običajno bolj zmogljive in lahko ustvarjajo bolj koherentne in smiselne odgovore.

Kaj nam omogoča?

ChatGPT nam omogoča:

- **Pogovarjanje v naravnem jeziku:** Z njim se lahko pogovarjaš o skoraj vsaki temi.
- **Dobivanje informacij:** Lahko mu zastaviš vprašanja o različnih temah in dobiš boš relevantne odgovore.
- **Pisanje besedil:** Lahko ga uporabiš za pisanje različnih vrst besedil, kot so eseji, pesmi, scenariji itd.
- **Prevajanje jezikov:** Lahko prevaja besedila iz enega jezika v drugega.
- **Programiranje:** Lahko ti pomaga pri pisanju kode.
- **Kreativno pisanje:** Lahko ustvarja zgodbe, pesmi in druge kreativne vsebine.

(ChatGPT, 2024) [11]

Pomembno je poudariti, da čeprav je ChatGPT zelo zmogljiv, ni popoln. Odgovori, ki jih daje, so lahko včasih napačni ali zavajajoči. Zato je vedno priporočljivo preveriti informacije iz več virov.

2.3.2 Gemini

Na zastavljena vprašanja oz. pozive je orodje Gemini generiralo sledeče odgovore.

Kaj je Gemini?

Gemini je najnovejši in najzmogljivejši velik jezikovni model, ki ga je razvil Google. Gre za vrhunski sistem umetne inteligence, ki je sposoben opravljati številne naloge,

povezane z jezikom, vključno s prevajanjem, pisanjem različnih vrst vsebin, odgovarjanjem na vprašanja in celo ustvarjanjem kode.

Kdaj je nastal?

Natančen datum nastanka ni povsem jasen, saj se modeli umetne inteligence nenehno razvijajo in izboljšujejo. Google pa je model Gemini predstavil javnosti šele v letu 2023.

Kdo je avtor?

Gemini je razvil Google, eno največjih tehnoloških podjetij na svetu. Google je v razvoj umetne inteligence vložil ogromna sredstva in ima dolgoletne izkušnje na tem področju.

Kje je na voljo?

Trenutno je dostop do Geminija delno omejen, vendar pa Google načrtuje, da bo v prihodnosti Gemini naredil bolj dostopnega širši javnosti.

Katere različice poznamo?

Google je predstavil več različic modela Gemini, ki se razlikujejo po zmogljivosti in namenu. Obstajajo različice, ki so optimizirane za različne naloge, kot so prevajanje, pisanje kode ter odgovarjanje na vprašanja.

Kaj nam omogoča?

Omogoča nam:

- Pogovarjanje v naravnem jeziku: Z njim se lahko pogovarjamo o različnih temah, podobno kot s človekom.
- Pisanje besedil: Lahko nam pomaga pri pisanju različnih vrst besedil, od e-pošte do esejev in celo računalniških programov.
- Prevajanje jezikov: Lahko prevaja besedila iz enega jezika v drugega.
- Iskanje informacij: Lahko nam pomaga najti informacije na spletu.
- Učenje: Lahko nam pomaga pri učenju novih stvari.
- Kreativno ustvarjanje: Lahko nam pomaga pri ustvarjanju različnih vrst vsebin, kot so pesmi, zgodbe in slike.

(Gemini, 2024) [8]

2.3.3 Copilot

Na zastavljena vprašanja oz. pozive je orodje Microsoft 365 Copilot generiralo sledeče odgovore.

Kaj je Copilot?

Copilot je generični izraz, ki se uporablja za opis različnih orodij umetne inteligence, zasnovanih tako, da delujejo kot virtualni pomočniki ali sodelavci. Ti sistemi uporabljajo napredne algoritme strojnega učenja, da analizirajo podatke, se učijo iz vzorcev in nato samodejno izvajajo naloge ali ponujajo predloge.

Kdaj je nastal?

Koncept Copilotov ima svoje korenine v zgodnjih raziskavah umetne inteligence, vendar je v zadnjih letih s hitrim razvojem velikih jezikovnih modelov doživel velik preboj. Natančen datum nastanka je težko določiti, saj se tehnologija nenehno razvija in izboljšuje.

Kdo je avtor?

Za Copilota ni enega samega avtorja. Različne tehnološke družbe in raziskovalne institucije razvijajo svoje različice Copilota, vsaka s svojimi specifičnimi funkcionalnostmi in namenom. Nekateri najbolj znani primeri Copilota vključujejo:

- **Microsoft 365 Copilot:** Integriran v Microsoft 365, ta Copilot pomaga pri ustvarjanju vsebin, analizi podatkov in avtomatizaciji delovnih tokov.
- **GitHub Copilot:** Zasnovan za razvijalce, ta Copilot predlaga kodo in pomaga pri odpravljanju napak.
- **Copilot (AI companion):** To je splošni AI sodelavec, ki lahko odgovarja na vprašanja, prevaja jezike in opravlja druge naloge.

Odločil sem se, da izberem aplikacijo oz. orodje Microsoft 365 Copilot.

Kje je na voljo?

Microsoft 365 Copilot je na voljo za različne vrste uporabnikov in organizacij. Na voljo je za podjetja vseh velikosti, vključno z malimi in srednje velikimi podjetji, prek Microsoftovih partnerjev Cloud Solution Provider (CSP). Prav tako je na voljo za uporabnike z naročninami na Microsoft 365 Business Standard ali Business Premium.

Poleg tega je Copilot vključen v naročnine na Microsoft 365 Personal in Family, kar pomeni, da ga lahko uporabljajo tudi posamezniki in družine.

Kaj nam omogoča?

Copiloti nam lahko pomagajo na številne načine, vključno z:

- **Avtomatizacijo nalog:** Copiloti lahko prevzamejo ponavljajoče se in rutinske naloge, kar nam omogoča, da se osredotočimo na bolj kompleksne in kreativne zadolžitve.
- **Izboljšanjem produktivnosti:** S predlaganjem idej, avtomatizacijo delovnih tokov in zagotavljanjem hitrega dostopa do informacij Copiloti lahko bistveno izboljšajo našo produktivnost.
- **Odkrivanjem novih možnosti:** Copiloti lahko analizirajo velike količine podatkov in nam pomagajo odkriti vzorce in trendi, ki bi jih sicer morda spregledali.
- **Učenjem novih veščin:** Copiloti lahko delujejo kot učitelji in nam pomagajo pri učenju novih veščin in znanj.

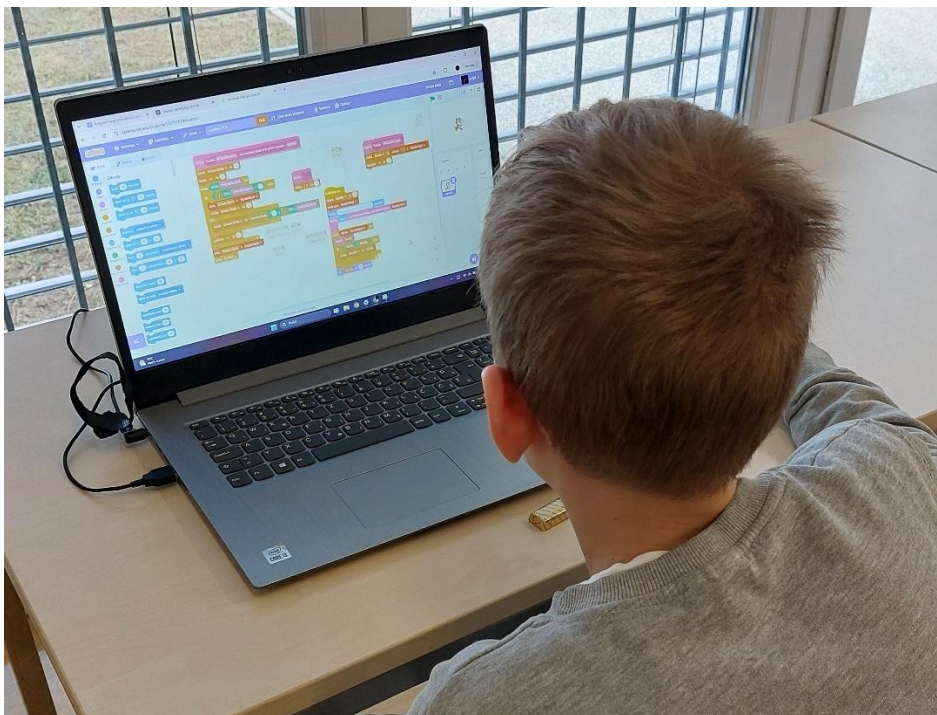
(Microsoft 365 Copilot, 2025) [9]

3 RAZISKOVALNI DEL

V tem poglavju bom predstavil celoten postopek svojega raziskovalnega dela.

