

ŠOLSKI CENTER VELENJE

Elektro in računalniška šola

Trg mladosti 3, 3320 Velenje

Mladi raziskovalci za razvoj Šaleške doline

Raziskovalna naloga

## **Algoritmi in grafično programiranje z Raptorjem**

(raziskava možnosti okolja Raptor za učenje programiranja)

Tematsko področje: Računalništvo

Avtorji: Lovro Suhovršnik, 2. letnik

Žan Pukmajster, 1. letnik

Aleksander Flis, 2. letnik

Mentorja:

Karmen Grabant

Nedeljko Grabant

Velenje, 2017

Raziskovalna naloga je bila opravljena na ŠC Velenje, Elektro in računalniški šoli, 2017.

Mentorja:

Mag. Karmen Grabant

Nedeljko Grabant, dipl. inž.

Datum predstavitve: marec 2017



By: Lovro Suhovršnik, Aleksander Flis, Žan Pukmajster, Nedeljko

Grabant, Karmen Grabant

## KLJUČNA DOKUMENTACIJSKA INFORMACIJA

ŠD	ŠC Velenje, šolsko leto 2016/2017
KG	Raptor   grafično programiranje   algoritem   diagram poteka
AV	Suhovršnik Lovro, Pukmajster Žan, Flis Aleksander
SA	GRABANT Karmen, GRABANT Nedeljko
KZ	3320 Velenje, SLO, Trg mladosti 3
ZA	ŠC Velenje, Elektro in računalniška šola, 2017
LI	2017
IN	Algoritmi in grafično programiranje z Raptorjem
TD	Raziskovalna naloga
OP	<i>X, 40 s., 9 tab., 50 sl., 0 p., 16 vir</i>
IJ	Slovenščina
JI	Slovenščina
AI	

V raziskovalni nalogi je opisan program Raptor, njegova praktična uporaba in grafično programiranje. Raptor je za začetek programiranja dokaj preprosto orodje za reševanje računalniških problemov, ki dovoli uporabniku, da ustvari izvršljive diagrame poteka (algoritme). Grafično se lahko spremlja potek in tudi vrednosti spremenljivk. Napisan je bil za študente ameriškega zrakoplovstva, ki se učijo »računalniških poklicev«, se pravi pravil, kako razviti sposobnosti reševanja problemov in učenja algoritmičnega razmišljanja. V računalništvu pod pojmom grafično programiranje ali vizualni programski jezik razumemo katerikoli programski jezik, ki uporabniku dovoljuje ustvarjanje programov z manipuliranjem grafičnih elementov programa namesto besedilnega pisanja kode. Na kratko smo predstavili tudi nekatere druge programe za grafično programiranje in primerjali s programom Visual Logic 2.7 ter ugotavljali razlike (prednosti in slabosti) med programoma. V raziskovalni nalogi so predstavljeni rezultati ankete in ugotovitve hipotez. To zanimivo programersko orodje je predstavljeno in dostopno na spletni strani <http://raptor.martincarlisle.com/>.

## KEY WORDS DOCUMENTATION

ND	ŠC Velenje, 2016/2017
CX	Raptor   graphic programming   algorithm   flowchart
AU	Suhovršnik Lovro, Pukmajster Žan, Flis Aleksander
AA	GRABANT Karmen, GRABANT Nedeljko
PP	3320 Velenje, SLO, Trg mladosti 3
PB	ŠC Velenje, Elektro in računalniška šola, 2017
PY	2017
TI	Algorithms and Graphic programming in Raptor
DT	RESEARCH WORK
NO	X, 40 p., 9. tab., 50 fig., 0 app., 16 ref.
LA	SL
AL	sl/en

In this research paper, we introduced the software Raptor, its practical use and graphic programming. Raptor is rather a simple tool for solving computing problems, which provides user with an option to create executable flowcharts (algorithms). The whole termination of the flowchart and values of variables can be seen through the course of the programme. Software was originally created for students of the US Air Force Academy, studying for the computer profession, namely the rules of how to develop ability to solve problems and learn about algorithmic thinking. In computing, the graphic programming concept refers to any programming language that provides the user with ability to create programmes by manipulating graphic elements instead of writing the code. We briefly introduced some other programmes for graphic programming, then compared them with the programme Visual Logic 2.7 and established the difference between them. In this research paper, the results of a survey and the verification of a hypothesis are presented as well. This interesting programming tool is represented and available on the following web page: <http://raptor.martincarlisle.com/>.

## KAZALO KRATIC

% – odstotek

€ – evro

3D – tridimenzionalno

ADL – Architecture Description Language (arhitekturno opisni jezik)

angl. – prevod iz angleškega jezika

dipl. – diplomirani

ERŠ – Elektro in računalniška šola

g. – gospod

ga. – gospa

HTML – angl. Hyper Text Markup Language (jezik za označevanje nadbесedila)

http – angl. hipertext transfer protcole (nadbесedilni prenosni protokol)

HTTP – Hyper Text Transfer Protocol (nadbесedilni prenosni protokol)

IDE – Integrated Development Enviroment (integrirano razvijalno okolje)

inž. – inženir

ipd. – in podobno

LARP – Logic of Algorithms and Resolution of Problems (algoritemska logika ter rešitev problemov)

npr. – na primer

OMG – Object Management Group (skupina strokovnjakov, ki skrbi za UML)

oz. – oziroma

t. i. – tako imenovano

UML – Poenoteni jezik modeliranja (angl. Unified Modelling Language)

VPL – Visual Programing Language (vizualni programski jezik)

## KAZALO VSEBINE

1	UVOD .....	1
1.1	Namen raziskovanja .....	1
1.2	Hipoteze.....	2
2	PREGLED OBJAV .....	3
2.1	Kaj je algoritem? .....	3
2.2	Kaj je grafično programiranje?.....	3
2.3	Krajši zgodovinski pregled grafičnega programiranja .....	4
2.3.1	Zgodovina grafičnega programiranja .....	4
2.3.2	Zgodovina izobraževalnih programskih jezikov .....	5
2.4	Opis najbolj razširjenih izobraževalnih programskih jezikov .....	7
2.4.1	Alice .....	7
2.4.2	Drakon .....	7
2.4.3	Microsoft Small Basic .....	9
2.4.4	Scratch.....	9
2.4.5	Visual Logic .....	10
2.4.6	LARP.....	10
3	MATERIALI IN METODE DELO.....	12
3.1	Prevod pomoči programa Raptor.....	12
3.1	Predstavitve programa Raptor .....	12
3.1.1	Simboli v Raptorju .....	14
3.2	Praktični primeri uporabe prevodov pomoči programa Raptor.....	15
3.2.1	Načini delovanja v Raptorju.....	15
3.2.2	Raptor referenca (angl. Reference).....	16
3.2.3	Izdelava učnega koncepta za učenje grafičnega programiranja s programom Raptor... 19	
3.2.2	Preizkus uporabe koncepta za učenje programiranja s programom Raptor .....	19
3.3	Anketa .....	20
4	RAZPRAVA .....	24
4.1.1	Primerjava programa Small Basic in programa Raptor .....	24
4.1.2	Primerjava programa LARP in programa Raptor .....	25
4.1	Težave pri učenju grafičnega programiranja .....	27
4.2	Rezultati ankete .....	27
4.3	Predlogi za izboljšanje učenja grafičnega programiranja s programom Raptor .....	33
5	ZAKLJUČEK.....	34
6	ZAHVALA.....	35
7	POVZETEK .....	36
8	VIRI.....	37

9	PRILOGA.....	39
10	AVTORJI RAZISKOVALNE NALOGE .....	40

### Kazalo slik

Slika 1: Primer diagrama poteka v programu Raptor, lasten vir .....	1
Slika 2: Diagram poteka za deljenje in uporabo ostanka deljenja z ukazom MOD ter REM .....	2
Slika 3: Programiranje v Sketchpadu, vir: [1] .....	4
Slika 4: Programski jezik ADL, vir: [2] .....	5
Slika 5: Programiranje v Pascalu, vir: [3] .....	5
Slika 6: Programski jezik Alice, vir: [4].....	6
Slika 7: Programski jezik Etoys, vir: [5] .....	6
Slika 8: Primer izgleda programa Alice, vir: [6] .....	7
Slika 9: Ikone v jeziku DRAKON, vir: [7].....	8
Slika 10: Primer ustvarjanja diagrama v jeziku DRAKON, vir: [8].....	8
Slika 11: Okno programa Small Basic ter primer kode, vir: [9].....	9
Slika 12: Primer programa Scratch, vir: [10] .....	10
Slika 13: Program Visual Logic, vir: [11].....	10
Slika 14: Primer algoritma v programu LARP, vir: [12].....	11
Slika 15: Okno o Raptorju.....	12
Slika 16: Možni izvozi iz Raptorja v načinu začetni (angl. Novice) in srednje zahtevno (angl. Intermediate) .....	12
Slika 17: Glavno okno programa Raptor ob zagonu programa, lasten vir .....	13
Slika 18: Primer algoritma v programu Raptor, lasten vir .....	14
Slika 19: Okno s simboli v Raptorju .....	14
Slika 20: Uporabnik lahko izbira med tremi načini uporabe programa Raptor.....	16
Slika 21: Okno UML z blokovnim simbolom za nov razred, nov vmesnik in novo naštevanje .....	16
Slika 22: Izgled 1. dela ankete.....	20
Slika 23: Primer 2. dela ankete.....	21
Slika 24: Primer 3. dela ankete.....	21
Slika 25: Primer 2. dela ankete.....	22
Slika 26: 5. del ankete .....	23
Slika 27: Zanje vprašanje ankete.....	23
Slika 28: Glavno okno Small Basic, lasten vir .....	24
Slika 29: Primer programa v Small Basicu, lasten vir.....	25
Slika 30: Glavno okno LARP za pisanje diagramov poteka, lasten vir .....	25
Slika 31: Okno LARP-a za pisanje kode, lasten vir .....	26
Slika 32: Primer sintakse v LARP-u, lasten vir.....	26
Slika 33: Primer diagrama poteka v LARP-u, lasten vir .....	27
Slika 34: Naslovna stran predstavitve, lasten vir.....	33
Slika 35: Predstavljanje simbola spremenljivke in uporabe le te v programu Raptor, lasten vir .....	33
Slika 36: Delujoča ura v Raptorju, lastna slika .....	34
Slika 37: Primer prevoda pomoči, lasten vir .....	39
Slika 38: Lovro Suhovršnik, Aleksander Flis, Žan Pukmajster (s leve na desno).....	40

### Kazalo grafov

Graf 1: Spol anketirancev, lastni graf.....	28
Graf 2: Starost anketirancev, lasten graf .....	28
Graf 3: Ali ste že kdaj naredili računalniški program?, lasten graf.....	29

Graf 4: Ali se želite naučiti izdelati računalniški program?, lasten graf .....	29
Graf 5: Kaj je Raptor v računalniškem svetu?, lasten graf.....	30
Graf 6: Kaj je diagram poteka, lastni graf .....	30
Graf 7: Kaj je programiranje?, lasten graf.....	30
Graf 8: Kaj je program v računalništvu?, lastni graf.....	31
Graf 9: Kaj je grafično programiranje? .....	31
Graf 10: Izberite področja uporabe grafičnega programiranja? .....	31
Graf 11: Kaj je računalniška konstanta?.....	32
Graf 12: Kaj je spremenljivka v računalniškem svetu? .....	32
Graf 13: Kaj je funkcija v računalništvu ali matematiki?.....	32

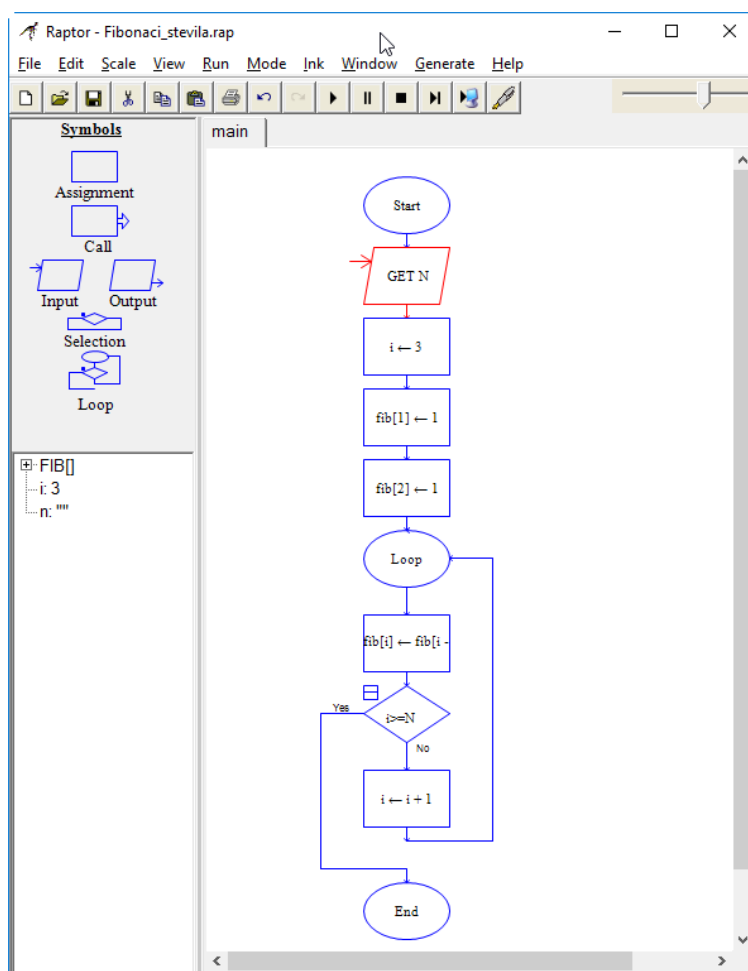
### **Kazalo tabel**

Tabela 1: Vrste izobraževalnih grafičnih programov za učenje programiranja.....	3
Tabela 2: Vrste multimedijskih grafičnih programov za učenje programiranja.....	4
Tabela 3: Matematični operatorji (angl. Math Operators).....	16
Tabela 4: Relacijski operatorji (Relational Operators).....	17



## 1 UVOD

Kratica RAPTOR je nastala iz angleških besed Rapid Algorithmic Prototyping Tool for Ordered Reasoning, predstavlja pa grafično orodje, ki so ga ustvarili Martin C. Carlisle, Terry Wilson, Jeff Humphries, Jason Moore, Tim Chamillard, Tom Rathbun in Tom Schorsch. Program vzdržuje nekdanja ameriška letalska akademija in univerza Carnegie Mellon (profesor dr. Martin Carlisle). Program omogoča študentom in učencem, da ustvarijo in izvršijo programe s pomočjo diagramov poteka (slika 1). Običajno ta program uporabljajo v šolah in akademijah za učenje programiranja. V raziskovalni nalogi Algoritmi in grafično programiranje z Raptorjem smo se ukvarjali z možnostmi, ki jih omogoča okolje Raptor za osnove učenja grafičnega programiranja (slika 1).



Slika 1: Primer diagrama poteka v programu Raptor, lasten vir

### 1.1 Namen raziskovanja

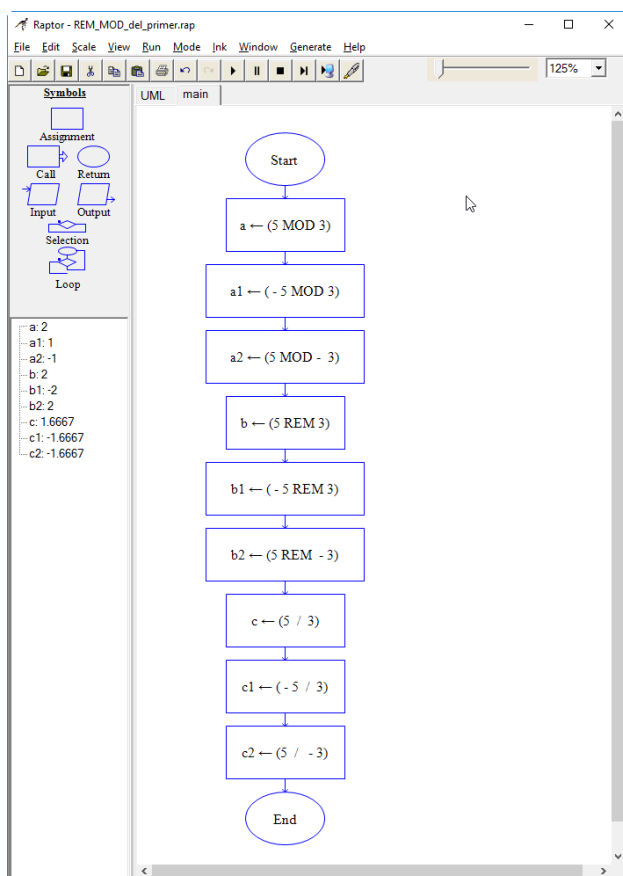
V raziskovalni nalogi je na kratko predstavljen program Raptor kot orodje za grafično programiranje, ustvarjanje algoritmov in načrtovanje programov. Prevedli smo angleško pomoč Raptorja, za kar upamo, da bo v veliko pomoč tistim, ki se začnejo ukvarjati s programiranjem v Raptorju. Pojasnjeni so tudi pojmi grafičnega programiranja v Raptorju in našteje najpogostejše metode in vrste ustvarjanja

algoritmov in programov v programu Raptor. Sicer v licenci programa piše: Ta programska oprema je na voljo "kot je" in brez ali implicitne garancije, vključno brez omejitev, naznačenih jamstev za prodajo ter ustreznosti za določen namen (v originalu THIS SOFTWARE IS PROVIDED "AS IS" AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.).

## 1.2 Hipoteze

Pred samim raziskovanjem smo si zastavili naslednje hipoteze.

1. Domnevamo, da več kot 50 % anketirancev ve, kaj so diagrami poteka.
2. Domnevamo, da več kot 30 % anketirancev ve, kaj je Raptor.
3. Predvidevamo, da je Raptor primerno grafično orodje za učenje osnov programiranja z diagrami poteka.
4. Predvidevamo, da je z Raptorjem mogoče ustvariti kompleksne diagrame poteka in razumljivo vizualizirati izvajanje le-teh.
5. Predvidevamo, da je z Raptorjem mogoče enostavno obogatiti pouk programiranja za osnovnošolce in srednješolce.



Slika 2: Diagram poteka za deljenje in uporabo ostanka deljenja z ukazom MOD ter REM

## 2 PREGLED OBJAV

Pri pregledu objav obravnavamo, kaj je grafično programiranje, krajši zgodovinski pregled le-tega, opis najbolj razširjenih izobraževalnih programskih jezikov.

### 2.1 Kaj je algoritem?

Naslednje besedilo je povzeto po viru [16].

Algorítem je navodilo, s katerim rešujemo nek problem. Običajno je zapisan kot seznam korakov, ki nas pripeljejo do rešitve problema. Kako podrobno razdelamo korake, je odvisno od tega, kdo izvaja algoritem (človek, računalnik). Če algoritem izvaja računalnik, potem govorimo o računalniškem programu. Primer algoritma iz vsakdanjega življenja je kuharski recept.

### 2.2 Kaj je grafično programiranje?

Naslednje besedilo je povzeto po viru [18].

V računalništvu pod pojmom grafično programiranje ali vizualni programski jezik – VPL (angl. Visual Programming Language) razumemo kateri koli programski jezik, ki uporabniku dovoljuje ustvarjanje programov z manipuliranjem grafičnih elementov programa namesto, pisanja kode s besedilom.

Definicija grafičnega programiranja

Na primer, večina VPL-jev temelji na ideji t. i. likov in puščic, ki te like povezujejo med seboj.

Vizualne programske jezike lahko podrobneje razvrstimo še glede na tip in obseg vizualnega izraza uporabljenega na jezikih, zasnovanih na:

- ikonah,
- oblikah,
- diagramih.

Vizualno programsko okolje zagotavlja grafične elemente ali ikone, ki jih lahko uporabniki uporabljajo na interaktivni način, glede na specifično določeno prostorsko slovnico za konstrukcijo programa.

Naslednja razpredelnica znanih vizualnih izobraževalnih programov, razvijalnih okolij (tabela 1).

*Tabela 1: Vrste izobraževalnih grafičnih programov za učenje programiranja*

Izobraževalni programi:		
Alice	App Inventor for Android	Snap!
Etoys	Flowgorithm	StarLogo
Hopscotch	Kodu	Visual Logic
LARP	Open Roberta	Stagecast Creator

RAPTOR	Scratch	ToonTalk in AgentSheets
--------	---------	----------------------------

Naslednja razpredelnica znanih vizualnih multimedijskih programov, razvijalnih okolij (tabela 2).

Tabela 2: Vrste multimedijskih grafičnih programov za učenje programiranja

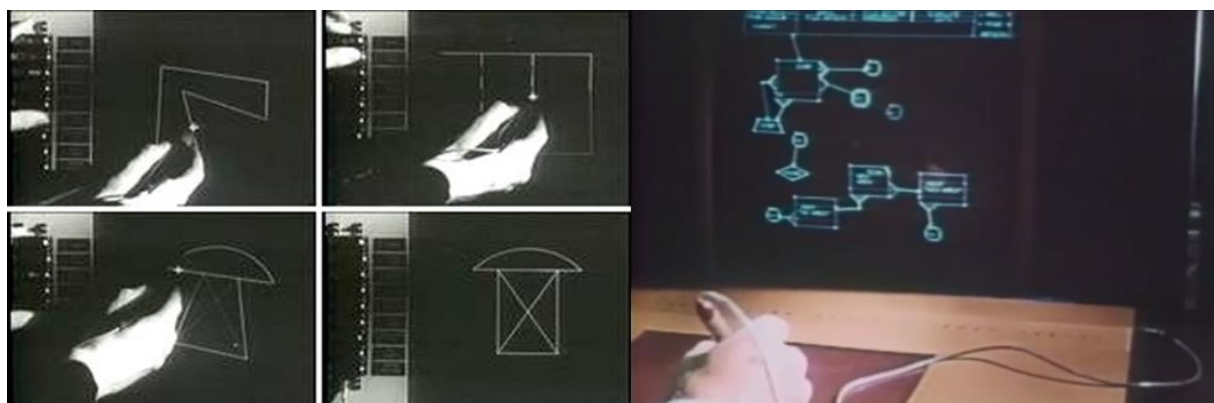
Multimedijski programi:		
Blender	Cameleon	Scala Multimedia
Clickteam's	Filter Forge	SynthEdit
Grasshopper 3d	Kyma	Virtools
Mama (software)	Max	VsXu
Max/MSP	Pure Data	Softimage
AudioMulch	Nuke	TouchDesigner
OpenMusic	Pure Data	WireFusion
Quartz Composer	Reaktor	Vvvv

## 2.3 Krajši zgodovinski pregled grafičnega programiranja

Predstavljena bo zgodovina grafičnega programiranja in izobraževalnih programskih jezikov.

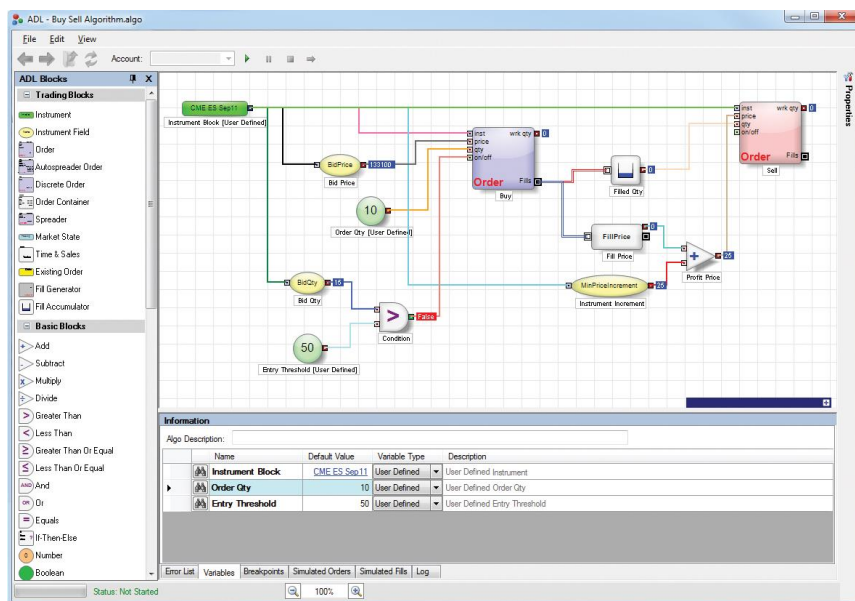
### 2.3.1 Zgodovina grafičnega programiranja

Eden izmed prvih grafičnih programskih jezikov je Sketchpad (slika 3). To je računalniški program, ki ga je leta 1963 napisal ameriški programer Ivan Sutherland. Novost programa je bila v tem, da je lahko uporabnik s pomočjo svetlobnega peresa risal po zaslonu in tako neposredno obdeloval predmete na njem, preden jih je shranil.



Slika 3: Programiranje v Sketchpadu, vir: [1]

ADL (angl. Architecture description language) (slika 4) je sodobnejši program v primerjavi s Sketchpadom, ter temelji na arhitekturnem programiranju. To pomeni, da je lažje berljiv tako človeku kot tudi računalniku. Program ADL nima standardnega programa, ker je več različic, npr. ADLS.



Slika 4: Programski jezik ADL, vir: [2]

### 2.3.2 Zgodovina izobraževalnih programskih jezikov

Eden izmed prvih ter najbolj znanih programskih jezikov je Pascal (slika 5), ki se je uporabljal od 1970 do 1980 kot glavni programski jezik za programiranje računalnikov ter učenje študentov. Njegova uporaba v aplikacijah se je v zadnjih letih zmanjšala, zato je obravnava jezika kot izobraževalnega postala nekoliko sporna.

```

File Edit Run Compile Options Debug Break/watch
Line 15 Col 39 Insert Indent Unindent * D:NONAME.PAS
program KenLovesTurboPascal;
uses
  crt;
var
  age: Integer;
  name: String;
  message: String;
begin
  clrscr;
  name := 'Ken Egozi';
  age := 30;
  if age < 10 then
    message := ' loves Turbo Pascal'
  else
    message := ' loved Turbo Pascal';
  write (name);
  writeln (message);
end.
Watch
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F10-Menu
    
```

Slika 5: Programiranje v Pascalu, vir: [3]

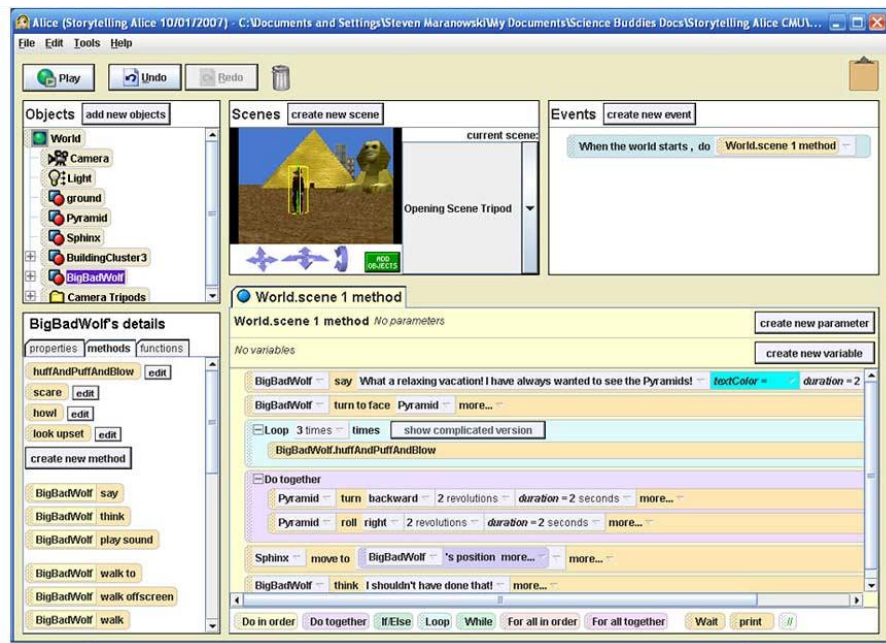
Ostali programski jeziki so še npr.:

- Alice
- Etoys

Te smo v nadaljevanju na kratko predstavili.

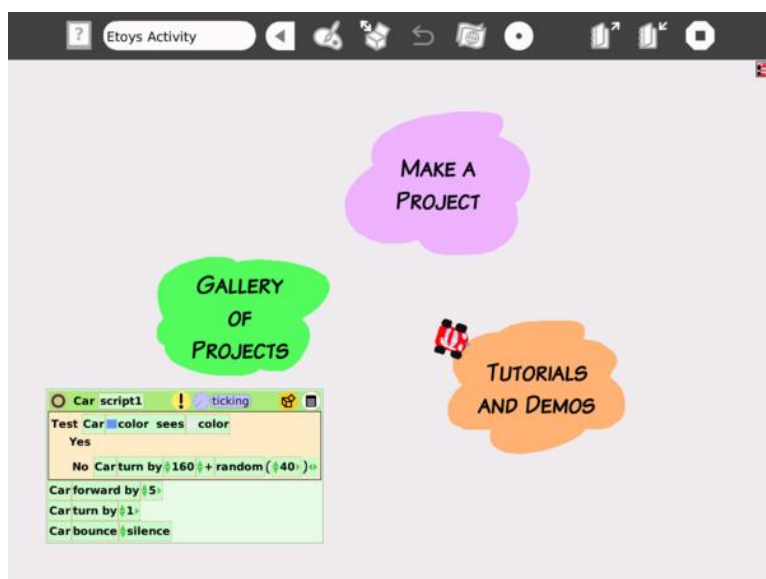
Alice (slika 6) je izobraževalni programski jezik, ki je bil razvit leta 1994 na univerzi v Virginiji.

Napisan je bil v programskem jeziku Java. Alice je odprtokodni program, namenjen za nekomercialne namene.



Slika 6: Programski jezik Alice, vir: [4]

Etoys (slika 7) je programski jezik, ki je bil razvit leta 1996. Namenjen je otrokom za učenje z vstavljanjem navideznih subjektov v program. Program zagotavlja medijsko bogate elemente v enostavnem okolju z objektnimi modeli, ki jih uporabnik ustvari. Vključuje 2D- in 3D-grafiko, slike, besedila, video posnetke, zvok ter MIDI. Program se je najbolj uporabljal v ZDA, Evropi, Južni Ameriki, na Japonskem, v Koreji ter tudi drugod.



Slika 7: Programski jezik Etoys, vir: [5]

## 2.4 Opis najbolj razširjenih izobraževalnih programskih jezikov

V nadaljevanju smo na kratko smo opisali programske jezike:

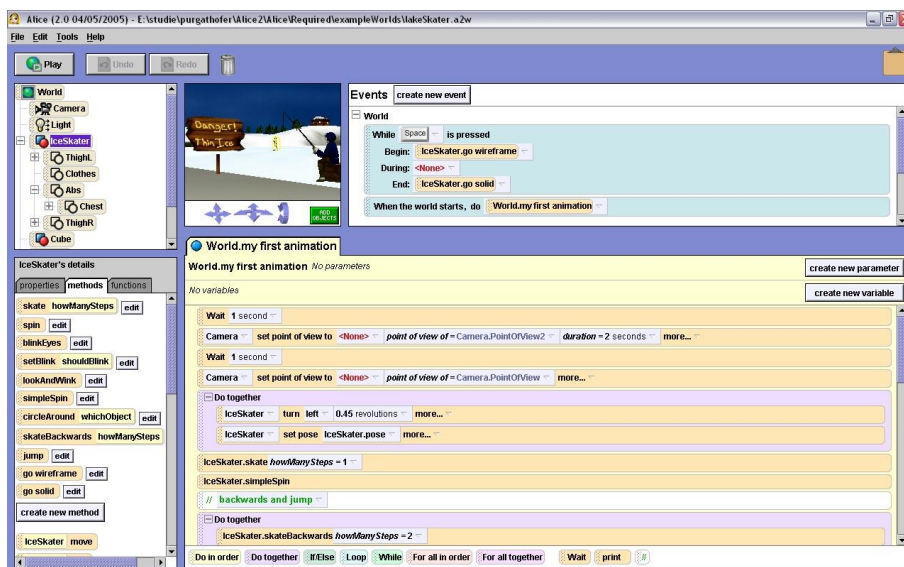
- Alice
- Drakon
- LARP
- Microsoft Small Basic
- Raptor
- Scratch
- Visual Logic 2.7

### 2.4.1 Alice

Naslednje je povzeto po viru [19]

Alice (slika 8) je inovativno 3D-programsko okolje, ki omogoča ustvarjanje animacij za pripovedovanje zgodbe, igranje interaktivnih iger, ali pa deljenje videov na internetu. Alice je brezplačno orodje za učenje, čigar namen je seznaniti učenca z objektnim programiranjem, ki je bilo napisano v Javi.