Opis postopka:

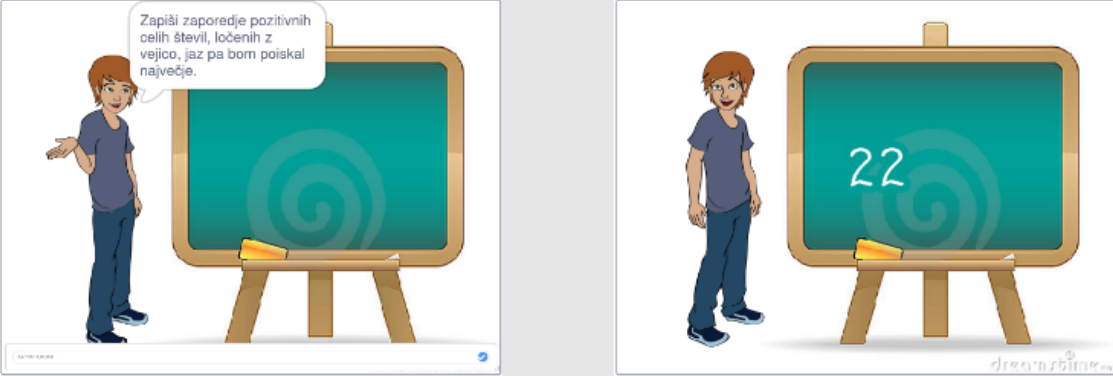
1. Najprej sem poiskal primer naloge s šolskega tekmovanja iz programiranja v Scratchu iz leta 2023 – 2. skupina (6. in 7. razred), ZOTKS.
2. Vsem trem aplikacijam UI (ChatGPT, Gemini in Microsoft 365 Copilot) sem podal enaka navodila:
*"Pomagaj mi rešiti spodnjo nalogo v Scratchu:
Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje."*
3. Nato sem jim postavil še enaka dodatna vprašanja.
4. Testiral sem njihove rešitve ter podal predloge za izboljšavo.
5. Njihove predloge sem primerjal z rešitvijo ZOTKS.
6. Po analizi sem zapisal svoje rezultate in ugotovitve.



Slika 4: Potek dela na raziskovalnem taboru (lasten vir)

Največje število

Oder



Naloga

Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje.

Napotek

Uporabite nastavek naloge na datoteki maks-naLOGA.sb3.

Slika 5: Naloga, ki sem jo izbral za svoje raziskovanje (vir: splet, ZOTKS)

3.1 Primer rešitve z uporabo ChatGPT

3.1.1 Odgovor (predlog rešitve) aplikacije UI – ChatGPT

Na zastavljeno vprašanje oz. poziv [*"Pomagaj mi rešiti spodnjo nalogo v Scratchu: Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje."*] je orodje ChatGPT generiralo sledeč odgovor:

Da bi rešili to nalogo v Scratchu, bomo morali ustvariti skripto, ki omogoča vnos števil, jih loči, prebere in nato poišče največje število. To lahko narediš v naslednjih korakih:

1. Dodaj spremenljivko: Ustvari spremenljivko, npr. števila, ki bo shranila vsa števila.
2. Vnos števil:
 - Uporabi blok zberi vnos (ali vprašaj [Katero zaporedje števil želiš vpisati?]) in počakaj), da uporabnik vnese številke ločene z vejico.
3. Razdelitev na števila:

- Uporabi blok razdeli za razdelitev vnosa na posamezna števila, ki so ločena z vejico. To bo ustvarilo seznam števil.

4. Iskanje največjega števila:

- Uporabi zanko (npr. ponavljanje za vsako), da preveriš vsako število v seznamu.
- Dodaj blok če [število > največje število], da boš preveril, ali je trenutna številka večja od trenutno največjega števila.
- Če je, posodobi največje število.

5. Izpis rezultata:

- Ko najdeš največje število, uporabi blok povej [največje število] za izpis rezultata.

Primer Scratch kode:

Vprašaj uporabnika za vnos števil:

scratch

KopirajUredi

vprašaj [Katero zaporedje števil želiš vpisati?] in počakaj

Razdeli vnos:

scratch

KopirajUredi

set [števila v seznamu v] to (razdeli (odgovor) z [','])

Zanka za iskanje največjega števila:

scratch

KopirajUredi

set [največje število v] to [0]

ponavljanje za vsak [število] v [seznam števil v]

če <(število) > (največje število)> potem

set [največje število v] to (število)

konec

konec

Izpis največjega števila:

scratch

KopirajUredi

povej (največje število v)

Tako boš lahko v Scratchu vnesel zaporedje števil, našel največje število in ga izpisal! (ChatGPT, 2024) [11]

3.1.2 Ali mi lahko narediš barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3?

Orodje ChatGPT je generiralo sledeč odgovor na zgornje vprašanje:

Trenutno ne morem neposredno ustvariti .sb3 datoteke, ampak ti lahko podrobno razložim, kako sestaviti bloke v Scratchu. Lahko pa ti dam slikovni prikaz blokov, če želiš.

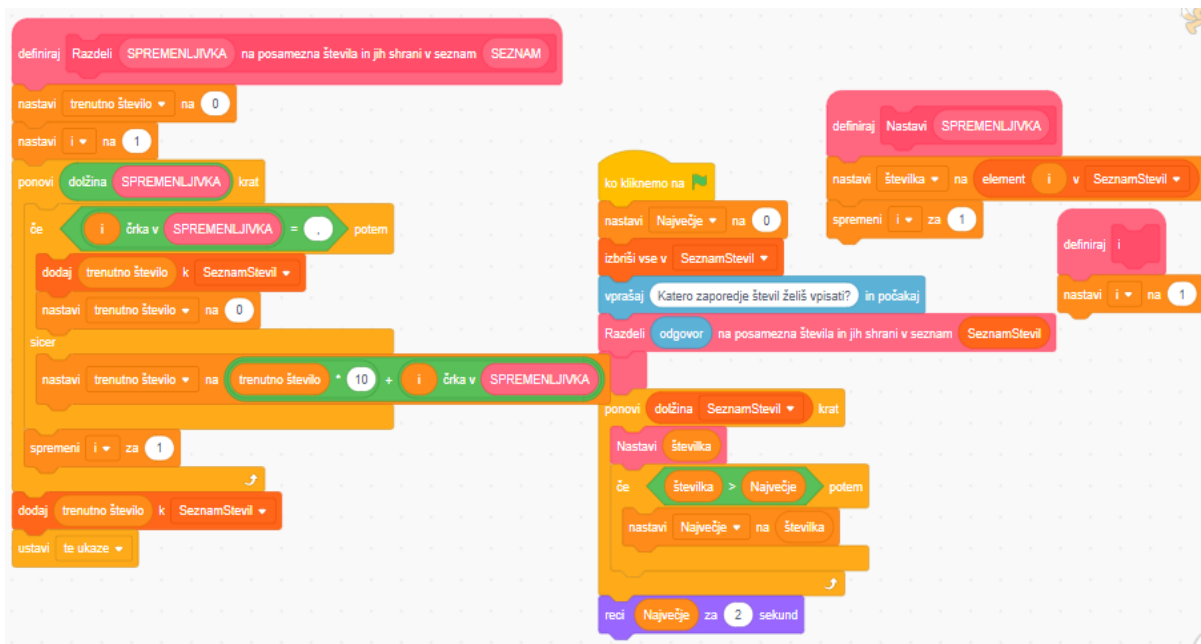
Če želiš .sb3 datoteko, lahko Scratch projekt ustvariš sam na Scratch spletni strani ali v Scratch Offline Editor in slediš mojim navodilom. Lahko ti pomagam tudi z natančnimi navodili za vsak korak.

Ali bi želel slikovni prikaz Scratch blokov ali podrobna navodila? (ChatGPT, 2024) [11]

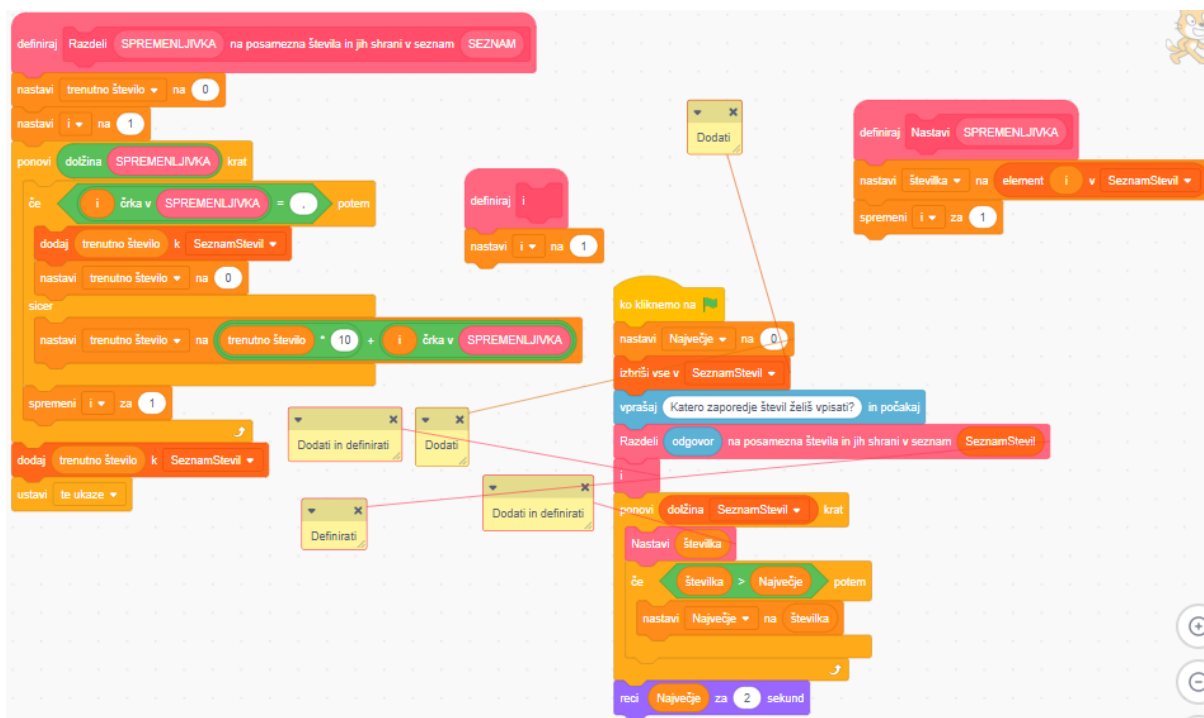
3.1.3 Moje ugotovitve

Predlagano rešitev sem najprej testiral v Scratchu, nato pa primerjal s pravo rešitvijo (ZOTKS, poglavje 3.4).

Program ChatGPT je v povprečju potreboval 12,03 sekunde, da je izpisal svoj odgovor. Rešitev ni preveč ustrezna, saj je razlaga preveč posplošena, ne pozna blokov, ne navaja definicije spremenljivk, seznamov ter slabo oblikuje komuniciranje programa z uporabnikom (npr. namesto stavka 'Največje število je: [največje_število]' on napiše zgolj '[največje_število]'). Vendar ob uporabi svojih zamisli ter dodatnih pozivov je lahko koristen pomočnik pri programiranju v Scratchu. Ker je podal dva odgovora, sem iz obeh izluščil bistvo, saj sta se nekoliko dopolnjevala.



Slika 6: Fotografija kode (blokov), ki jo je predlagal ChatGPT (lasten vir)



Slika 7: Fotografija kode (blokov), ki jo je bilo potrebno dodati (lasten vir)

Na osrednji kodi (tisti, ki se začne z zeleno zastavico) so izvečeni komentarji, ki prikazujejo, kaj je bilo potrebno spremeniti na določenih blokih. Dodati je bilo potrebno tudi vse bloke oz. ukaze izven osrednje kode (tisto, kar se začne z roza bloki).

3.2 Primer rešitve z uporabo Gemini

3.2.1 Odgovor (predlog rešitve) aplikacije UI – Gemini

Na zastavljeno vprašanje oz. poziv ["Pomagaj mi rešiti spodnjo nalogo v Scratchu: Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje."] je orodje Gemini generiralo sledeč odgovor:

Reševanje naloge v Scratchu: Iskanje največjega števila

1) Razumevanje naloge

Naloga zahteva, da v Scratchu ustvarimo program, ki bo:

1. **Zbral** vsa vnesena števila.
2. **Primerjal** med seboj vsa vnesena števila.
3. **Izpisal** največje število.

2) Rešitev v Scratchu

1. Ustvari nove spremenljivke:

- **stevilo**: Za shranjevanje trenutno vnesenega števila.
- **najvecje_stevilo**: Za shranjevanje največjega najdenega števila.
- **seznam_stevil**: Za shranjevanje vseh vnesenih števil.

2. Ustvari skript:

- Ko kliknem na zastavico:
 - Postavi najvecje_stevilo na 0.
 - Počisti vse v seznamu seznam_stevil.
 - Vprašaj "Vpiši števila, ločena z vejicami:" in odgovor shrani v spremenljivko vnesena_stevila.
 - Razdeli vnesena_stevila na posamezna števila in jih shrani v seznam* seznam_stevil.
 - Za vsako število v seznamu seznam_stevil:
 - Če je trenutno število** večje od najvecje_stevilo:
 - Postavi najvecje_stevilo na trenutno število**.
 - Izpiši "Največje število je: " & najvecje_stevilo.

3) Pojasnilo blokov

- **Vprašaj in odgovor**: S tem blokom uporabnika vprašamo po številkah in shranimo odgovor.

- **Razdeli:** Ta blok razdeli vneseni niz števil na posamezne elemente (števila) in jih shrani v seznam.
- **Za vsako:** Ta blok ponavlja določeno dejanje za vsak element v seznamu.
- **Če ... potem:** Ta blok preveri pogoj in izvede določene ukaze, če je pogoj izpolnjen.

4) Vizualni prikaz v Scratchu

[Tukaj bi moral biti vstavljen posnetek zaslona Scratch projekta. Ker to ni mogoče neposredno v besedilu, ti priporočam, da si ustvariš nov projekt v Scratchu in slediš zgornjim navodilom. Če potrebuješ dodatno pomoč, lahko poiščeš video vadnice na YouTubeu.]

5) Dodatne opombe

- **Preverjanje vnosa:** Lahko dodaš blok, ki preveri, ali je uporabnik vnesel samo številke.
- **Napake pri vnosu:** Če uporabnik vpiše neveljavne podatke, lahko obdelaš te napake in uporabnika opozoriš.
- **Uporabniški vmesnik:** Lahko oblikuješ lepši uporabniški vmesnik z ozadji, slikami in bolj informativnimi sporočili.

S tem programom boš lahko učinkovito našel največje število iz poljubnega zaporedja števil, ki ga boš vnesel.

Ali potrebuješ dodatno pomoč pri ustvarjanju tega projekta v Scratchu? Rad ti bom pomagal pri reševanju morebitnih težav.

Če želiš, lahko ti pomagam tudi pri reševanju naloge v drugem programskem jeziku.