Trenutna najnovejša verzija programa Alice je Alice 3.3 (dostop: 31. 1. 2017). Alice temelji na ustvarjanju iger in video animacij (slika 8), dobra je za učenje, saj učencu omogoča, da program napiše pravilno ne glede na znanje. Če želi ustvariti program, mora povezati predloge, ki so mu ponujeni in ga tako ustvari.



Slika 8: Primer izgleda programa Alice, vir: [6]

### 2.4.2 Drakon

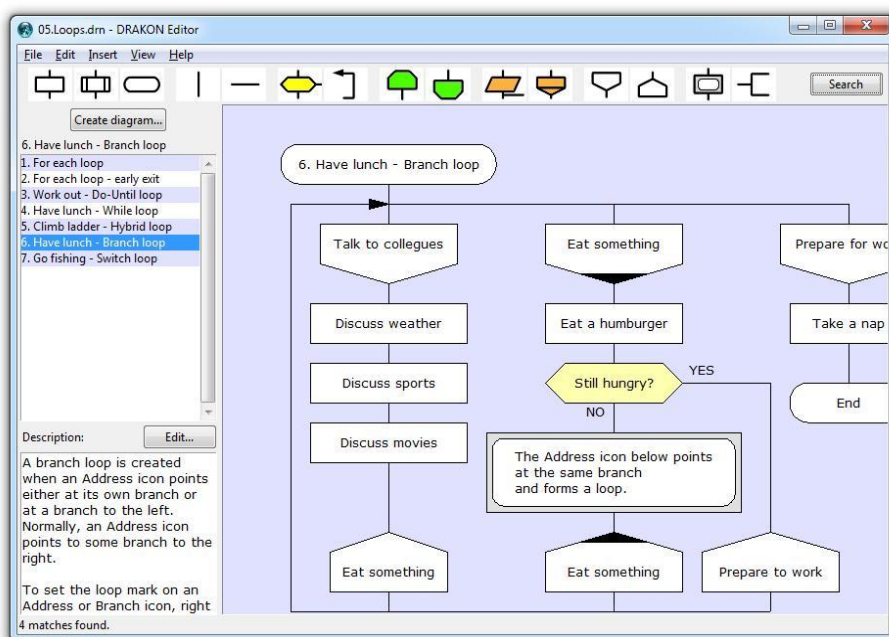
Povzeto po viru <http://drakon-editor.sourceforge.net/>.

DRAKON je algoritmično vizualni programski jezik (slika 9), ki je bil razvit za namene vesoljskega projekta Buran. Programski jezik je namenjen dokumentiranju algoritmov in programiranju. DRAKON Editor je napisan za operacijske sisteme Windows, OS X in Linux. Delovanje urejanja v DRAKON Editor temelji na prostem risanju. Uporabnik vnaša in upravlja s posebnimi ikonami (slika 9) in oblikami, kot so npr. pravokotnik, navpične in vodoravne črte ipd.

Icon	Name of Icon	Icon	Name of Icon	Macroicon	Name of Macroicon	Macroicon	Name of Macroicon
1	Title	14	Output	1	Title with parameters	11	Switch by timer
2	End	15	Input	2	Fork	12	SIMPLE loop by timer
3	Action	16	Pause	3	Switch (number of cases N >= 2)	13	SWITCH loop by timer
4	Question	17	Period	4	SIMPLE loop	14	FOR loop by timer
5	Choice	18	Start timer	5	SWITCH loop	15	WAIT loop by timer
6	Case	19	Synchronizer	6	FOR loop	16	Insertion by timer
7	Headline	20	Realtime parallel process	7	WAIT loop	17	Output by timer
8	Address	21	Comment	8	Action by timer	18	Input by timer
9	Insertion	22	Right comment	9	Shelf by timer	19	Start timer by timer
10	Shelf	23	Left comment	10	Fork by timer	20	Parallel process by timer
11	Formal parameters	24	Loop arrow			21	TREE loop
12	Begin of FOR loop	25	Silhouette arrow				
13	End of FOR loop	26	Connector				
		27	Concurrent process				

Slika 9: Ikone v jeziku DRAKON, vir: [7]

Zadnja posodobitev je bila 10. marca 2016 (DRAKON Editor 1.27), kar pomeni, da se program posodablja. Opis programa DRAKON ter posodobitev sta na spletni strani: <http://drakon-editor.sourceforge.net/>.



Slika 10: Primer ustvarjanja diagrama v jeziku DRAKON, vir: [8]



### 2.4.3 Microsoft Small Basic

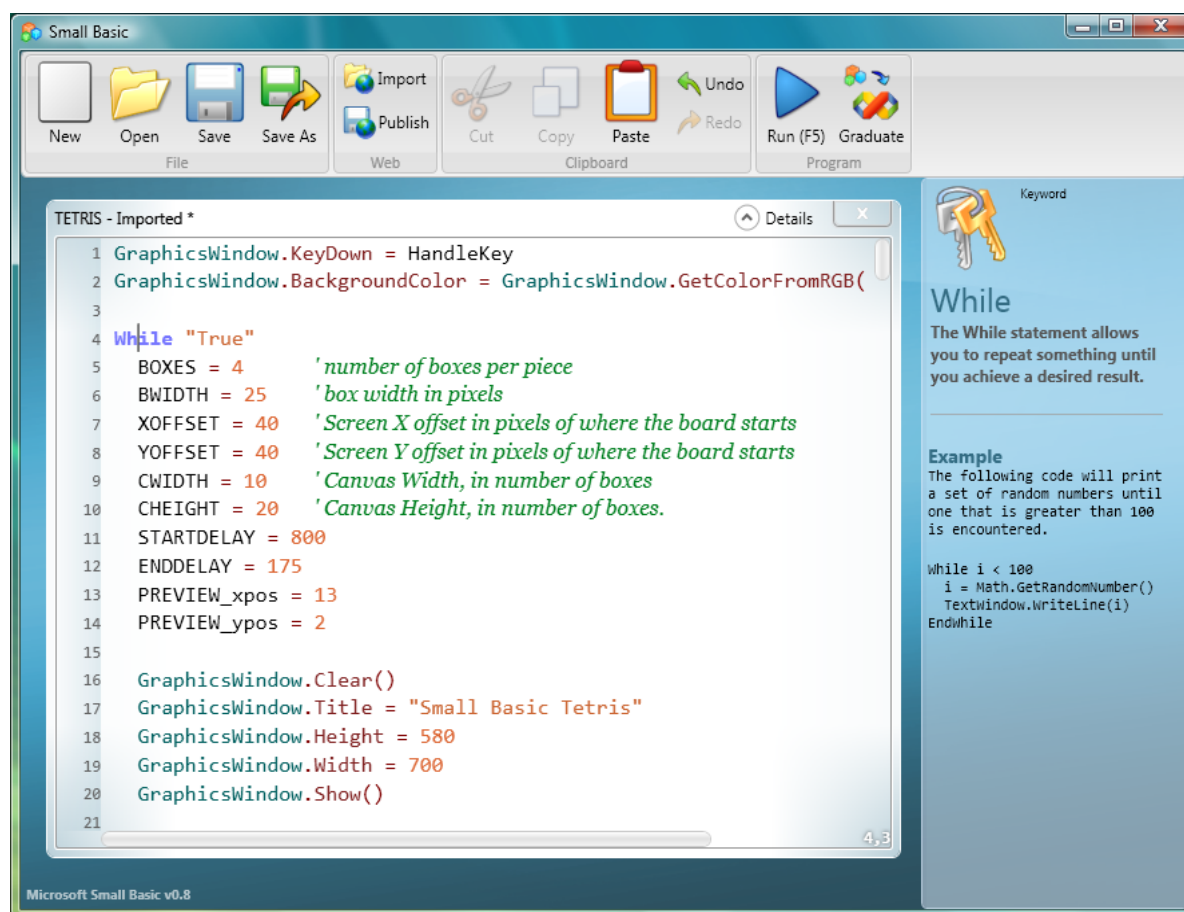
Povzeto po viru [20].

Spletišče Microsoft Small Basic: <http://smallbasic.com/>.

Small Basic je združitev programskega jezika in integriranega razvojnega okolja (IDE – Integrated development environment). Small Basic (slika 11) je poenostavljena različica programskega jezika BASIC, ki je namenjen začetnikom programiranja. Small Basic je izšel oktobra 2008. Prva stabilna različica je izšla 12. julija 2011. Nazadnje so program posodobili 1. oktobra 2015, na različico 1.2.Small Basic, ki podpira pogoje, zanke, več podatkovnih tipov (npr. nize, cela in realna števila), Microsoft Kinect ter t. i. željvo grafiko (angl. Turtle graphics).

Primer: Izpis »Hello world«:

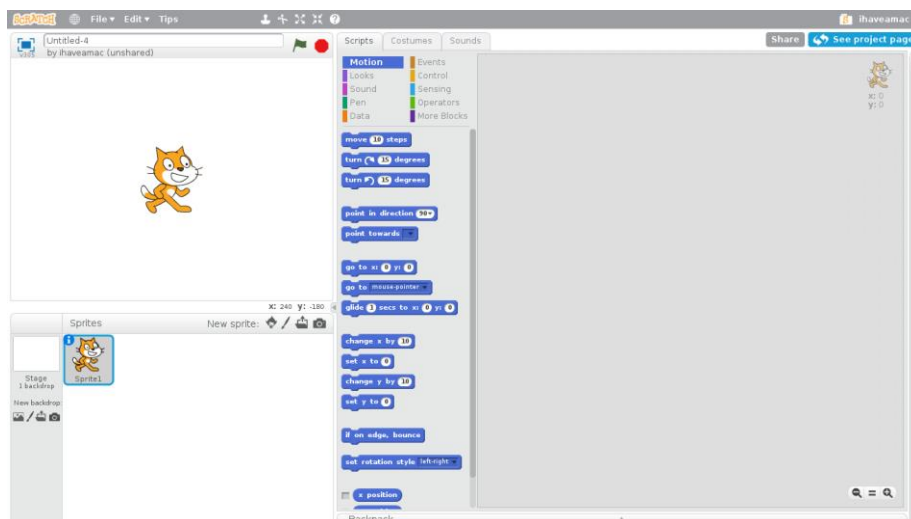
```
TextWindow.WriteLine("Hello World!")
```



Slika 11: Okno programa Small Basic ter primer kode, vir: [9]

### 2.4.4 Scratch

Scratch (slika 12) je brezplačen vizualni programski jezik, ki ga je razvil MIT Media Lab. Scratch uporabljajo študenti, učenci, učitelji, za ustvarjanje animacij, iger ipd. Scratch deluje podobno kot Alice, pri obeh programih ima učenec napisane predloge, ki jih uporabi za ustvarjanje programa (sestavlja jih skupaj).

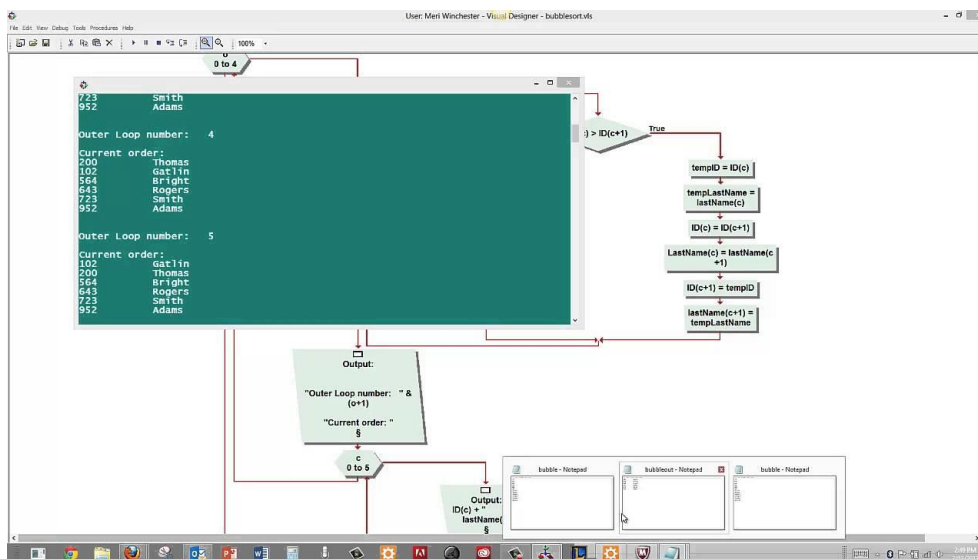


Slika 12: Primer programa Scratch, vir: [10]

## 2.4.5 Visual Logic

Visual Logic (slika 13) je grafično orodje, ki omogoča študentom pisanje in izvajanje programov s pomočjo diagramov poteka. Programski jezik se uporablja za učenje študentov o osnovah programiranja. Visual Logic se uporablja z vnašanjem različnih znakov, ki skupaj tvorijo program. Najnovejša posodobitev je Visual Logic 2.2.10 / 2014.

Spletna stran Visual Logic: <http://www.visuallogic.org/>.



Slika 13: Program Visual Logic, vir: [11]

## 2.4.6 LARP

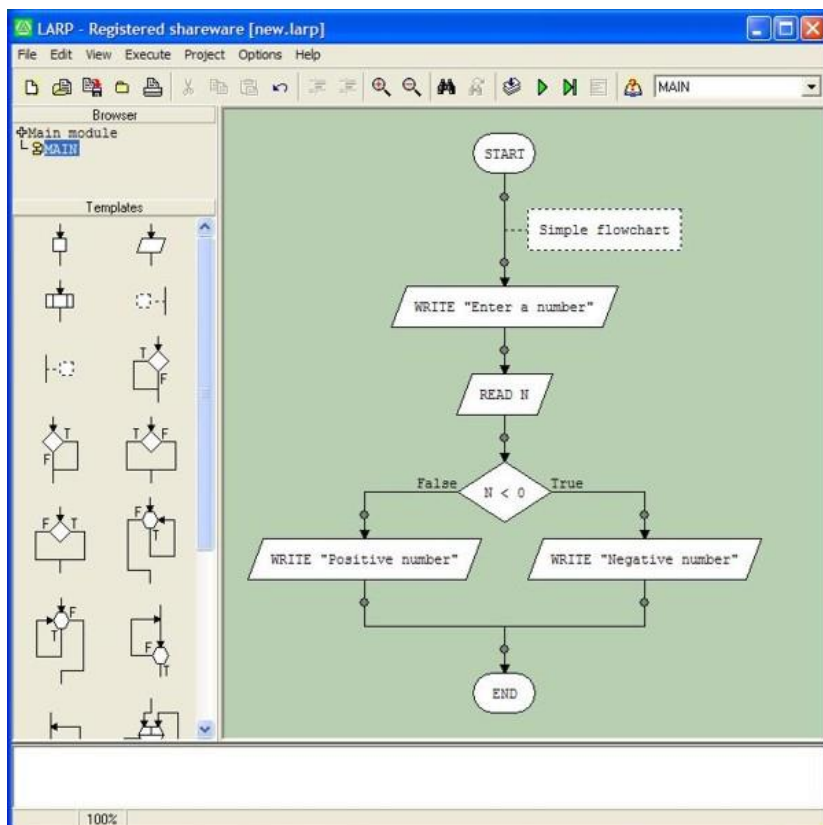
Spletišče: <http://larp.marcolavoie.ca/en/default.htm>

Kratka za Logics of Algorithms and Resolution of Problems (logika algoritmov in rešitev problemov).

LARP (slika 14) je programski jezik, ustvarjen z namenom hitrega načrtovanja preprostih algoritmov. Ustvaril ga je učitelj računalniškega programiranja Marco Lavoie. Njegova preprosta sintaksa izredno olajša razumevanje programiranja za začetnike. LARP podpira tudi vizualno/grafično programiranje, ki uporabnikom omogoča ustvarjanje algoritmov v obliki diagramov. Nazadnje je bil posodobljen leta 2008, kar pomeni, da je program zastarel, ampak še vedno dela in je še uporaben.