Prosim, sporoči mi, če imaš kakšno dodatno vprašanje. (Gemini, 2024) [8]

3.2.2 Ali mi lahko narediš barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3?

Orodje Gemini je generiralo sledeč odgovor na zgornje vprašanje:

Na žalost ne morem neposredno ustvariti barvnih blokov, kot jih vidiš v Scratchu, ne pa tudi .sb3 datoteke.

Kot jezikovni model nimam vizualnega vmesnika, kot ga ima Scratch. Moja naloga je, da ti pomagam razumeti koncept programiranja in ti zagotovim navodila, kako lahko sam ustvariš svoj projekt v Scratchu.

Kaj pa lahko naredim zate:

- **Podrobnejša navodila:** Lahko ti zagotovim še bolj podrobna navodila za vsak korak v tvojem projektu.
- **Razlaga kodnih blokov:** Če imaš težave z razumevanjem posameznih blokov, ti lahko razložim njihovo funkcijo.
- **Alternativne rešitve:** Lahko ti predlagam različne pristope k reševanju problema.
- **Odpravljanje napak:** Če naletiš na napako v svojem programu, ti lahko pomagam pri iskanju in odpravljanju.

Kako lahko skupaj ustvarimo tvoj projekt:

1. **Opisi mi težavo:** Povej mi natančno, kje se zatakneš ali kaj ne razumeš.
2. **Pošlji mi posnetek zaslona:** Če imaš že kakšen del projekta narejen, mi pošlji posnetek zaslona, da vidim, kaj si naredil.
3. **Vprašaj specifična vprašanja:** Namesto splošnih vprašanj, mi postavi čim bolj specifična vprašanja.

Primer:

Namesto, da vprašaš: "Kako naj naredim, da se mačka premika?" lahko vprašaš: "Kako naj naredim, da se mačka premika za 10 korakov naprej, ko kliknem na zeleno zastavico?"

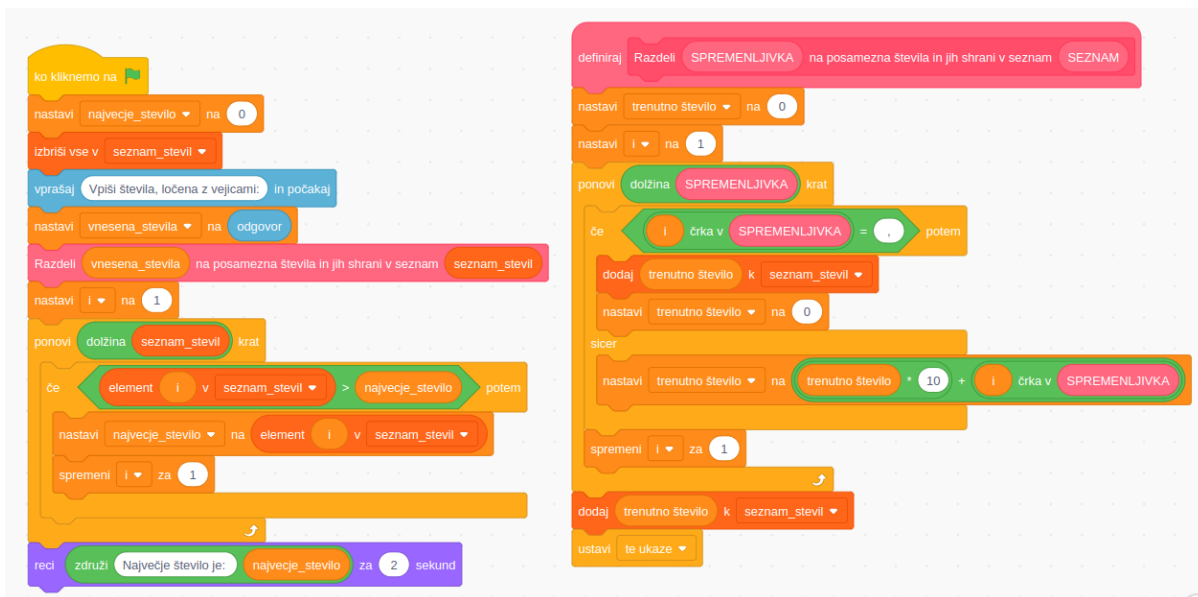
S sodelovanjem lahko skupaj ustvarimo odličen projekt v Scratchu!

Ali imaš kakšno konkretno vprašanje glede tvojega projekta? (Gemini, 2024) [8]

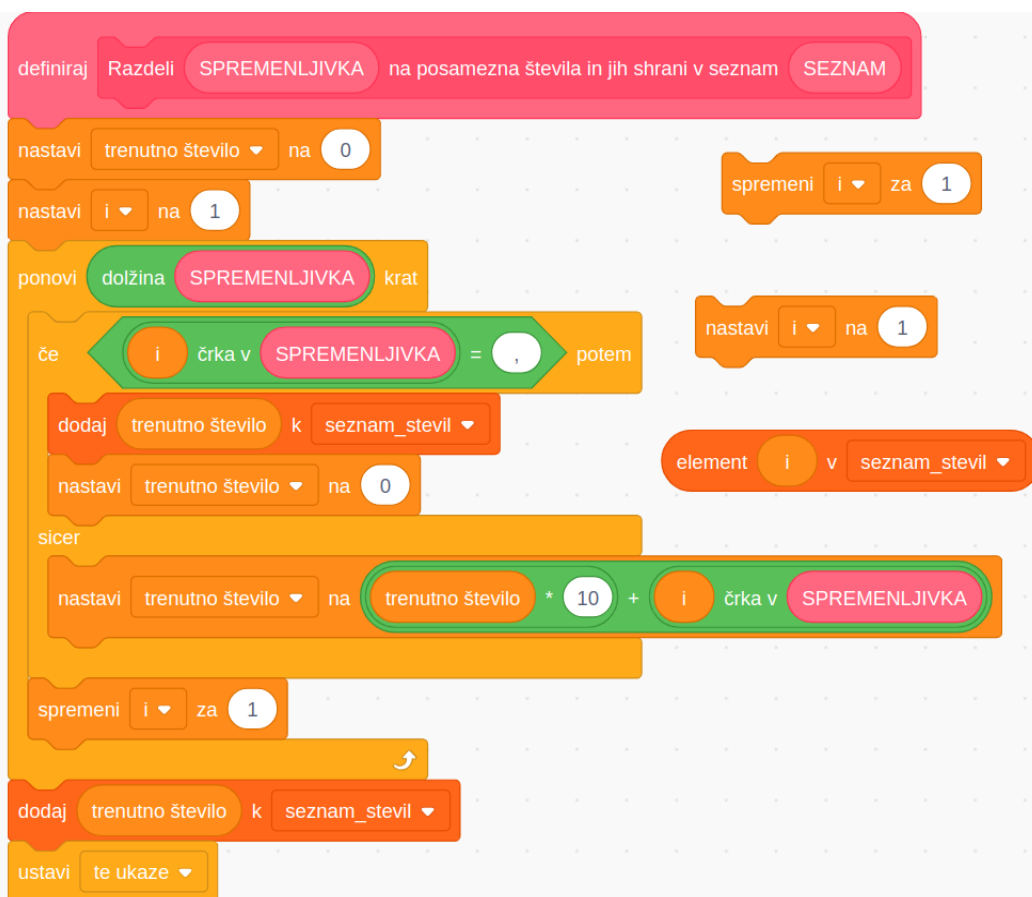
3.2.3 Moje ugotovitve

Predlagano rešitev sem ponovno testiral v Scratchu, nato pa jo primerjal s pravo rešitvijo naloge (ZOTKS, poglavje 3.4).

Program Gemini je v povprečju potreboval 8 sekund, da je izpisal svoj odgovor. Rešitev je ustrezna, vendar zaradi izmišljenega bloka žal ne deluje. [razlike]. [mnenje]. Ugotovil sem, da je Geminijev problem predvsem to, da ne pozna blokov – vendar lahko ob dodatnem pozivu izključi uporabo neobstojećih blokov in s tem ustvari delujoč program.