Njegova popolna dokumentacija se najde na naslednjem spletišču:

<http://larp.marcolavoie.ca/en/DocHTML/index.htm>



Slika 14: Primer algoritma v programu LARP, vir: [12]

### 3 MATERIALI IN METODE DE LA

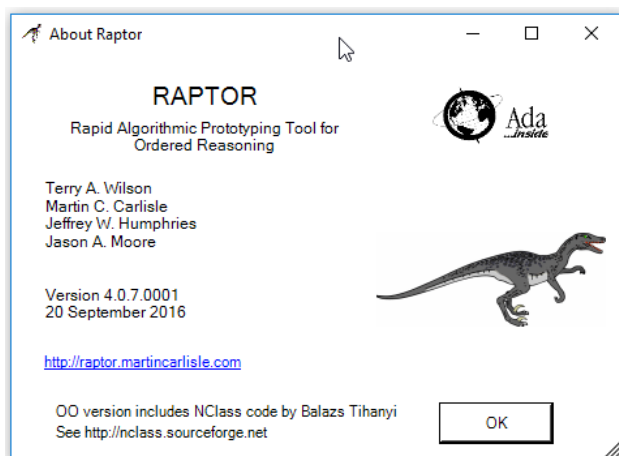
V tem poglavju so na kratko opisani načini dela in materiali, ki smo jih uporabljali za delo.

#### 3.1 Prevod pomoči programa Raptor

Najprej smo iz angleščine v slovenščino prevedli pomoč, ki je vključena v program Raptor. Delo smo si razdelili, da je potekalo hitreje. Prevode smo poskusili vključiti v program, oz. program v celoti prevesti. Prevodi so dodani v prilogi.

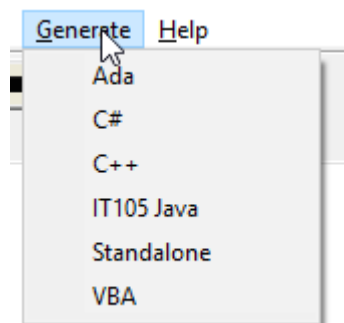
#### 3.1 Predstavitev programa Raptor

Raptor (slika 17) je brezplačen program za Windows, namenjen za programiranje/oblikovanje diagramov poteka ter učenje logike programiranja. Trenutno je aktualna verzija Raptorja 4.0.7.0001 (slika 15). Razvoj Raptorja ni najhitrejši, vsaj nazadnje je bil posodobljen 1. 10. 2016. Dostopen je s spletnega mesta: <http://raptor.martincarlisle.com/>.



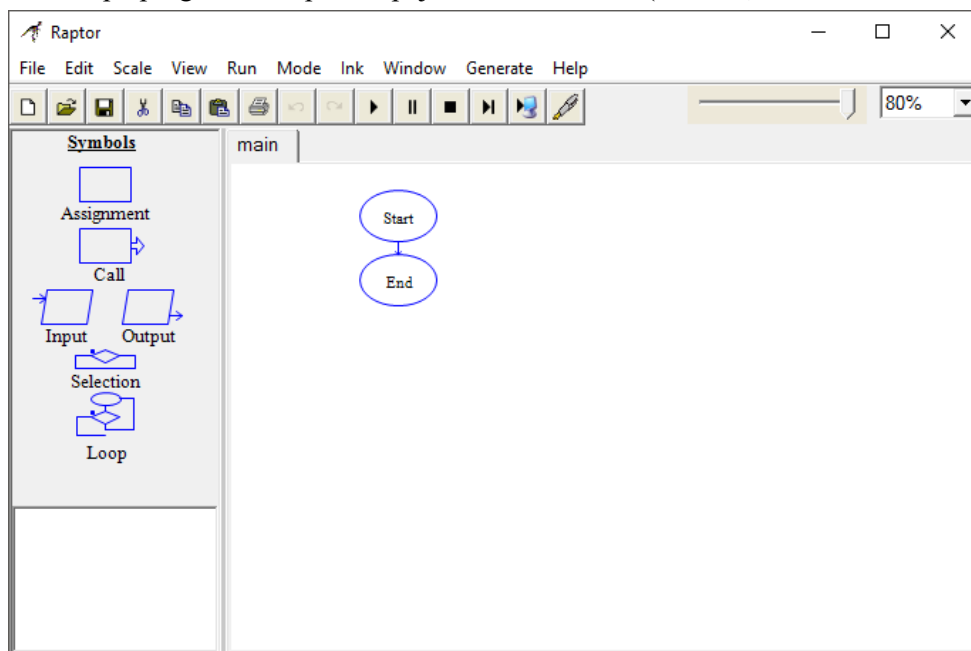
Slika 15: Okno o Raptorju

Diagrame sestavljamo z uporabo ikon, ki imajo različne funkcije. Raptor podpira osnovne podatkovne tipe: nize, tabele, števila ter realna števila. Raptor tudi podpira tabele ali polja (angl. Arraye), delo z datotekami, predvajanje zvočnih datotek in grafično risanje. Končne programe lahko izvozimo v .exe datoteke. Zraven je nameščena tudi zelo dobro obdelana dokumentacija.



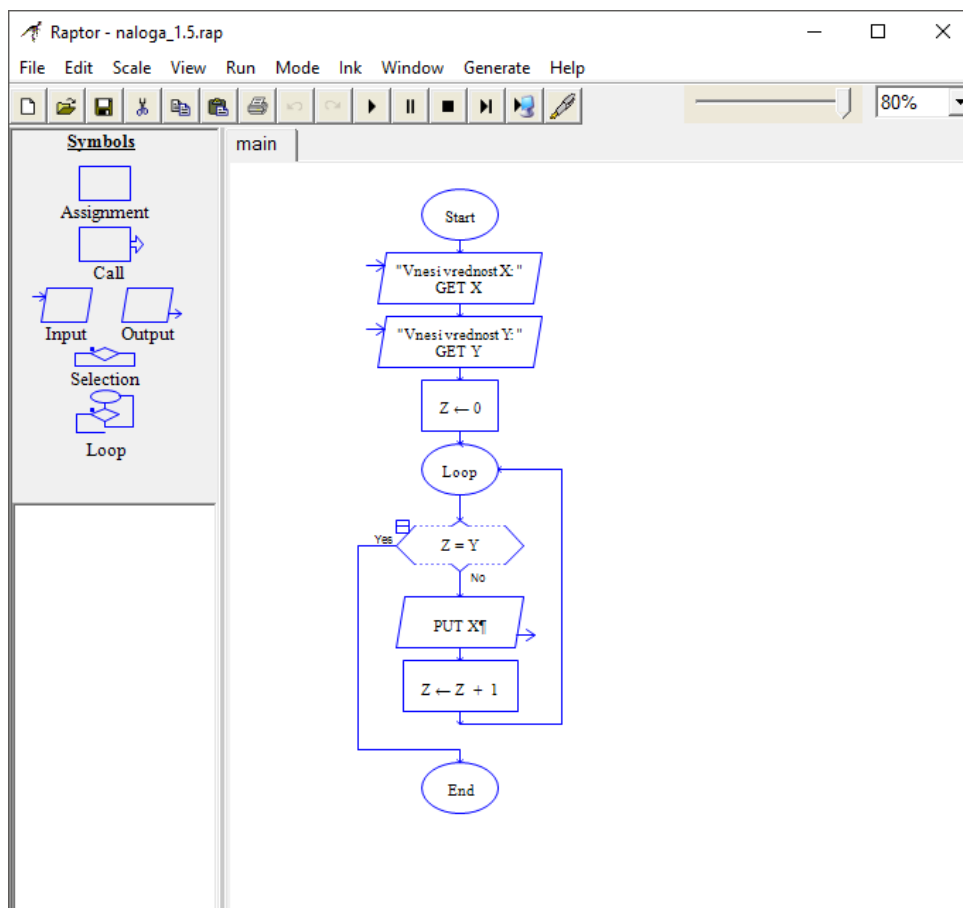
Slika 16: Možni izvozi iz Raptorja v načinu začetni (angl. Novice) in srednje zahtevno (angl. Intermediate)

Ob vklopu programa Raptor se pojavi značilno okno (slika 17).



Slika 17: Glavno okno programa Raptor ob zagonu programa, lasten vir

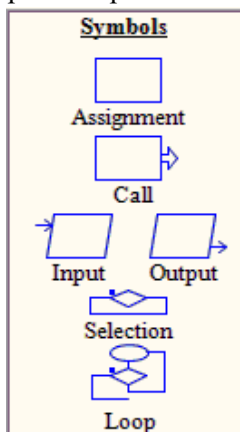
Primer preprostega algoritma v obliki diagrama poteka v programu Raptor je viden na naslednji sliki (slika 18). V tem diagramu poteka uporabnik vnese dve deli števili,  $x$  in  $y$ . Program potem izpiše vsa cela števila med mejami  $x$  in  $y$ .



Slika 18: Primer algoritma v programu Raptor, lasten vir

### 3.1.1 Simboli v Raptorju

Šesti simboli, ki se uporabljajo v Raptorju, so prikazani v oknu s simboli (Symbol Window) v zgornjem levem kotu glavnega okna (slika 19). Ti so v nadaljevanju na kratko opisani iz našega prevoda pomoči.



Slika 19: Okno s simboli v Raptorju

#### 3.1.1.1 Prireditvev (Assignment)

Simbol za določitev se uporablja za spreminjanje, določanje vrednosti spremenljivke. Desna stran spremenljivke poteče in rezultat je vrednost, ki se shrani na levo stran. Na primer, vrednost spremenljivke  $x$  je trenutno 5 in prireditveni stavek se glasi  $x \leftarrow x + 1$ . Najprej je ovrednotena vrednost  $x + 1$ , ki da v tem primeru rezultat 6, nato se ta vrednost shrani v spremenljivko  $x$ . Se pravi  $x$

ni več 5 ampak 6. POMEMBNO! Prireditvev je zelo drugačna od matematične enakosti. V tem primeru bi se ta določitev prebrala takole : »*V spremenljivko x shrani x + 1.*« ali »*V spremenljivko x shrani vsoto x in 1.*«, po angleško: »*Set x to x + 1*«.

### 3.1.1.2 KLIC (Call)

Simbol klic se uporablja za klicanje procedur. kot so grafične rutine in druge podobne procedure. Simbol klic se uporablja tudi za izvajanje poddiagramov (podprogramov), ki so vključeni v Raptor program.

### 3.1.1.3 VNOS (Input)

Simbol za vnos se uporablja, da od uporabnika pridobimo število ali niz medtem, ko se diagram poteka izvaja. Ko se simbol izvaja, bo uporabnika pozval k vnosu vrednosti, ki je lahko ali število ali niz, odvisno od uporabnika, kaj vnese. Uporabnik lahko določi vir vnosa z določanjem text datoteke in pozicije na tipkovnici (glej [Vnos iz datoteke](#)).

### 3.1.1.4 IZPIS ali izhod (Output)

Simbol za izpis se uporablja za izpis številke ali besedila v konzoli. Uporabnik lahko nastavi mesto izpisa z določanjem besedilne (angl. Text) datoteke, ki jo naj izpiše simbol (glej [Izpis iz datoteke – v Prilogi 1 Pomoči](#)).

### 3.1.1.5 POGOJ ali ODLOČITEV (Selection)

Struktura pogoja se uporablja za izvajanje odločitev. Programer v romb vnese izraz, ki je lahko ovrednoten samo z **DA** (Yes, True) in **NE** (No, False). Takšni izrazi se v osnovi nanašajo na Boolovo algebro. Na osnovi rezultata izraza bo romb določil, ali se bo diagram poteka nadaljeval po levi ali po desni strani. Levo (DA, True) ali desno (NE, False). Uporabnik lahko zamenja funkcije levo in desno, npr. Levo (NE, False) in desno (DA, True). (Za več informacij glej [Boolovi izrazi – v Prilogi 1 Pomoči](#)).

### 3.1.1.6 ZANKA (Loop)

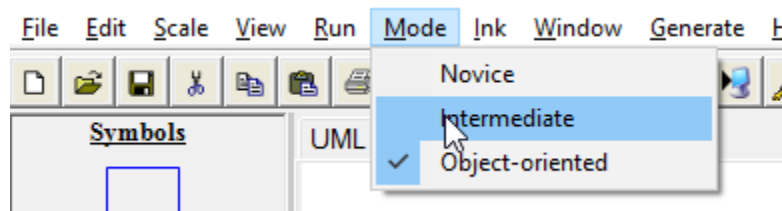
Zanka se uporablja, za ponavljanje izraza, doklej ni zadoščeno pogoju, ki smo ga določili. Ko potek v zanki pride do konca (dna), se začne ponovno na vrhu. Izhod iz zanke se zgodi, ko je pogoju zadoščeno.

## 3.2 Praktični primeri uporabe prevodov pomoči programa Raptor

Prevode smo uporabljali pri projektne tednu. Nameravamo ustvariti HTML Help datoteko, ki vsebuje prevode in je med sabo povezana. Če bo možno, jo bomo v kratkem poslali razvijalcu, da jo vključi v programsko kodo in bo namestitev pri nalaganju programa možna.

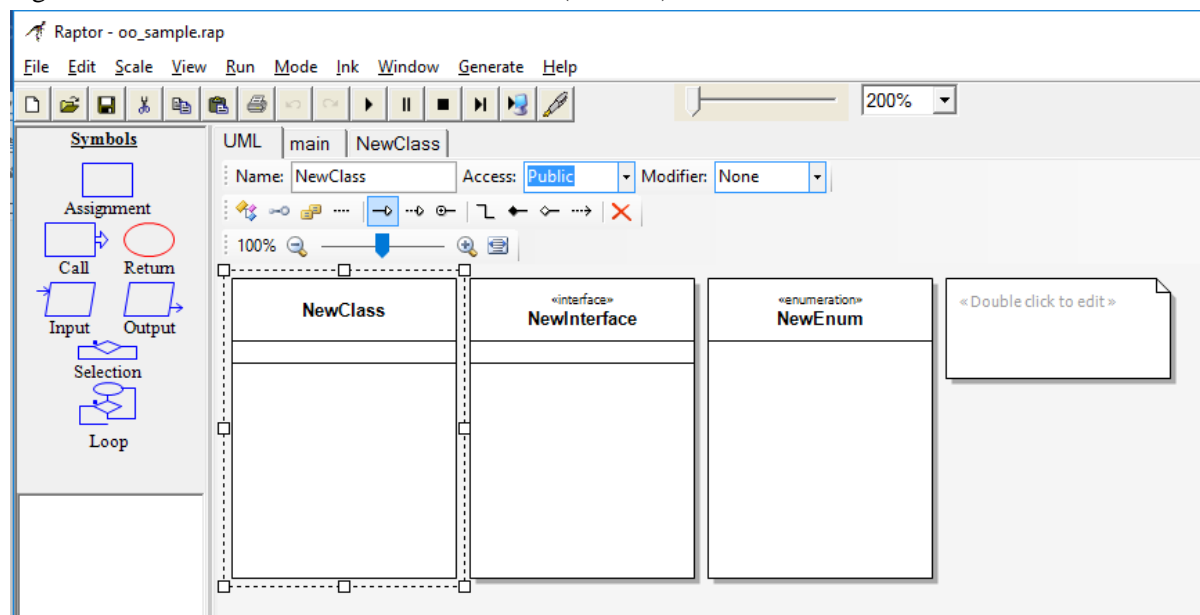
### 3.2.1 Načini delovanja v Raptorju

V Raptorju imamo, gledamo na znanje uporabnika, tri osnovne načine (angl. Mode) in ti so: začetni (angl. Novice), srednji (angl. Intermediate) in objektno orientiran (angl. Object oriented – slika 20).



Slika 20: Uporabnik lahko izbira med tremi načini uporabe programa Raptor

V objektno orientiranem načinu imamo pri simbolih dodatno ikono vrnitev ali angl. Return. Ta način je najbolj zahteven in sodoben. V tem načinu je mogoče izbirati takoj pod prodno vrstico levo zgoraj v glavnem delovnem oknu med Main in UML (slika 21).



Slika 21: Okno UML z blokvnim simbolom za nov razred, nov vmesnik in novo naštevanje

UML (angl. Unified Modeling Language) pomeni programski jezik za modeliranje v širšem pomenu. [23]

To je po prevodu gledano enoten jezik za modeliranje – skratka, je grafični jezik za modeliranje, določanje, gradnjo in dokumentiranje programskih komponent in drugih sistemov. To je razvila skupina [Object Management Group](#) (OMG) in je standardiziran z ISO (ISO /IEC 19505 za različico 2.4.1 [23]). UML definira identifikatorje za najbolj pomembne pogoje modeliranja in opredeljuje možne povezave med temi pogoji. Definira dodatne grafične notacije teh konceptov in modelov za statične strukture in dinamične procese, ki jih lahko formuliramo s temi pogoji. Danes je prevladujoči jezik za modeliranje kompleksnih sistemov programske opreme.