Slika 8: Fotografija kode (blokov), ki jo je predlagal Gemini (lasten vir)



Slika 9: Fotografija kode (blokov), ki jo je bilo potrebno dodati (lasten vir)

3.3 Primer rešitve z uporabo Microsoft 365 Copilot

3.3.1 Odgovor (predlog rešitve) aplikacije UI – Copilot

Na zastavljeno vprašanje oz. poziv ["Pomagaj mi rešiti spodnjo nalogo v Scratchu: Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje."] je orodje Copilot generiralo sledeč odgovor:

Seveda, pomagam ti z veseljem! Tukaj je primer, kako lahko rešiš to nalogo v Scratchu:

1. Ustvari novo spremenljivko "Največje število".
2. Uporabi blok "vprašaj in čakaj", da uporabniku omogočiš vnos zaporedja števil.
3. Shranjena odgovore v spremenljivko, recimo "Vnos".
4. S pomočjo bloka "razdeli () glede na ()" razdeli vnos v seznam števil.
5. Preberi številke iz seznama in poišči največje.

Tukaj je koda v Scratchu:

1. Začnemo z začetkom dogodka "ko kliknemo na zeleno zastavico":

ko kliknemo na zeleno zastavico

nastavi [Največje število v] na (0)

vprašaj [Vnesi zaporedje števil, ločenih z vejico:] in čakaj

nastavi [Vnos v] na (odgovor)

razdeli [Vnos v] glede na [,] in shrani v [Seznam v]

2. Nato preberemo seznam in poiščemo največje število:

za vsak [Število v] v [Seznam v]

če <(Število) > (Največje število)> potem

nastavi [Največje število v] na (Število)

konec

konec

3. Na koncu izpišemo največje število:

povej (združi [Največje število je] [Največje število])

Ta koda bo uporabnika pozvala, da vnese zaporedje števil, razdelila vnos v seznam, prebrala seznam in našla največje število ter ga nato izpisala.

Če imaš kakšna vprašanja ali potrebuješ dodatno pomoč, se kar oglasi!

(Microsoft 365 Copilot, 2024) [9]

3.3.2 Ali mi lahko narediš barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3?

Orodje Microsoft 365 Copilot je generiralo sledeč odgovor na zgornje vprašanje:

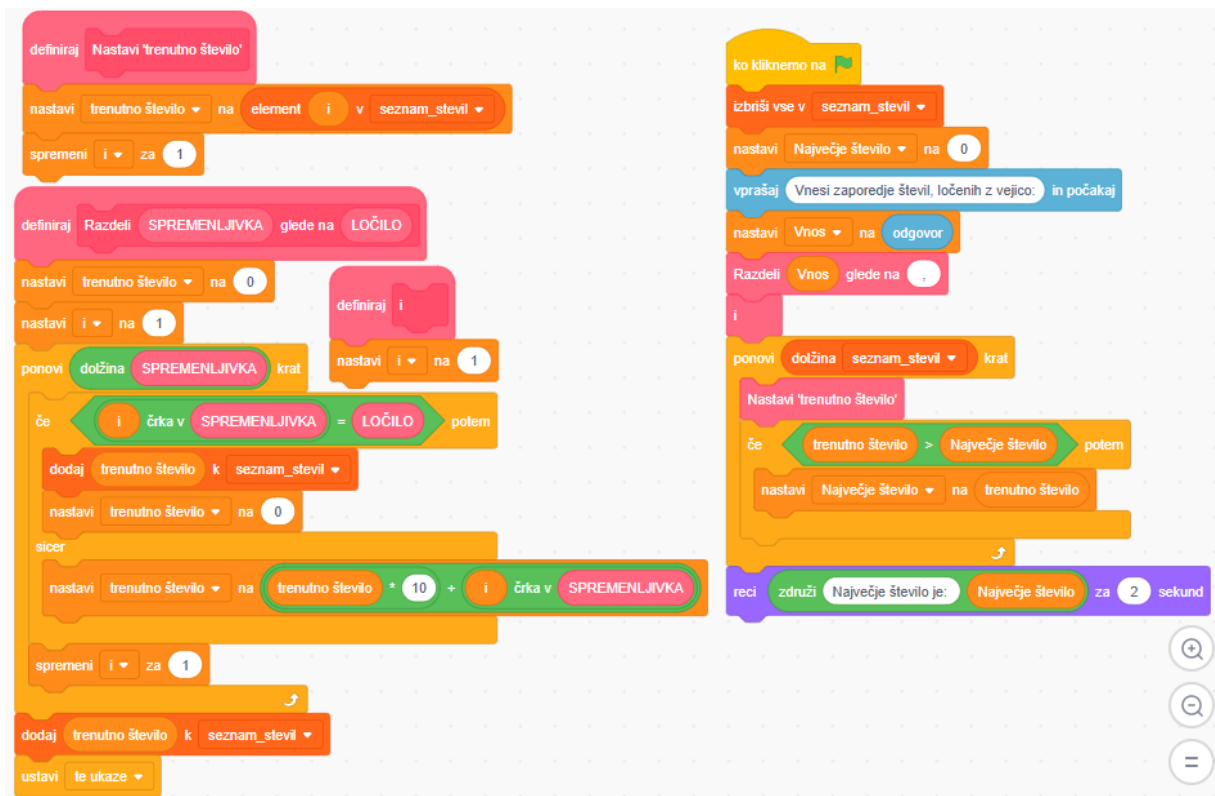
Na žalost ne morem neposredno ustvariti Scratch datoteke .sb3 ali prikazati barvnih blokov, kot jih vidimo v Scratchu. Lahko pa te vodim skozi postopek, da boš sam ustvaril te bloke v Scratchu. (Microsoft 365 Copilot, 2024) [9]

3.3.3 Moje ugotovitve

Predlagano rešitev sem ponovno testiral v Scratchu, nato pa jo primerjal s pravo rešitvijo naloge (ZOTKS, poglavje 3.4).

Program Microsoft 365 Copilot je v povprečju potreboval 6,73 sekunde, da je izpisal svoj odgovor. Rešitev je ustrezna, vendar zaradi izmišljenega bloka žal ne deluje. Toda podal je natančen, kratek in jedrnat odgovor. Ob dodatnem pozivu bi gotovo izključil uporabo neobstojećih blokov in s tem ustvari delujoč program.

Izmed vseh treh aplikacij UI je Microsoft 365 Copilot bil še najbližje rešitvi. Najhitreje je tudi zgeneriral odgovore na moja vprašanja oz. pozive.



Slika 10: Fotografija kode (blokov), ki jo je predlagal Copilot (lasten vir)

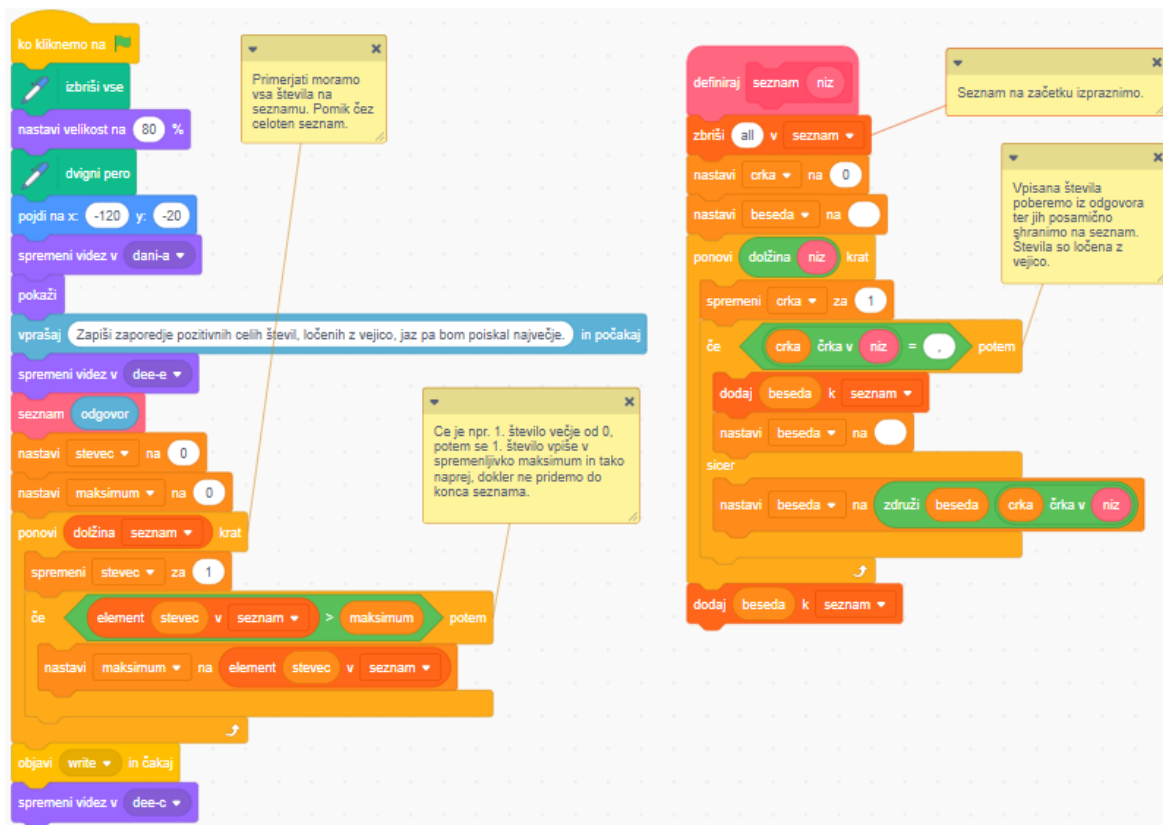


Slika 11: Fotografija kode (blokov), ki jo je bilo potrebno dodati (lasten vir)

Na osrednji kodi (tisti, ki se začne z zeleno zastavico) so izvečeni komentarji, kaj je bilo potrebno spremeniti na določenih blokih. Dodati je bilo potrebno tudi vse izven osrednje kode (tisto, kar se začne z roza bloki).

3.4 Rešitev ZOTKS

Na portalu Zveze za tehnično kulturo Slovenije sem poiskal rešitev naloge iz šolskega tekmovanja Scratch 2023 – 2. skupina (6. in 7. razred) [2]. Ugotovil sem, da je način razmišljanja vseh aplikacij UI podoben, vendar je razlika v tem, da rešitve vključujejo neobstoječe bloke oziroma jih z možnostjo kreiranja lastnega bloka oz. funkcije ustvari in definira. Se pravi, način razmišljanja je pri vseh treh primerih podoben, vendar vse tri aplikacije umetne inteligence uporabljajo neobstoječi blok v smislu [Razdeli {SPREMENLJIVKA} po {,} in shrani v seznam].



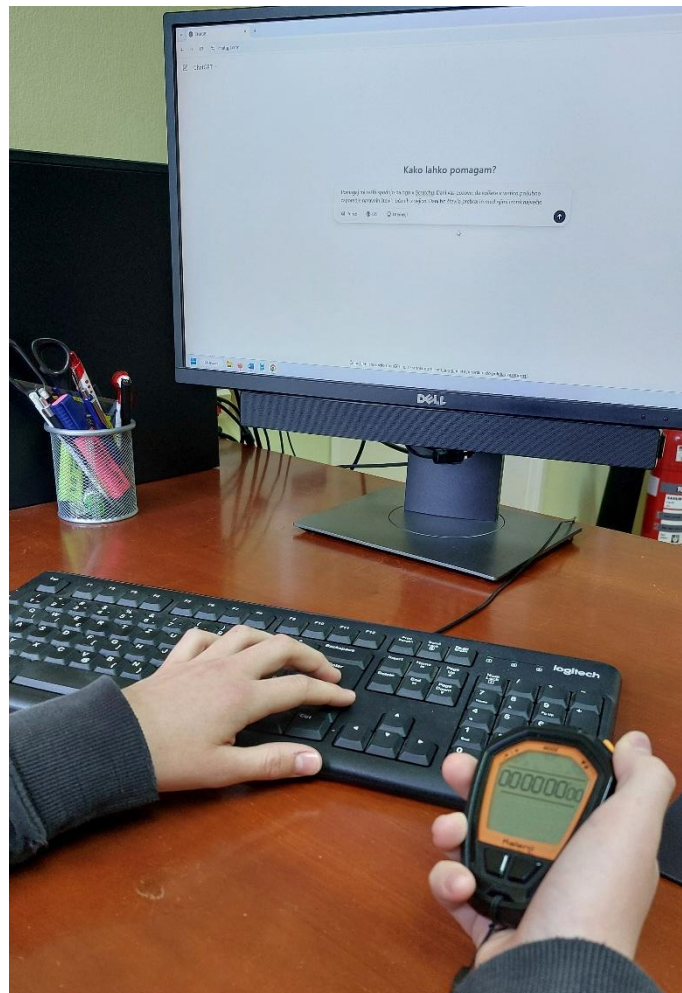
Slika 12: Koda (bloki) rešitve naloge (vir: ZOTKS)

4 REZULTATI IN UGOTOVITVE

Rezultati in ugotovitve moje raziskovalne naloge so sledeče.

Pri raziskovanju me je zanimal tudi čas, ki so ga orodja UI porabila za izdelavo odgovora oz. predloga rešitve. Testiranja so bila opravljena na enakem ter zelo zmogljivem računalniku, hitrem žičnem internetu (optika) ter ob različnih časih in dnevih.

Čas sem meril za naslednje vprašanje oz. poziv: *"Pomagaj mi rešiti spodnjo nalogo v Scratchu: Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje."*

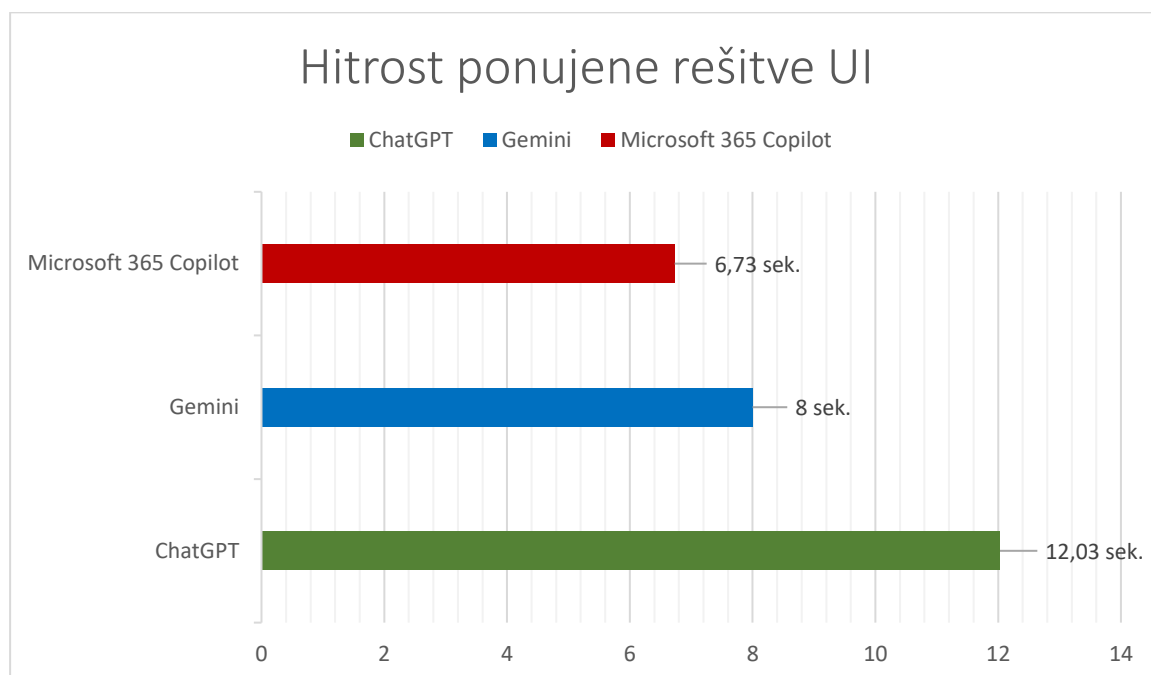


Slika 13: Merjenje časa generiranja odgovora UI s štoparico (lasten vir)