### 3.2.2 Raptor referenca (angl. Reference)

Sledi referenca vseh ukazov v Raptorju, da bralec dobi občutek o moči tega vizualnega orodja.

#### 3.2.2.1 Matematika

Tabela 3: Matematični operatorji (angl. Math Operators)



Oznaka	Pomen
-	minus
$\wedge$ , **	potenciranje
*	množenje
/	deljenje
rem, mod	ostanke deljenja po modulu (celoštevilčni ostanek)
+	seštevanje
-	odštevanje

### Osnovne matematične funkcije (angl. **Basic Math Functions**)

**abs** – absolutna vrednost

**ceiling** – zaokroževanje (navzgor)

**e** – vrne vrednost e (osnoven za naravni logaritem)

**floor** – zaokroževanje (navzdol)

**log** – naravni logaritem

**max** – maksimalna od dveh vrednosti

**min** – minimalna od dveh vrednosti

**pi** – vrne vrednost pi ( $\pi$ )

**powermod** – uporaba pri kriptiranju/dekriptiranju

**random** – vrne naključno vrednost v območju [0.0,1.0]

**sqrt** – kvadratni koren

#### 3.2.2.2 *Kotne funkcije (angl. Trigonometric Functions):*

**arccos** – arkus cosinus

**arccot** – arkus kotagens

**arcsin** – arkus sinus

**arctan** – arkus tangens

**cos** – kosinus

**cot** – kotagens

**sin** – sinus

**tan** – tangens

#### 3.2.2.3 *Nadzor programa (angl. Program Control)*

Tabela 4: Relacijski operatorji (Relational Operators)

=	Je enako le v
>	Večje
<	Manjše

!=	Ni enako
/=	Ni enako
>=	Večje ali enako
<=	Manjše ali enako

### **Boolovi operatorji (angl. Boolean Operators)**

**and** – logična IN funkcija (velja, le v primeru, da sta oba izraza resnična, sicer je neresnično)

**not** – logični NE (negira izraz, resničen postane neresničen in obratno)

**or** – logični ali (je resnična izjava v primeru, da je vsaj ena ali obe vrednosti pravilni, sicer je izraz napačen angl. false)

**xor** – logični 'xor' (je resnična izjava v primeru, da sta obe vrednosti različni)

**clear\_console** – počisti konzolo

**delay\_for** – zakasnitev za vrednost x v sekundah

#### *3.2.2.4 Raptor grafika (angl. Raptor Graphics)*

### **Barve, ki so na voljo (Available colors)**

Črna, modra, zelena, cian, rdeča, magentna, rjava, svetlo siva, temno siva, svetlo modra, svetlo zelena, svetli cian, svetlo rdeča, svetlo magentna, rumena, bela.

Black, Blue, Green, Cyan, Red, Magenta, Brown, Light\_Gray, Dark\_Gray, Light\_Blue, Light\_Green, Light\_Cyan, Light\_Red, Light\_Magenta, Yellow, White

### **Možnosti polnila (Available fill values)**

- Filled, Unfilled (Zapolnjeno, prazno)
- True, False (Res, Ni res)
- Yes, No (Da, Ne)

### **Možne vrednosti klika miške (angl. Mouse button values)**

Levi\_klik (and. Left\_Button), Desni\_klik(ang. Right\_Button).

### **Grafične funkcije Raptorja (RaptorGraph Functions (return a value))**

Closest\_Color, Get\_Font\_Height, Get\_Font\_Width, Get\_Key, Get\_Key\_String, Get\_Max\_Height, Get\_Max\_Width, Get\_Mouse\_X, Get\_Mouse\_Y, Get\_Pixel, Get\_Window\_Height,

Get\_Window\_Width, Is\_Open, Key\_Down, Key\_Hit,

Mouse\_Button\_Down, Mouse\_Button\_Pressed, Mouse\_Button\_Released, Random\_Color,

Random\_Extended\_Color

### **Klici grafičnih funkcije Raptorja (angl. RaptorGraph Calls)**

Clear\_Window, Close\_Graph\_Window, Display\_Text, Display\_Number, Draw\_Arc, Draw\_Box,

Draw\_Bitmap, Draw\_Circle, Draw\_Ellipse, Draw\_Ellipse\_Rotate, Draw\_Line, Flood\_Fill,

Freeze\_Graph\_Window, Load\_Bitmap, Open\_Graph\_Window, Wait\_For\_Key,

Wait\_For\_Mouse\_Button, Get\_Mouse\_Button, Put\_Pixel, Set\_Font\_Size, Set\_Window\_Title,  
Unfreeze\_Graph\_Window, Update\_Graph\_Window

#### 3.2.2.5 Tabela ali polje (angl. Array)

##### **Operatorji tabele** (angl. **Array Operations**)

**Length\_Of** – vrne dolžino enodimenzionalnega tabele (polja)

[] – indeksiranje, izbere 1 element v tabeli (polju)

**To\_Character** – vrne znak iz vnesene ASCII-vrednosti

**To\_ASCII** – vrne numerično vrednost ASCII-kode za vneseni znak

#### 3.2.2.6 Nizi (angl. Strings)

##### **Operatorji nad nizi:**

+ – Združevanje

**Length\_Of** – vrne dolžino niza

[] – indeksiranje: izbere en znak iz niza (referenca na ASCII kodo)

#### 3.2.2.7 Poizvedba o tipu spremenljivke (angl. Variable Type Queries)

##### **Boolovi operatorji** (angl. **Boolean Functions**)

Is\_Array, Is\_Number, Is\_String, Is\_2D\_Array

#### 3.2.2.8 Zvok (Sound):

##### **Klici (funkcija za predvajanje) zvoka** (angl. **Sound Calls**)

Play\_Sound, Play\_Sound\_Background, Play\_Sound\_Background\_Loop

#### 3.2.2.9 Datoteke (angl. Files)

##### **Klici (funkcije za prikazovanje) datotek** (angl. **File Calls**)

**Redirect\_Input** – začne ali preneha uporabljati datoteko kot vir vnosa podatkov.

**Redirect\_Output** – začne ali preneha uporabljati datoteko kot vir izhoda podatkov.

### **3.2.3 Izdelava učnega koncepta za učenje grafičnega programiranja s programom Raptor**

Za potrebe projektnega tedna smo predhodno sestavili naloge in pripravili MS-PowerPoint predstavitev, v katerih smo posebej predstavili in razložili posamezno funkcijo, simbol in uporabo letih. Npr. razložili smo, kaj je prireditvev v računalniškem svetu, natančneje v grafičnem programiranju, in uporaba prireditvev v programu Raptor. Vsak simbol in funkcijo smo najprej predstavili posebej in jo kasneje utemeljili na praktičnih primerih.

#### **3.2.2 Preizkus uporabe koncepta za učenje programiranja s programom Raptor**

Pri projektne tednu smo dijake učili po naslednjem postopku

1. Učencem smo na PowerPoint predstavitvi predstavili temo, ki jo bomo obravnavali.

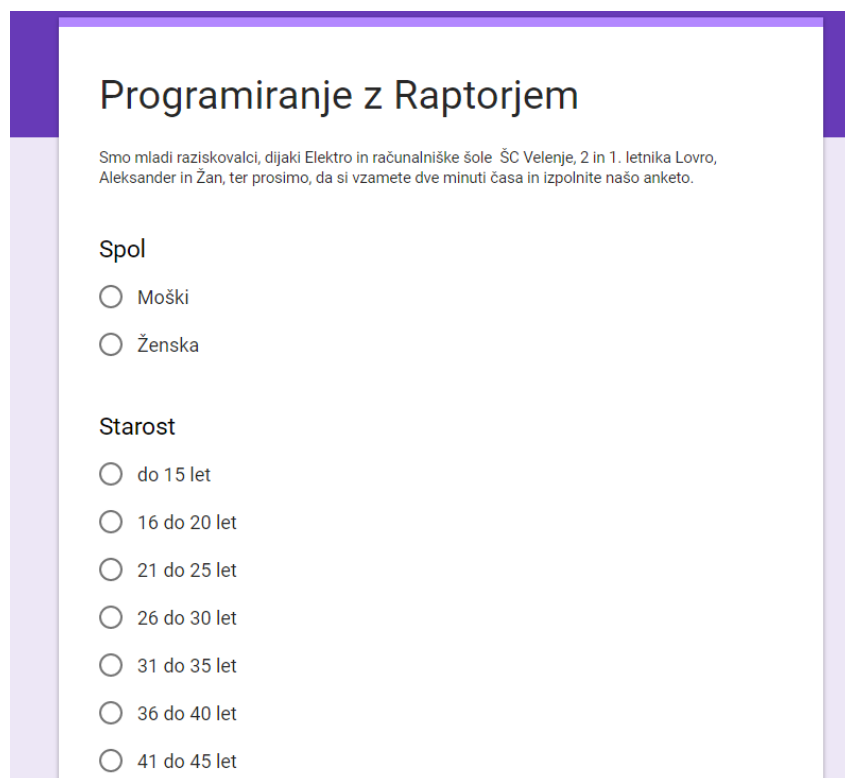
2. Pokazali smo jim primer na predstavljeno temo.
3. Skupaj z dijaki smo prebrali navodilo naloge.
4. Ko je večina dijakov naredila nalogo, smo skupaj preverili in rešili nalogo.

Večinoma smo med projektним tednom uporabljali postopek, opisan zgoraj. Preizkusili smo tudi postopek, pri katerem smo dijakom namesto besedila naloge pokazali diagram poteka, da so ga prepisali, vendar so nato, ko so morali sami napisati program, imeli veliko več problemov, saj je mogoče isti program napisati na več načinov. Ker pa smo jim dali grafični zgled, so vsi naredili tako, kot prikazoval zgled.

### 3.3 Anketa

Da bi potrdili zastavljene hipoteze, smo ustvarili anketo v Google docsu, ki je obsegala 14 vprašanj. V anketi smo postavljali vprašanja, ki bi nam pomagala potrditi ali zavreči hipoteze. Anketo je rešilo 165 anketirancev.

Prvo vprašanje ankete se je nanašalo na spol in drugo na starost anketirancev (slika 22).



The image shows a survey form with a purple header and a light purple border. The title is 'Programiranje z Raptorjem'. Below the title, there is a short introduction in Slovenian: 'Smo mladi raziskovalci, dijaki Elektro in računalniške šole ŠC Velenje, 2 in 1. letnika Lovro, Aleksander in Žan, ter prosimo, da si vzamete dve minuti časa in izpolnite našo anketo.' The form contains two sections: 'Spol' (Gender) with radio buttons for 'Moški' (Male) and 'Ženska' (Female), and 'Starost' (Age) with radio buttons for age ranges: 'do 15 let', '16 do 20 let', '21 do 25 let', '26 do 30 let', '31 do 35 let', '36 do 40 let', and '41 do 45 let'.

Slika 22: Izgled 1. dela ankete

Na naslednji sliki (slika 23) opazimo vprašanja, ki se nanašajo na osnovno znanje anketiranca o programiranju ter če je že kakšnega naredil.

Ali ste že kdaj naredili računalniški program?

- Da
- Ne

Ali se želite naučiti izdelati računalniški program?

- Da
- Ne
- Nevem

Kaj je Raptor v računalniškem svetu?

- Dinozaver
- Igra ki je izšla leta 2014.
- Program za uničevanje računalnikov.
- Orodje za ustvarjanje izvršljivih diagramov poteka (algoritmov).
- Program za ustvarjanje rap glasbe.
- Novejši tip grafične kartice.
- Ne vem.
- Nič od naštetega.

Slika 23: Primer 2. dela ankete

Naslednja vprašanja (slika 24) sprašujejo o tem, kaj je diagram poteka, in osnovni dve vprašanji, pri katerih ne potrebujemo programerskega predznanja, temveč splošno razgledanost.

Kaj je diagram poteka?

- Grafični prikaz strukture programa.
- To je glavna funkcija v programskih jezikih.
- Je splošen postopek za reševanje problema.
- Nič od naštetega.
- Ne vem.

Kaj je programiranje?

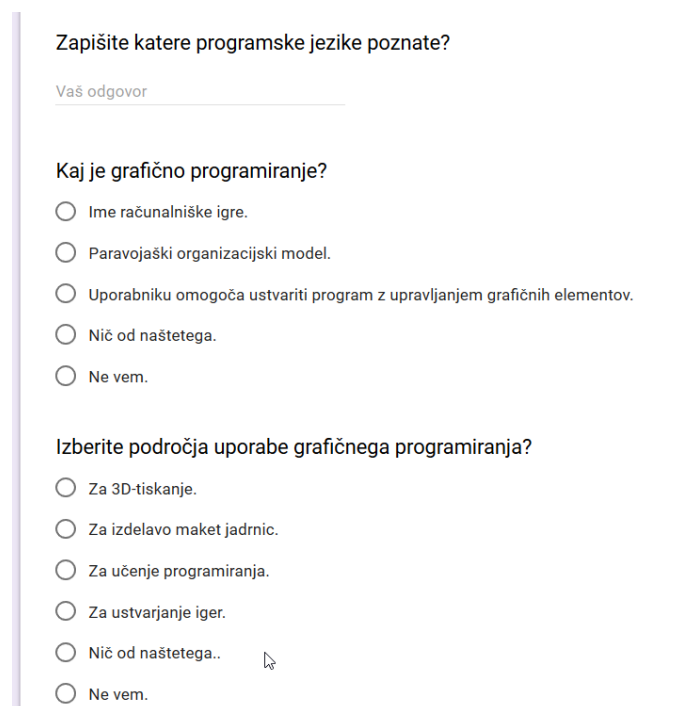
- To je risanje risbe, katera prikazuje končni rezultat programa.
- To je ustvarjanje programa s programskim jezikom.
- Del gledališke predstave, kateri se izvaja za odrom.
- Nič od naštetega.
- Ne vem.

Kaj je program v računalništvu?

- Program je del gledališke predstave.
- Je algoritem, zapisan v programskem jeziku, ki se ga lahko izvaja na računalniku.
- Nič od naštetega..
- Ne vem.

Slika 24: Primer 3. dela ankete

Naslednja tri vprašanja (slika 25) sprašujejo po poznavanju programskih jezikov (anketirancu znane programske jezike vpišejo v vnosno polje) ter o grafičnem programiranju in uporabi le-tega.



Zapišite katere programske jezike poznate?

Vaš odgovor \_\_\_\_\_

Kaj je grafično programiranje?

- Ime računalniške igre.
- Paravojaški organizacijski model.
- Uporabniku omogoča ustvariti program z upravljanjem grafičnih elementov.
- Nič od naštetega.
- Ne vem.

Izberite področja uporabe grafičnega programiranja?

- Za 3D-tiskanje.
- Za izdelavo maket jadrnic.
- Za učenje programiranja.
- Za ustvarjanje iger.
- Nič od naštetega..
- Ne vem.

Slika 25: Primer 2. dela ankete

Naslednji dve vprašanji (slika 26) sprašujeta po konstantah ter spremenljivkah v računalniškem svetu, kar poznajo le anketiranci s predznanjem iz programiranja.

**Kaj je to konstanta v računalniškem svetu?**

- Nekaj kar se le enkrat spreminja.
- Vrednost števila  $\pi=3,1415926$ .
- Vrednost, ki je določena in se med programom ne spreminja.
- To je del življenja.
- Nič od naštetega.
- Ne vem.

**Kaj je spremenljivka v računalniškem svetu?**

- Beseda, ki ima več pomenov.
- Funkcija.
- Neznana količina.
- Ne vem.
- Ime, kateremu se priredi neka vrednost in se med izvajanjem programa spreminja.
- Nič od naštetega.

Slika 26: 5. del ankete

Za zadnje vprašanje nas je zanimalo, ali anketiranci vedo, kaj je funkcija (slika 27).

**Kaj je FUNKCIJA v računalništvu ali matematiki?**

- Je v program del kode ali podprogram, ki jo uporabimo tako, da jo "kličemo".
- Namen uporabe neke stvari.
- Nič od naštetega.
- Preslikava, ki vsakemu elementu množice A priredi natanko en element množice B.
- Ne vem.

**Hvala, da ste si vzeli čas in izpolnili našo anketo!**

Lovro, Aleksander in Žan

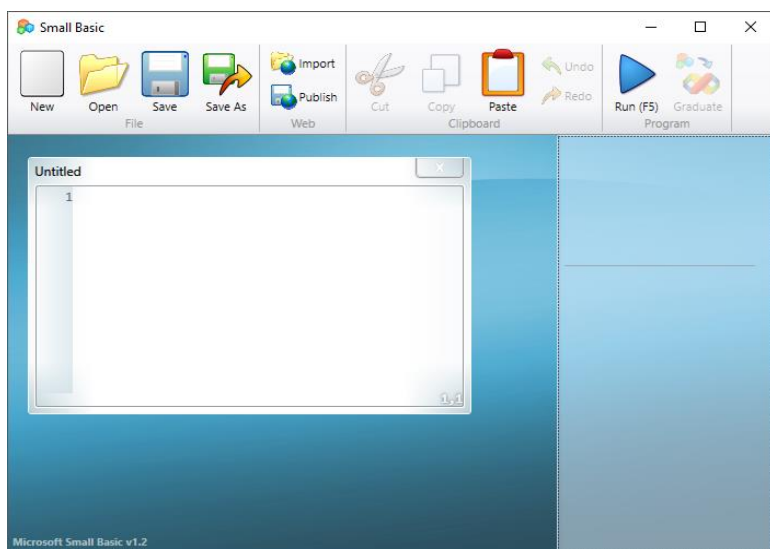
Slika 27: Zanje vprašanje ankete

## 4 RAZPRAVA

Sledi razprava, v kateri smo primerjali programa Small Basic in LARP s programom Raptor.

### 4.1.1 Primerjava programa Small Basic in programa Raptor

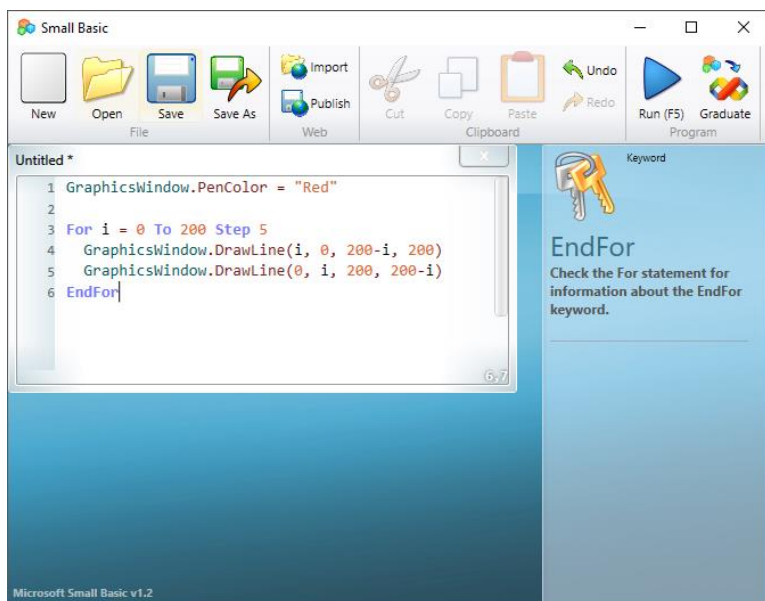
**Small Basic** je brezplačen program za Windows, ki ga je razvilo podjetje Microsoft. Namenjen je za programiranje oz. pisanje programov (slika 29) v jeziku, ki je podoben BASIC-u. Raptor podpira osnovne podatkovne tipe: nize, cela ter realna števila. Raptor tudi podpira grafično risanje (s knjižnico Turtle Graphics), Microsoft Kinect, delo z datotekami, predvajanje zvočnih datotek ter razširitvene knjižnice.



Slika 28: Glavno okno Small Basica, lasten vir

Končne programe lahko objavimo na njihovem spletišču, vendar jih ne moremo izvoziti kot .exe tip datoteke. Small Basic (slika 28) nam ponuja tudi barvno označevanje sintakse (angl. Syntax Highlighting) med urejanjem ter uradno dokumentacijo, objavljeno na spletu. Razvoj programa Small Basic se je najverjetneje ustavil, saj je bil nazadnje posodobljen oktobra leta 2015.

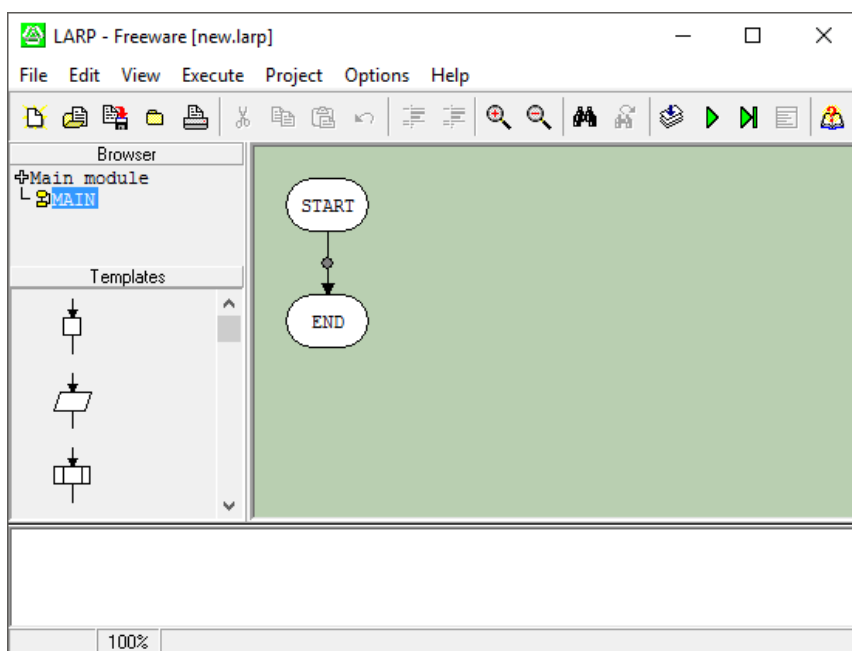




Slika 29: Primer programa v Small Basicu, lasten vir

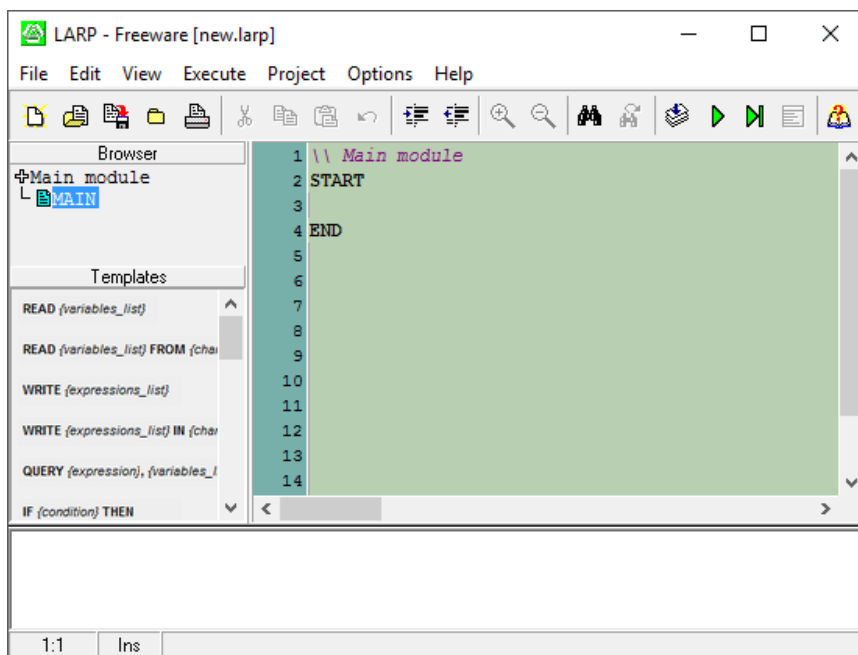
#### 4.1.2 Primerjava programa LARP in programa Raptor

LARP je brezplačen program za Windows. Možno ga je tudi kupiti, za \$ 14.99. Namenjen je programiranju in oblikovanju programov, z uporabo programskega jezika (slika 31) ali oblikovanjem diagramov poteka (slika 30).



Slika 30: Glavno okno LARP-a za pisanje diagramov poteka, lasten vir

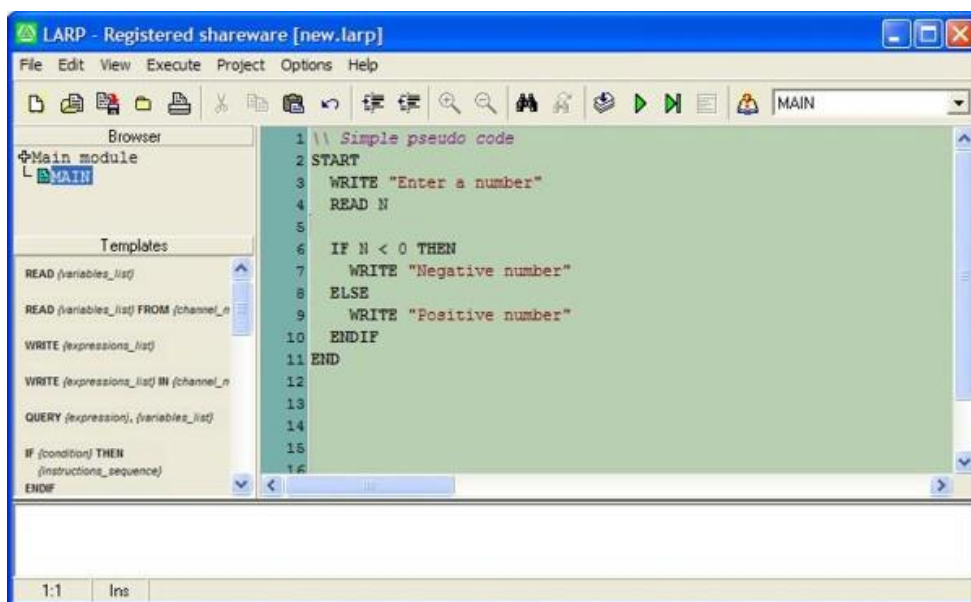
LARP podpira osnovne podatkovne tipe: nize ter realna števila.



Slika 31: Okno LARP-a za pisanje kode, lasten vir

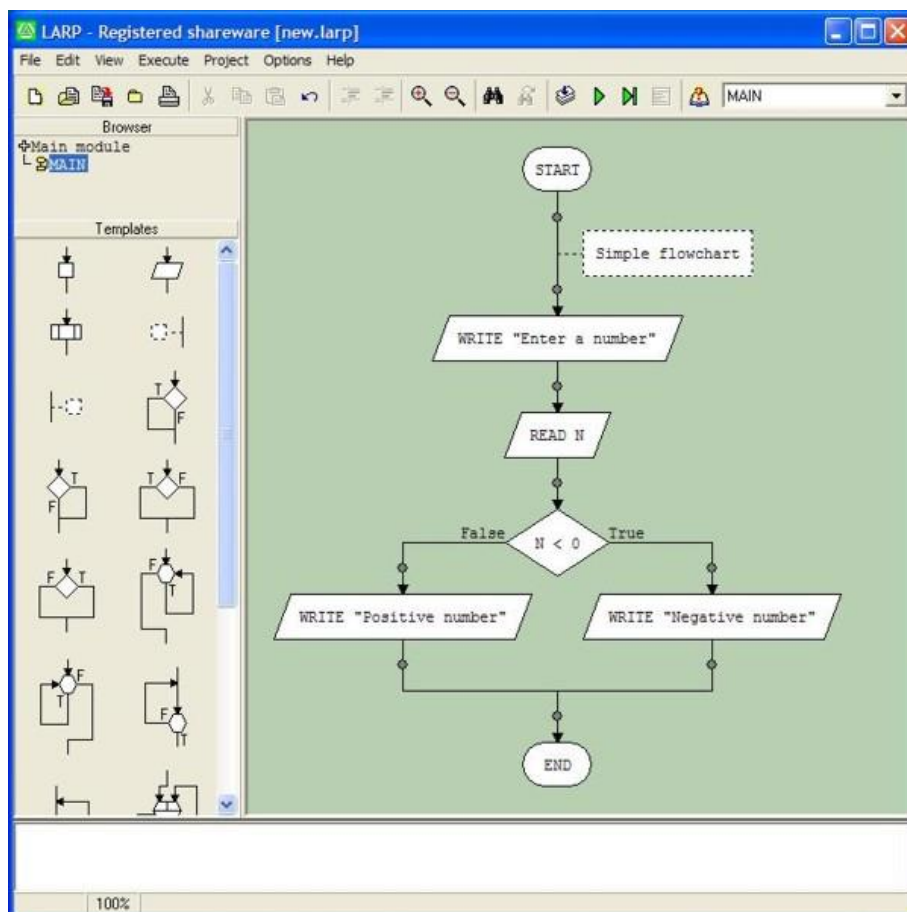
Primer programa v LARP-u, sintaksi ter diagramu poteka:

Uporabnik vnese število, program pa preveri, a je to število negativno ali pozitivno.



Slika 32: Primer sintakse v LARP-u, lasten vir

Razvoj programa LARP se je ustavil, saj je bil nazadnje posodobljen leta 2008.



Slika 33: Primer diagrama poteka v LARP-u, lasten vir

LARP ponuja programiranje o obliki sintakse (slika 32) ter diagramov poteka (slika 33), vendar ni tako razširjen kot sta Small Basic in Raptor. Programe v LARP-u ne moramo izvršiti. Logično, je da bi Small Basic ali Raptor bili boljši odločitvi, a moramo izbrati med samo sintakso (Small Basic) ali vizualnim programiranjem (Raptor).

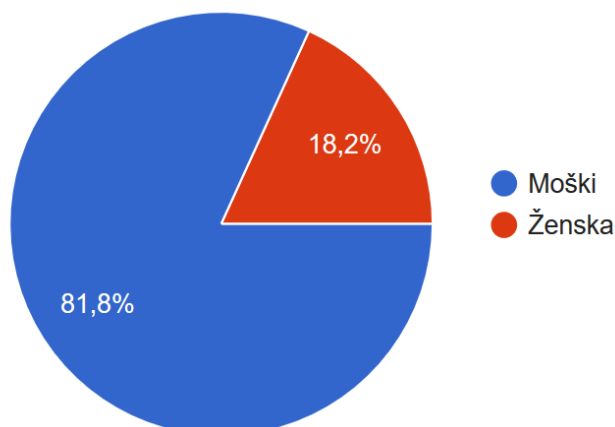
#### 4.1 Težave pri učenju grafičnega programiranja

Pri projektne tednu smo opazili nekatere težave pri učenju grafičnega programiranja. Na primer, dijaki so imeli težave z razumevanjem simbola za prireditev, delovanja in uporabe le-tega v kasnejših diagramih poteka. To težavo smo rešili s ponovno razlago in z lažjimi primeri.

#### 4.2 Rezultati ankete

Pogledali bomo, kako so anketiranci odgovarjali na vprašanja. Anketiranci so bili večinoma moškega spola, kar prikazuje (graf 1).

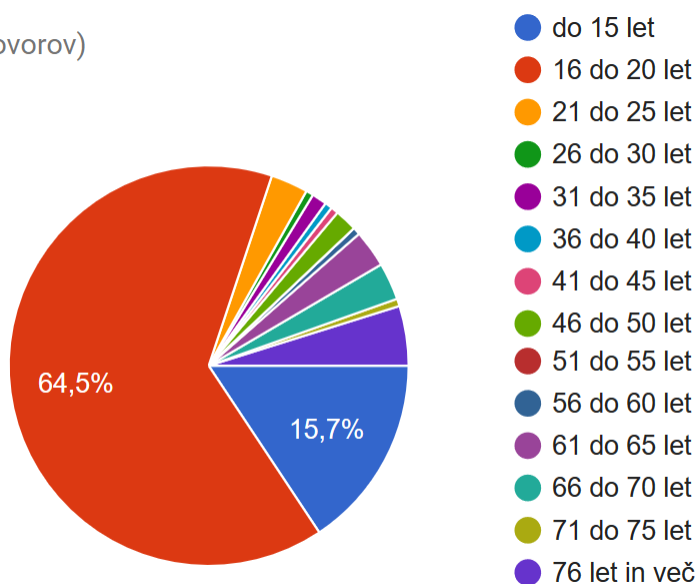
## Spol (165 odgovorov)



Graf 1: Spol anketirancev, lastni graf

Največ anketirancev je bilo starih od 15 do 20 let (graf 2).