Tabela 2: Podatki merjenja hitrosti generiranja odgovora na vprašanje (poziv UI)

Orodje UI / Dan	Čas v sekundah			
	Ponedeljek (zjutraj)	Sreda (popoldan)	Nedelja (zvečer)	Povprečni čas
ChatGPT	8,2	15,4	12,5	12,03
Gemini	5,6	9,7	8,7	8,00
Microsoft 365 Copilot	6,6	7,2	6,4	6,73

Graf 2: Kako hitro je aplikacija UI ponudila rešitev naloge (čas v sekundah)



Opazimo lahko, da je odgovor v povprečju najhitreje ustvaril Microsoft 365 Copilot, katerega odgovor je tudi bil najbolj blizu rešitvi ZOTKS. Vsa tri orodja UI so dokaj hitro ponudila predlog rešitve, vendar nobena ni bila popolnoma pravilna. Vsak odgovor je bilo treba še nato dodelati, da je projekt v Scratchu deloval, kot zahtevajo navodila naloge. Nobena od uporabljenih aplikacij UI tudi ni znala zgenerirati delujoče .sb3 datoteke. So pa aplikacije v vmesnem času že napredovale tudi v tej smeri.

4.1 Mnenja učiteljev

Učitelj A: Raziskovalna naloga ponuja odličen vpogled v uporabo umetne inteligence pri programiranju v Scratchu. Avtor je jasno predstavil prednosti in omejitve orodij oz. aplikacij UI ter izpostavil pomen kritičnega razmišljanja pri njihovi uporabi. Vsebina je

dobro strukturirana in analitično podprta, kar omogoča bralcem, da pridobijo realen vpogled v to, kako umetna inteligenca lahko pomaga pri učenju programiranja.

Predlogi izboljšav programske kode, ki jih je podal učenec, so ustrezni in nazorno (slikovno) predstavljeni.

Učitelj B: Naloga na razumljiv in sistematičen način preučuje vpliv umetne inteligence na poučevanje programiranja v osnovnih šolah. Še posebej lahko pohvalim analizo različnih orodij UI in njihovo primerjavo z realnimi rešitvami. Rešitve, ki jih je podal učenec so pravilne, testirane ter jasno predstavljene.

Izbrana tema je zelo aktualna, ki lahko služi kot osnova za nadaljnje raziskave in izboljšanje metod poučevanja programiranja.

Učitelj C: Zelo cenim trud in raziskovalni pristop, ki je bil uporabljen v tej nalogi. Ugotovitve učenca so relevantne in prinašajo pomembna spoznanja o omejitvah ter prednostih umetne inteligence v izobraževanju. Strinjam se s predlaganimi rešitvami projektov (nalog) učenca, saj vse privedejo do zelenega rezultata oz. pravilne rešitve podane naloge.

Posebej bi poudaril pomen, ki ga avtor pripisuje odgovorni uporabi UI in potrebi po preverjanju rezultatov, kar je ključnega pomena pri izobraževanju prihodnjih generacij programerjev.

4.2 Hipoteza 1

H1: ChatGPT mi ne bo podal prave rešitve naloge.

Hipoteza 1 je potrjena.

Obrazložitev: Rešitev ni ustrezna, saj je razlaga preveč posplošena in potrebno še je veliko pozivov ter dela, da koda začne delati. Hipoteza 1 je potrjena, saj ChatGPT ni podal ustrezne rešitve naloge.

4.3 Hipoteza 2

H2: Gemini mi bo podal popolnoma ustrezno rešitev naloge.

Hipoteza 2 je ovržena.

Obrazložitev: Podana rešitev je delno ustrezna, a zaradi izmišljenega bloka ne deluje.

4.4 Hipoteza 3

H3: Microsoft 365 Copilot bo znal ustvariti .sb3 datoteko.

Hipoteza 3 je ovržena.

Obrazložitev: Orodje ne zna ustvariti Scratch datoteke .sb3 ali prikazati barvnih blokov.

4.5 Hipoteza 4

H4: Orodja UI nam lahko pomagajo pri programiranju v Scratchu.

Hipoteza 4 je potrjena.

Obrazložitev: Na podlagi mojih rezultatov lahko zatrdim, da nam UI lahko pomaga pri programiranju v Scratchu.

4.6 Hipoteza 5

H5: Učitelji računalništva se bodo strinjali z mojimi predlogi izboljšav.

Hipoteza 5 je potrjena.

Obrazložitev: Vsi trije učitelji so se v njihovih mnenjih strinjali z mojim predlogi izboljšav oz. rešitvami, zato je hipoteza 5 potrjena.

5 DRUŽBENA ODGOVORNOST, TRAJNOST, NAPREDEK

Umetna inteligenca (UI) postaja vse bolj prisotna v našem vsakdanu, od pametnih telefonov in virtualnih asistentov do kompleksnih algoritmov, ki analizirajo podatke in sprejemajo odločitve namesto nas. Kljub številnim koristim, ki jih prinaša, pa je pomembno tudi kritično presojanje njenega vpliva ter zavedanje morebitnih nevarnosti in osveščanje uporabnikov o tem.

5.1 Koristi umetne inteligence

UI omogoča avtomatizacijo nalog, povečuje učinkovitost in izboljšuje produktivnost. Pospešuje analizo podatkov, omogoča personalizacijo storitev ter napredek v medicini in ostalih področjih. Pomaga pri izboljšanju delovnih procesov, kjer zaposleni lahko opravljajo bolj ustvarjalne naloge, hitreje in učinkovitejše.

5.2 Nevarnosti in pomisleki

UI lahko prispeva k širjenju dezinformacij in manipulaciji z vsebinami. Obstaja tveganje pristranskosti algoritmov, ki lahko vodijo do nepoštenih odločitev. zasebnost je ogrožena zaradi množičnega zbiranja podatkov. Prav tako lahko avtomatizacija povzroči izgubo delovnih mest in poveča varnostna tveganja.

5.3 Kritična presoja in odgovorna uporaba

Uporabniki morajo preverjati vire informacij, ki jih generira UI. Pomembno je zavedanje o pristranskosti algoritmov in uporaba ustreznih varnostnih nastavitev za zaščito zasebnosti. Etična uporaba UI zahteva upoštevanje regulacij in smernic, da bi zmanjšali morebitna tveganja. Evropska unija je v ta namen že sprejela akt o umetni inteligenci¹.

Umetna inteligenca ima velik potencial za izboljšanje kakovosti življenja, vendar je ključno, da jo uporabljamo odgovorno. Z ozaveščanjem o njenih koristih in nevarnostih lahko kot družba bolje izkoristimo njene prednosti in se hkrati učinkovito spopademo z izzivi, ki jih prinaša.

6 ZAKLJUČEK

V raziskovalni nalogi sem preučeval vlogo umetne inteligence pri programiranju v Scratchu ter ugotavljal, kako lahko orodja, kot so ChatGPT, Gemini in Microsoft 365 Copilot, pomagajo pri reševanju nalog. Moja analiza je pokazala, da vsa preizkušena orodja UI omogočajo določeno stopnjo pomoči pri programiranju, vendar je njihova uporaba omejena zaradi nepopolnega razumevanja strukture Scratcha in nepoznavanja specifičnih ukazov (blokov) v tem programskem okolju. Medtem ko lahko umetna inteligenca poda splošna navodila in osnutke rešitev, je za pravilno implementacijo še vedno potrebna kritična presoja uporabnika.

¹Prvega avgusta 2024 je začel veljati evropski akt o umetni inteligenci. Namen akta je spodbujati odgovoren razvoj in uvajanje umetne inteligence v EU.

Akt o umetni inteligenci, ki ga je komisija predlagala aprila 2021 in o katerem sta se decembra 2023 dogovorila Evropski parlament in Svet, obravnava morebitna tveganja za zdravje, varnost in temeljne pravice državljanek in državljanov. Razvijalcem in uvajalcem nalaga jasne zahteve in obveznosti, kar zadeva posebne uporabe umetne inteligence, hkrati pa zmanjšuje upravna in finančna bremena za podjetja. (Evropska komisija, Začetek veljavnosti akta o umetni inteligenci, dostopno na: https://commission.europa.eu/news/ai-act-enters-force-2024-08-01_sl, 10. 2. 2025).