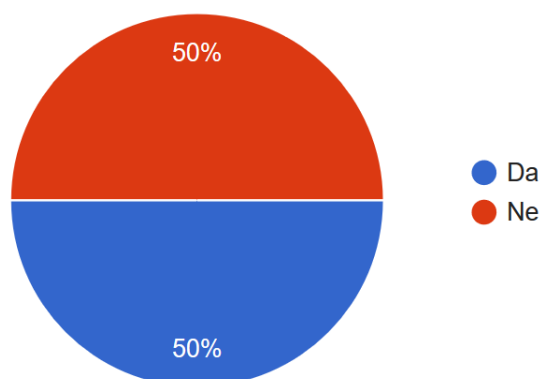
## Starost (166 odgovorov)



Graf 2: Starost anketirancev, lasten graf

V naslednjem grafu je razvidno, da je kar polovica anketirancev naredilo že vsaj en računalniški program (graf 3).

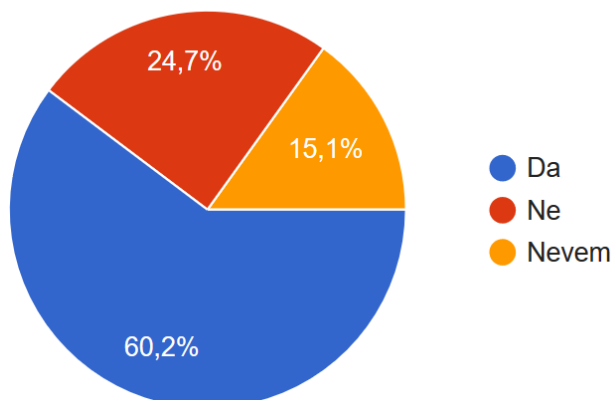
## Ali ste že kdaj naredili računalniški program? (166 odgovorov)



Graf 3: Ali ste že kdaj naredili računalniški program?, lasten graf

V naslednjem grafu so prikazani odgovori anketirancev na vprašanje, ali se želijo naučiti izdelati računalniški program. 60 % se jih želi naučiti, kako izdelati računalniški program, skoraj 25 % pa se jih tega ne želi.

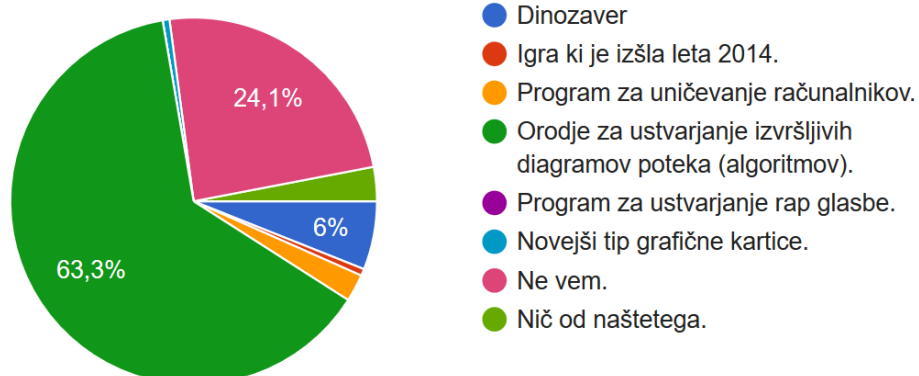
## Ali se želite naučiti izdelati računalniški program? (166 odgovorov)



Graf 4: Ali se želite naučiti izdelati računalniški program?, lasten graf

Graf 5 prikazuje odgovore anketirancev na vprašanje, kaj je Raptor v računalniškem svetu. Iz grafa je razvidno, da 63 % anketirancev ve, kaj je Raptor v računalniškem svetu (graf 5).

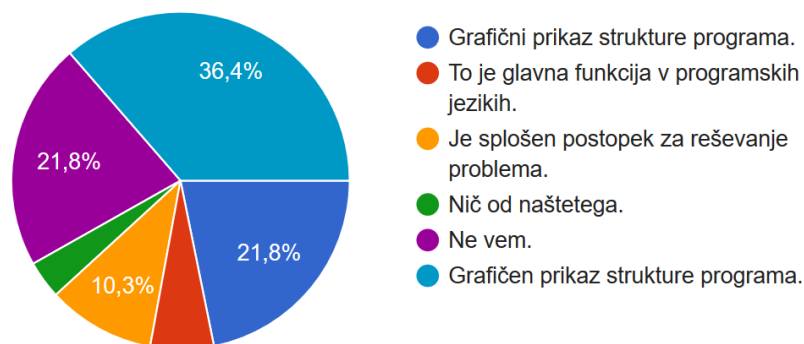
## Kaj je Raptor v računalniškem svetu? (166 odgovorov)



Graf 5: Kaj je Raptor v računalniškem svetu?, lasten graf

Graf 6 prikazuje odgovore anketirancev na vprašanje, kaj je diagram poteka. 58% anketirancev je pravilno odgovorilo na vprašanje.

## Kaj je diagram poteka? (165 odgovorov)



Graf 6: Kaj je diagram poteka, lastni graf

Na vprašanje kaj je programiranje, je več kot 90 % anketirancev odgovorilo pravilno (graf 7).

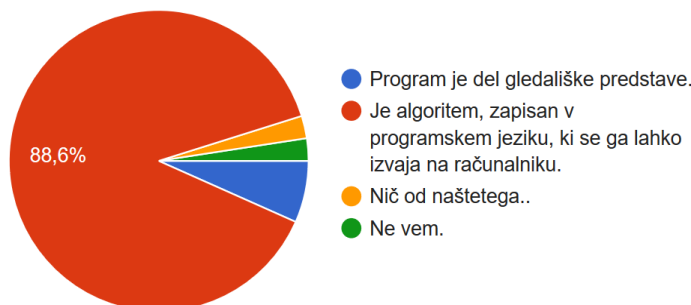
## Kaj je programiranje? (166 odgovorov)



Graf 7: Kaj je programiranje?, lasten graf

Na vprašanje kaj je program v računalništvu, je več kot 88,6 % anketirancev odgovorilo pravilno (graf 8).

**Kaj je program v računalništvu?** (166 odgovorov)

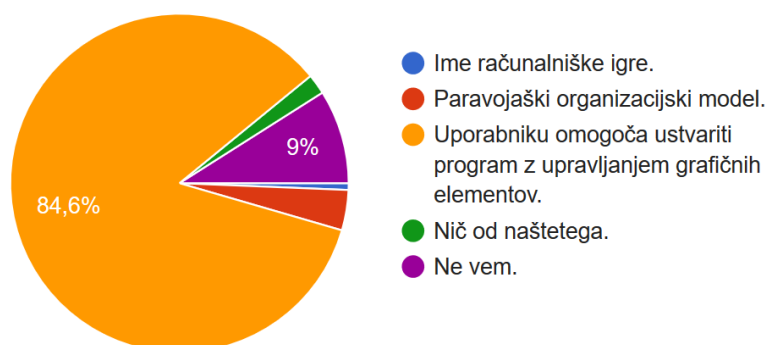


Graf 8: Kaj je program v računalništvu?, lastni graf

Na deveto vprašanje: zapišite, katere programske jezike poznate, smo dobili 130 odgovorov. V vnosno polje so lahko anketiranci vpisali njim znan programski jezik; od teh odgovorov je bilo 33 neveljavnih. V ostalih odgovorih je bilo najpogosteje zapisano: C#, C++, Python, Raptor in JavaScript. Presenetilo nas je, da je povprečje anketirancev poznalo vsaj 4 programske jezike.

Skoraj 85 % anketirancev je vedelo, kaj pomeni grafično programiranje.

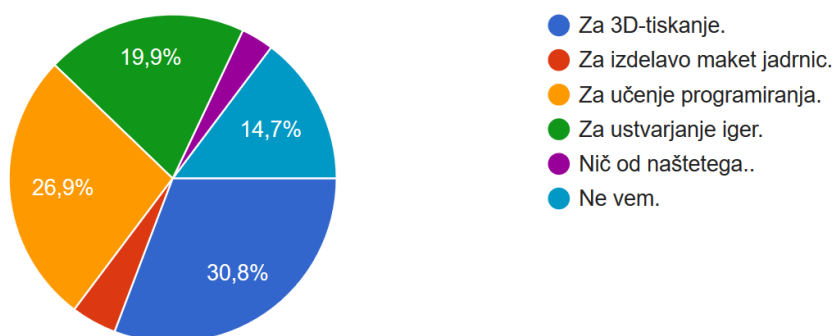
**Kaj je grafično programiranje?** (156 odgovorov)



Graf 9: Kaj je grafično programiranje?

Skoraj 27 % anketirancev je vedelo za področja uporabe grafičnega programiranja.

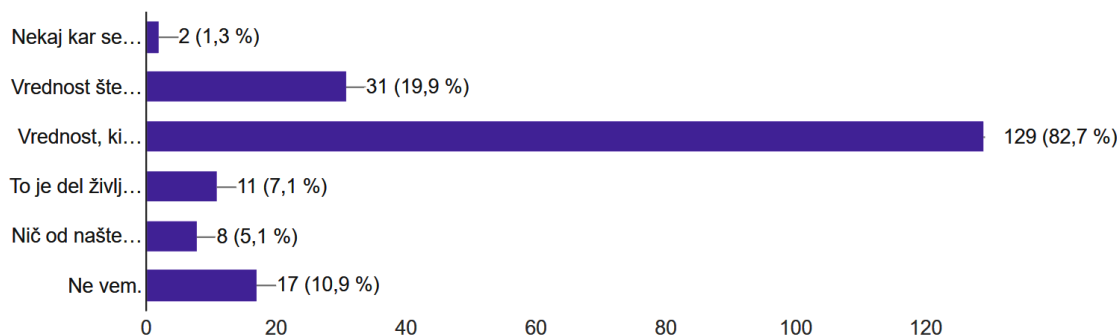
**Izberite področja uporabe grafičnega programiranja?** (156 odgovorov)



Graf 10: Izberite področja uporabe grafičnega programiranja?

Na 12. vprašanje o tem, kaj je računalniška konstanta, je pravilno odgovorilo 129 anketirancev ali 82 %, ki vedo, da je to: Vrednost, ki je določena, in se med programom ne spreminja.

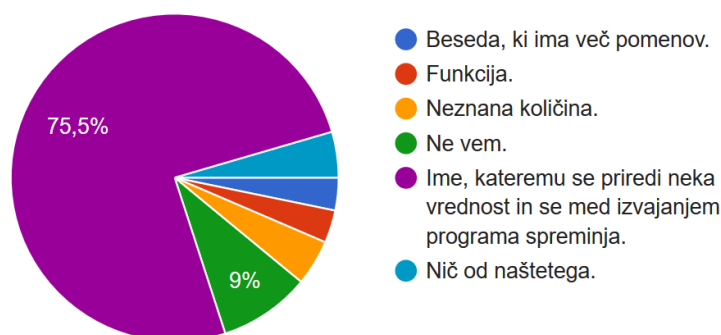
### Kaj je to konstanta v računalniškem svetu? (156 odgovorov)



Graf 11: Kaj je računalniška konstanta?

Na 13. vprašanje o tem, kaj je spremenljivka v računalniškem svetu, je 75,5 % poznalo pravilen odgovor.

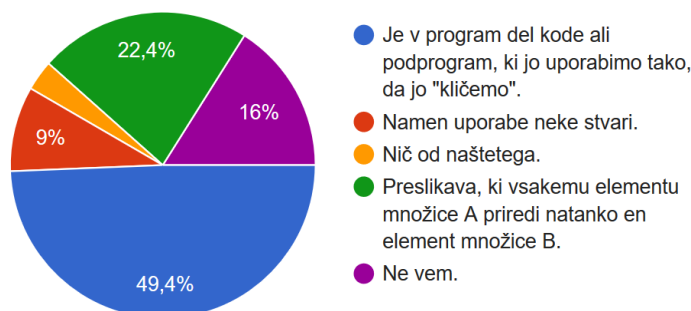
### Kaj je spremenljivka v računalniškem svetu? (155 odgovorov)



Graf 12: Kaj je spremenljivka v računalniškem svetu?

Na 14. vprašanje o tem, kaj je funkcija v računalništvu ali matematiki, je skupaj vedelo skoraj 71 % (pravilna odgovora sta prvi in četrti). Torej, iz tega sklepamo, da je povprečno zanje matematike le 22,4 %, kar je res skromno.

### Kaj je FUNKCIJA v računalništvu ali matematiki? (156 odgovorov)



Graf 13: Kaj je funkcija v računalništvu ali matematiki?



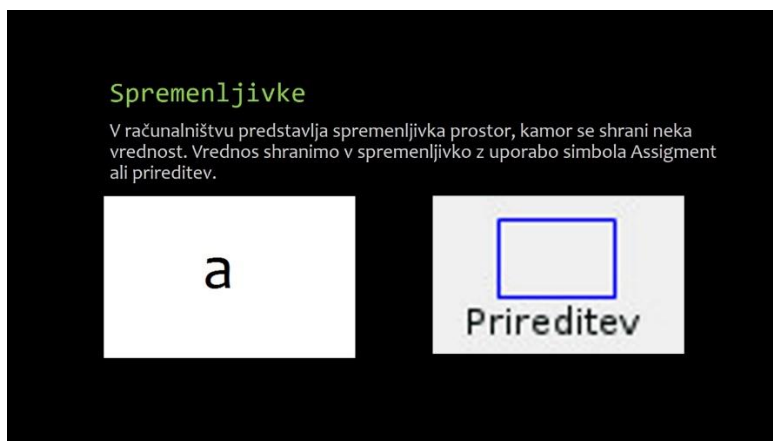
### 4.3 Predlogi za izboljšanje učenja grafičnega programiranja s programom Raptor

Te predloge lahko strnemo v: najučinkoviteje bi bilo, da imajo učeči se pred učenjem grafičnega programiranja približno enako predznanje (da kasneje ne pride do zapletov), smiselno je prilagoditi težavnost in zapletenost nalog (glede na stopnjo (pred)znanja skupine), uporabo prevedene pomoči in uporabo predstavitve npr. PowerPointa (slika 34).



Slika 34: Naslovna stran predstavitve, lasten vir

Tako npr. smo v PowerPoint predstavitvi predstavili osnovne simbole za namene grafičnega programiranja v programu Raptor (slika 35).



Slika 35: Predstavljanje simbola spremenljivke in uporabe le te v programu Raptor, lasten vir

PowerPoint predstavitev vsebuje predstavitev posameznih simbolov, podobno kot v pomoči, angl. »help« datoteki. Predstavljena je uporaba in namen simbolov ter primeri uporabe le-teh v programiranju. Pri učenju programiranja je zelo pomembno, da učeča se oseba razume osnovne simbole, kot so simbol za prireditev, vhod (vnos), izhod (izpis), da lahko kasneje razume delovanje simbola za odločitev, simbola za klic funkcij ter simbola zanka. Skozi učenje osnov grafičnega programiranja se privaja na logiko programiranja, načrtovanje obsežnejših programov ter pisanje le-teh v kodi.

## 5 ZAKLJUČEK

Pred samim raziskovanjem smo si zastavili naslednje hipoteze.

1. Domnevamo, da več kot 50 % anketirancev ve, kaj so diagrami poteka.

Glede na rezultate ankete lahko to hipotezo ovržemo, saj kaj je diagram poteka, ve le 37 % anketirancev.

2. Domnevamo, da več kot 30 % anketirancev ve, kaj je Raptor.

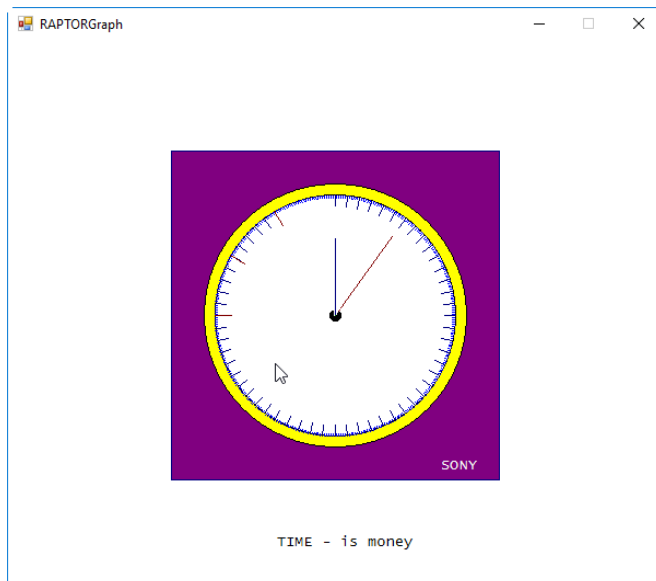
Glede na rezultate ankete lahko to hipotezo potrdimo, saj 64 % anketirancev ve, kaj je Raptor.

3. Predvidevamo, da je Raptor primerno grafično orodje za učenje osnov programiranja z diagrami poteka.

Glede na odzive učencev na projektne tednu ter na primerjavo s programom LARP in Visual Logic 2.7, lahko to hipotezo potrdimo. Rezultati primerjave so predstavljeni v razpravi.

4. Predvidevamo, da je z Raptorjem mogoče ustvariti kompleksne diagrame poteka in razumljivo vizualizirati izvajanje le-teh.

Četrto hipotezo lahko potrdimo, saj ima Raptor v mapi samples (vzorci) zahtevne primere vizualnega programiranja, kot je npr. delujoča ura (slika 36) ali učenje objektnega programiranja in diagramov UML.



Slika 36: Delujoča ura v Raptorju, lastna slika

5. Predvidevamo, da je z Raptorjem mogoče enostavno obogatiti pouk programiranja za osnovnošolce in srednješolce.

Peto hipotezo lahko potrdimo, saj smo sami med projektne tednom uporabili Raptor in Python kot orodje za učenje osnov programiranja. Tako udeleženi dijaki kot mi smo se marsikaj novega naučili. Prav tako smo ga tudi sami uporabljali pri spoznavanju osnov programiranja in algoritmov.

## 6 ZAHVALA

Raziskovalna naloga ne bi bila v takšni obliki, če nam pri nastajanju ne bi pomagalo veliko ljudi.

Zahvala je torej namenjena naslednjim:

- mentorjema Karmen Grabant in Nedeljko Grabantu za pomoč, voljo, vztrajnost, njun prosti čas in spodbudo pri nastajanju;
- staršem;
- Sonji Lubej, prof., za lektoriranje;
- Jolandi Melanšek, prof., za lektoriranje angleškega povzetka;
- učiteljem ERŠ in ravnatelju Simonu Konečniku, univ. dipl. inž. za vso podporo in spodbudo;
- recenzentu raziskovalne naloge;
- komisiji Mladih raziskovalcev in koordinatorici gibanja Mladi raziskovalci Karmen Hudournik;
- dijakom, učiteljem in ostalim, ki so izpolnjevali anketo,
- vsem neomenjenim, ki so kakorkoli pomagali pri izdelavi naloge.

## 7 POVZETEK

V raziskovalni nalogi je opisan program Raptor, njegova praktična uporaba in grafično programiranje. Raptor je za začetek programiranja dokaj preprosto orodje za reševanje računalniških problemov, ki dovoli uporabniku, da ustvari izvršljive diagrame poteka (algoritme). Grafično se lahko spremlja potek in tudi vrednosti spremenljivk. Napisan je bil za študente ameriškega zrakoplovstva, ki se učijo »računalniških poklicev«, se pravi pravil, kako razviti sposobnosti reševanja problemov in učenja algoritmičnega razmišljanja. V računalništvu pod pojmom grafično programiranje ali vizualni programski jezik razumemo katerikoli programski jezik, ki uporabniku dovoljuje ustvarjanje programov z manipuliranjem grafičnih elementov programa namesto besedilnega pisanja kode. Na kratko smo predstavili tudi nekatere druge programe za grafično programiranje in primerjali s programom Visual Logic 2.7 ter ugotavljali razlike (prednosti in slabosti) med programoma. V raziskovalni nalogi so predstavljeni rezultati ankete in ugotovitve hipotez. To zanimivo programersko orodje je predstavljeno in dostopno na spletni strani <http://raptor.martincarlisle.com/>.

Ugotovili smo, da ve kaj je Raptor 64% anketirancev in s tem potrdili hipotezo, da več kot 30 % anketirancev ve, kaj je Raptor v računalniškem svetu.

## 8 VIRI

- [1] Programiranje v Sketchpadu: [http://blog.interfacevision.com/assets/img/posts/example\\_visual\\_language\\_sketchpad\\_01.jpg](http://blog.interfacevision.com/assets/img/posts/example_visual_language_sketchpad_01.jpg), 4. 10. 2016
- [2] Programski jezik ADL: [http://blog.interfacevision.com/assets/img/posts/example\\_visual\\_language\\_adl.jpg](http://blog.interfacevision.com/assets/img/posts/example_visual_language_adl.jpg), 4. 10. 2016
- [3] Primer v Pascalu: <https://ucarecdn.com/fd283b2b-345e-4f8a-b916-842627544a31/>, 6. 10. 2016
- [4] Programski jezik Alice: [http://www.sciencebuddies.org/Files/5066/8/CompSci\\_img066.jpg](http://www.sciencebuddies.org/Files/5066/8/CompSci_img066.jpg), 4. 10. 2016
- [5] Programski jezik Etoys: <https://wiki.sugarlabs.org/images/thumb/0/0b/Etoys-Launcher.png/600px-Etoys-Launcher.png>, 4. 11. 2016
- [6] Primer izgleda programa Alice: <https://upload.wikimedia.org/wikipedia/commons/5/57/Alice-2-screenshot.jpg>, 16. 12. 2016
- [7] Ikone v jeziku DRAKON: [https://upload.wikimedia.org/wikipedia/commons/e/e4/Icons\\_of\\_Visual\\_Programming\\_Language\\_-\\_DRAKON-.png](https://upload.wikimedia.org/wikipedia/commons/e/e4/Icons_of_Visual_Programming_Language_-_DRAKON-.png), 16. 12. 2016
- [8] Primer ustvarjanja diagrama v jeziku DRAKON [[https://a.fsdn.com/con/app/proj/dragon-editor/screenshots/win\\_normal2.png](https://a.fsdn.com/con/app/proj/dragon-editor/screenshots/win_normal2.png)], 16. 12. 2016
- [9] Okno programa Small Basic, ter primer kode: <http://img.creativemark.co.uk/uploads/images/706/10706/largeImg.png>, 22. 12. 2016
- [10] Primer programa Scratch :  
[https://upload.wikimedia.org/wikipedia/commons/7/76/Scratch\\_2.0\\_Default\\_screen.png](https://upload.wikimedia.org/wikipedia/commons/7/76/Scratch_2.0_Default_screen.png), 16. 12. 2016
- [11] Program Visual Logic: <https://i.ytimg.com/vi/TbG3hMwO0wA/maxresdefault.jpg>, 18. 12. 2016
- [12] Primer algoritma v programu LARP: <http://larp.marcolavoie.ca/en/description/images/descr3.jpg>, 16. 12. 2016
- [13] Primer sintakse v LARP [<http://larp.marcolavoie.ca/en/description/images/descr1.jpg>], 22. 12. 2016
- [14] Primer diagrama poteka v LARP <http://larp.marcolavoie.ca/en/description/images/descr3.jpg>, 22. 12. 2016
- [15] knjiga: Duhovnik, M. Kljanin, M. Opalić, Inženirska grafika, UL Fakulteta za strojništvo, ISBN: 978-961-6536-32-5, 2009.
- [16] O algoritmu: <https://sl.wikipedia.org/wiki/Algoritem>, 4. 12. 2016
- [17] [[http://blog.interfacevision.com/assets/img/posts/example\\_visual\\_language\\_grail\\_01.png](http://blog.interfacevision.com/assets/img/posts/example_visual_language_grail_01.png)], 4. 12. 2016
- [18] O algoritmu: <https://sl.wikipedia.org/wiki/Algoritem>, 4. 12. 2016
- [19] [[http://blog.interfacevision.com/assets/img/posts/example\\_visual\\_language\\_grail\\_01.png](http://blog.interfacevision.com/assets/img/posts/example_visual_language_grail_01.png)], 4. 12. 2016
- [20] O vizualni programskih jezikih: [https://en.wikipedia.org/wiki/Visual\\_programming\\_language](https://en.wikipedia.org/wiki/Visual_programming_language), 22. 12. 2016

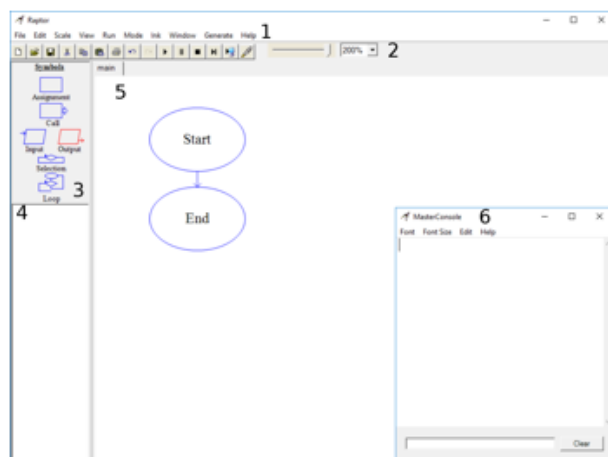
[21] [https://en.wikipedia.org/wiki/Alice\\_\(software\)](https://en.wikipedia.org/wiki/Alice_(software)), 4. 12. 2016

[22] Program: Microsoft Small Basic: [https://en.wikipedia.org/wiki/Microsoft\\_Small\\_Basic](https://en.wikipedia.org/wiki/Microsoft_Small_Basic), 4. 12. 2016

[23] O jeziku UML: [https://de.wikipedia.org/wiki/Unified\\_Modeling\\_Language](https://de.wikipedia.org/wiki/Unified_Modeling_Language), 26. 12. 2016

## 9 PRILOGA

Prevod angleške pomoči za program Raptor je obsežen in presega 70 strani v papirnem formatu A4. Zato je tu samo primer prvega dela. To pomoč pa nam ni uspelo še strokovno in slovnično urediti.



1. Menijska vrstica
2. Orodna vrstica
3. Podokno simboli
4. Podokno spremenljivke
5. Glavno okno
6. Glavna konzola

### 2.1 Imena, pravila in predlogi (Names, rules and suggestions)

Ker je Raptor programsko okolje za grafično programiranje, se v njem ukvarjamo manj z besedami kot v drugih programskih jezikih. Kakorkoli, se besede še vedno pojavljajo v Raptorju kot imena spremenljivk, imena diagramov, postopkov, funkcij. V računalniških krogih, so ta imena znana kot *identifikatorji* in programi sledijo določenim pravilom, da lahko uporabljajo slovnično pravilna imena.

Pravila za imena v Raptorju:

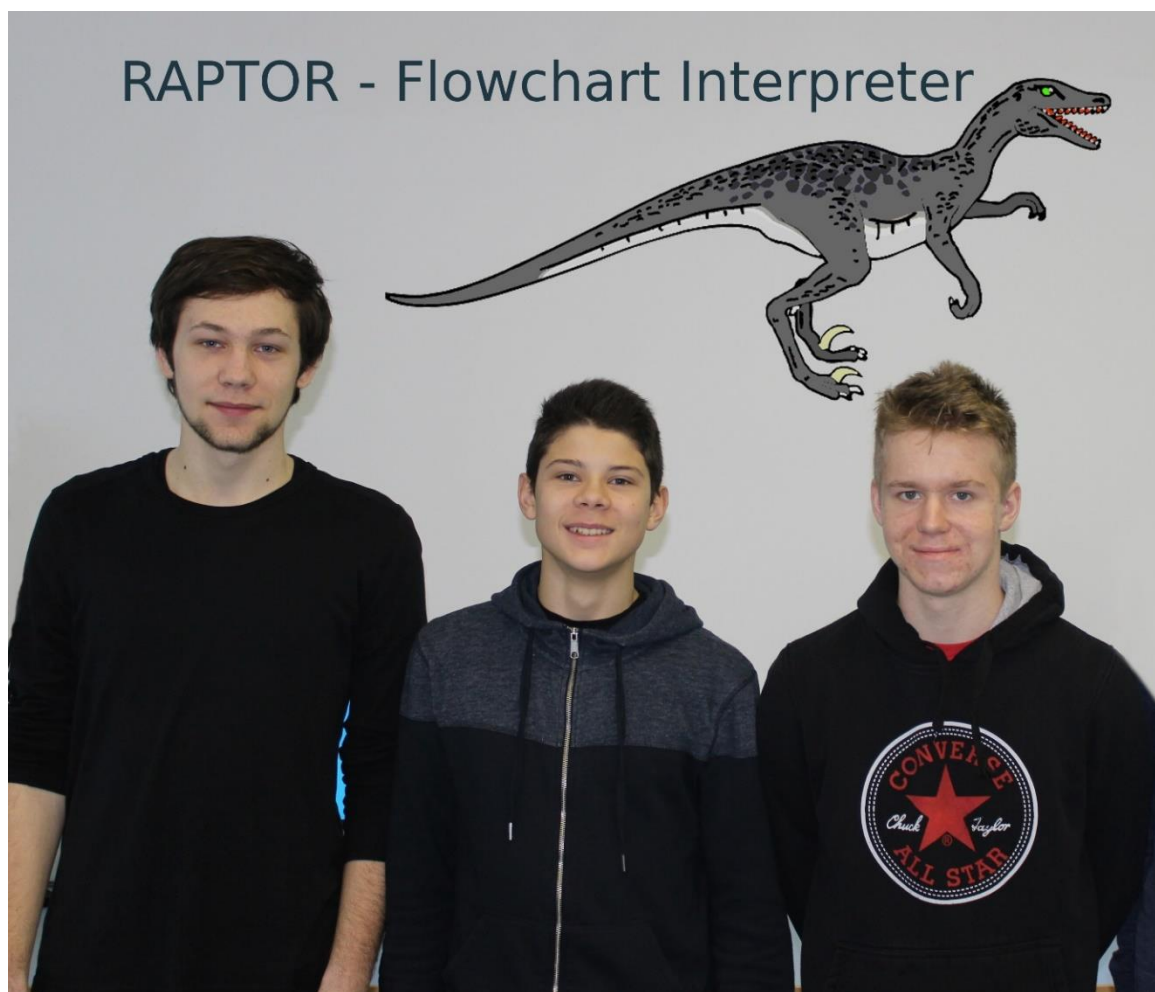
1. Ime se mora začeti s črko.
2. Po prvi črki lahko sledi katerikoli stavek sestavljen iz črk, števil, podčrtaji.
3. Presledki v imenu niso dovoljeni, saj presledek označuje konec imena.
4. Začetnice črke nimajo vpliva na ime naprimer `get mouse button` pomeni isto kot `Get_Mouse_Button` ali `GET_MOUSE_BUTTON` (v drugih programskih jezikih pa je običajna ločevanje velikih in malih črk t.j. angl. case sensitive).
5. Ime ne sme predstavljati več kot eno spremenljivko v programu. Zato:
  - a. program ne sme uporabiti imena spremenljivke `e`, saj je `e` v Raptorju rezervirana beseda (angl. Pre-defined)
  - b. program ne sme uporabiti spremenljivke `red`, ker je `red` v Raptorju rezerviran beseda (angl. Pre-defined)
  - c. programer ne sme ustvariti poddiagrama z imenom `Get_Key`, saj je `Get_Key` v Raptorju rezerviran (angl. Pre-defined)

## 10 AVTORJI RAZISKOVALNE NALOGE

**Lovro Suhovršnik** je dijak 2. letnika Elektro in računalniške šole (ERŠ) v Velenju. Za raziskovalno nalogo se je odločil, ker so ga zanimala lastnosti in funkcije algoritmov in grafičnega programiranja z Raptorjem. Zanima ga tudi programiranje s Pythonom, urejanje slik v programu GIMP. V prihodnosti se želi ukvarjati s računalništvom.

**Žan Pukmajster** je dijak 1. letnika Elektro in računalniške šole (ERŠ) v Velenju. Za raziskovalno nalogo se je odločil, ker se želi poglobiti v vizualno programiranje. Ukvarja se z programskimi jeziki C++, C#, GML, ter Python. V prihodnosti si želi biti odličen programer.

**Aleksander Flis** je dijak 2. letnika Elektro in računalniške šole (ERŠ) v Velenju. Za raziskovalno nalogo se je odločil. Ker so ga zanimala funkcije in lastnosti algoritmov, ter grafično programiranje v Raptorju. Zanima ga tudi programiranje s Pythonom in C++. V prihodnosti se želi ukvarjati z programiranjem.



Slika 38: Lovro Suhovršnik, Aleksander Flis, Žan Pukmajster (s leve na desno)