Poleg tehničnih vidikov sem se dotaknil tudi vprašanja družbene odgovornosti, trajnosti in napredka. Umetna inteligenca ima ogromen potencial za izboljšanje učnega procesa, vendar njena uporaba zahteva zavedanje o njenih omejitvah. Ključno je, da se uporabniki naučijo kritičnega razmišljanja in preverjanja podanih informacij. Prav tako je pomembno, da se umetna inteligenca uporablja odgovorno in etično, če želimo, da prispeva k napredku družbe in izobraževanja.

Rezultati moje raziskave potrjujejo, da orodja UI lahko pripomorejo k lažjemu reševanju programerskih nalog v Scratchu, vendar niso nadomestilo za razumevanje osnov programiranja in logičnega razmišljanja. Njihova učinkovitost je največja, če se uporabljajo kot pomožno orodje, ki spodbuja učenje in kreativnost, ne pa kot avtomatska rešitev. Menim, da bo prihodnji razvoj umetne inteligence morda odpravil nekatere trenutne pomanjkljivosti, vendar bo kljub temu vloga človeka pri programiranju in razvoju algoritmov ostala nenadomestljiva.

Ob raziskovanju mi je bilo zelo zanimivo, kaj vse umetna inteligenca zmore, kako zelo koristna in uporabna je lahko. Spoznal sem, da lahko kljub napakam in spodrslijajem ob dodatnih pozivih ustvarja vedno boljše in koristnejše zadeve, pa naj bo to pesem, strip, besedilo, slika, video ...

V prihodnosti bi lahko nalogo nadgradil tako, da bi raziskoval več modelov umetne inteligence, predvsem tistih, ki so še bolj specializirane za programiranje (Qodo, Github Copilot, Windsurf, TurboWarp, Claude idr.), jih pozval k rešitvi več primerov nalog, v več programskih jezikih ipd.

7 VIRI IN LITERATURA

7.1 Literatura

[1] S. Lajovic, Scratch: [nauči se programirati in postani računalniški maček], 1. natis. Ljubljana: Pasadena, 2011, str. 15–16.

7.2 Spletni viri

[2] B. Jurčič Zlobec, "Šolsko tekmovanje – Scratch (januar 2023) – 2. skupina," Zveza za tehnično kulturo Slovenije. [Online]. Dostopno na: <https://zotks.si/naloge-in-resitve/>. [Dostop: 15. november 2024].

[3] D. Žmavc, "Dva državna podprvaka iz programiranja v Scratchu," OŠ Angela Besednjaka Maribor, 27. marec 2024. [Online]. Dostopno na: <https://www.abesednjak.si/2024/03/27/dva-drzavna-podprvaka-iz-programiranja-v-scratchu/>. [Dostop: 25. januar 2025].

[4] D. Žmavc, "Dva državna prvaka in podprvak iz programiranja v Scratchu," OŠ Angela Besednjaka Maribor, 28. marec 2023. [Online]. Dostopno na: <https://www.abesednjak.si/2023/03/28/dva-drzavna-prvaka-in-podprvak-iz-programiranja-v-scratchu/>. [Dostop: 25. januar 2025].

[5] D. Žmavc, "Imamo državnega prvaka in podprvaka (programiranje – Scratch)," OŠ Angela Besednjaka Maribor, 24. marec 2022. [Online]. Dostopno na: <https://www.abesednjak.si/2022/03/24/imamo-drzavnega-prvaka-in-podprvaka-programiranje-scratch/>. [Dostop: 25. januar 2025].

[6] D. Žmavc, "Šolsko in državno tekmovanje iz programiranja – Scratch," OŠ Angela Besednjaka Maribor, 23. marec 2021. [Online]. Dostopno na: <https://www.abesednjak.si/2021/03/23/solsko-in-drzavno-tekmovanje-iz-programiranja-scratch/>. [Dostop: 25. januar 2025].

[7] Google. (2024). Gemini (različica iz 19. novembra). [Velik jezikovni model]. Dostopno na: <https://gemini.google.com/> (Kaj je Scratch?, Kaj je umetna inteligenca?, Tekmovanje iz Scratcha).

[8] Google. (2024) Gemini (različica iz 14. decembra). [Velik jezikovni model]. Dostopno na: <https://gemini.google.com/> (Kaj je Gemini? Kdaj je nastal? Kdo je avtor? Kje je na voljo? Katere različice poznamo? Kaj nam omogoča? Pomagaj mi rešiti spodnjo nalogo v Scratchu: Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje. Ali mi lahko narediš kar barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3?).

[9] Microsoft. (2025) Microsoft 365 Copilot (različica iz 18. januar). [Velik jezikovni model]. Dostopno na: <https://copilot.microsoft.com/> (Kaj je Copilot? Kdaj je nastal? Kdo je avtor? Kje je na voljo? Kaj nam omogoča? Pomagaj mi rešiti spodnjo nalogo v Scratchu: Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje. Ali mi lahko narediš kar barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3?).

[10] "O Scratchu," Scratch, [Online]. Dostopno na: <https://scratch.mit.edu/about> [Dostop: 3. februar 2025].

[11] OpenAI. (2024). ChatGPT (različica iz 26. decembra). [Velik jezikovni model]. Dostopno na: <https://ChatGPT.com/> (Kaj je ChatGPT?, Kdaj je nastal?, Kdo je avtor?, Katere različice poznamo?, Kaj nam omogoča? Pomagaj mi rešiti spodnjo nalogo v Scratchu: Dani vas pozove, da vpišete v vrstico poljubno zaporedje naravnih števil, ločenih z vejico. Dani bo števila prebral in med njimi izbral največje. Ali mi lahko narediš kar barvne bloke, kot je v Scratchu, ali celo datoteko formata .sb3?).

7.3 Viri slik

Slika 1: Logotip Scratcha in glavna figura, "Scratch Coding for Kids: A Beginner's Guide," Code Galaxy, 26. avgust 2024. [Online image]. Dostopno na: <https://thecodegalaxy.com/blog/scratch-coding-for-kids-a-beginners-guide/> [Dostop: 26. november 2024].

Slika 2: Grafični vmesnik Scratcha (lasten vir) – posnetek zaslona

Slika 3: Knjižnica vgrajenih figur (lasten vir) – posnetek zaslona

Slika 4: Potek dela na raziskovalnem taboru (lasten vir)

Slika 5: Naloga, ki sem jo izbral za moje raziskovanje, dr. Borut Jurčič Zlobec, Zveza za tehnično kulturo Slovenije, januar 2023. [Online image]. Dostopno na: https://zotks.si/wp-content/uploads/2023/06/solska_23_G.pdf [Dostop: 4. december 2024].

Slika 6: Fotografija kode (blokov), ki jo je predlagal ChatGPT (lasten vir) – posnetek zaslona

Slika 7: Fotografija kode (blokov), ki jo je bilo potrebno dodati (lasten vir) – posnetek zaslona

Slika 8: Fotografija kode (blokov), ki jo je predlagal Gemini (lasten vir) – posnetek zaslona

Slika 9: Fotografija kode (blokov), ki jo je bilo potrebno dodati (lasten vir) – posnetek zaslona

Slika 10: Fotografija kode (blokov), ki jo je predlagal Copilot (lasten vir) – posnetek zaslona

Slika 11: Fotografija kode (blokov), ki jo je bilo potrebno dodati (lasten vir) – posnetek zaslona

Slika 12: Koda (bloki) rešitve naloge, dr. Borut Jurčič Zlobec, Zveza za tehnično kulturo Slovenije, januar 2023. [Online image]. Dostopno na: <https://zotks.si/naloge-in-resitve/> [Dostop: 4. december 2024].

Slika 13: Merjenje časa generiranja odgovora UI s štoparico (lasten vir)