

UNIVERZA V MARIBORU  
FAKULTETA ZA NARAVOSLOVJE IN MATEMATIKO

Oddelek za matematiko in računalništvo

# **DIPLOMSKO DELO**

Gregor Nemec

Maribor, 2015

# KAZALO

<b>1</b>	<b>UVOD</b>	<b>1</b>
<b>2</b>	<b>Uporaba računalnika v izobraževanju</b>	<b>2</b>
2.1	Zgodovin uporabe računalnika v izobraževanju računalništva. . . . .	2
2.2	Pregled aplikacij v izobraževanju računalništva. . . . .	2
2.2.1	Sisatemi CAI ( <i>ang Computer Assisted Instruction</i> ) . . . . .	2
2.2.2	Sistemi CAA ( <i>ang. Computer Assisted assessment</i> ). . . . .	3
2.3	Prihodnost in smer aplikacij v izobraževanju računalništva. . . . .	3
<b>3</b>	<b>Učenje programiranja</b>	<b>3</b>
3.1	Zgodovina programskih jezikov v izobraževanju . . . . .	3
3.2	Kaj je programiranje? . . . . .	4
3.3	Programske paradigme . . . . .	4
3.4	Problematika začetkov učenja programiranja . . . . .	5
3.5	Predlagane rešitve . . . . .	5
3.6	Osnovni koncepti programiranja . . . . .	6
3.7	Programiranje v OŠ . . . . .	6
3.8	Programiranje v SŠ . . . . .	6
<b>4</b>	<b>Spletni portali za učenje programiranja</b>	<b>6</b>
4.1	Prednosti spletnih portalov za učenje programiranja. . . . .	6
4.2	Primer sistemske arhitekture spletnega portala za učenje programiranja . . . .	6
4.2.1	Analiza programske kode . . . . .	7
<b>5</b>	<b>Metode in strategije pri uporabi spletnih portalov</b>	<b>7</b>
5.1	Učne strategije . . . . .	7

5.1.1	Model aktivnega učenja . . . . .	7
5.1.2	Učenje na daljavo . . . . .	8
5.2	Tipi nalog . . . . .	8
5.2.1	Zapolni prazna mesta . . . . .	8
<b>6</b>	<b>Kategoriziranje spletnih portalov</b>	<b>8</b>
6.1	Vrsta vsebine . . . . .	8
6.2	Programski jeziki . . . . .	9
<b>7</b>	<b>Ovrednotenje izbranih spletnih portalov in njihove posebnosti</b>	<b>9</b>
7.1	Določitev Kriterijev . . . . .	9
<b>8</b>	<b>Možni načini uporabe spletnih portalov pri puku</b>	<b>9</b>

## PREGLED UPORABLJENIH SIMBOLOV IN OZNAČB

1. **IKT** - informacijsko komunikacijska tehnologija

# 1 UVOD

V svetovnem merilu se pojavlja trend po popularizaciji programiranja ali kodiranja. V zadnjem obdobju so se na spletu pojavili številni portali, kot je na primer *CodeAcademy*, ki ponujajo učenje programiranja.

Testiranje todojev

Testiranje  
novega  
ukaza!

V večini se učenci prvič srečajo s pojmi programiranja pri izbirnih predmetih *Urejanje besedil*, *Multimedija in Računalniška omrežja*. Dijaki se srečajo s programiranjem v 1. letniku pri predmetu informatike. Posebnost so strokovni programi, katerim je osnova računalništvo. Zanimali nas bodo novinci in njihove težave pri začetnih korakih učenju programiranja. Torej vsi učenci in dijaki, ki se šele srečujejo s programiranjem.

Najprej se je uporaba spletne tehnologije in nastanek spletnih portalov za namen učenja programiranja pojavila v akademskem okolju na posameznih univerzah. Zanimal nas bo razlog za nastanek takšnih okolij na univerzah, zato bomo pregledali literaturo in poskušali ugotoviti, zakaj in kako se na višje šolskem področju uporabljajo spletne tehnologije za poučevanje programiranja.

Izluščili bomo predlagane rešitve za uporabo spletnih tehnologij pri učenju programiranja. Spoznali bomo kaj so osnovni koncepti programiranja s katerimi se srečajo novinci in katere so strategije in metode, ki se pri učenju uporabljajo.

Na podlagi pregledanega bomo določili kriterije in kategorizirali ter ovrednotili spletne portale. Najbolj bojo zanimivi tisti spletni portali, ki ponujajo številne programske jezike, urejevalnik besedil, zaganjanje napisane programske kode in neko obliko odziva, ki uporabniku omogoča odkrivanje napak.

Ogledali si bomo kje se uči programiranja na osnovni (OŠ) in srednji šoli (SŠ). Pri katerih izbirnih vsebinah, predmetih in kakšna je vsebina, ki jo predvideva učni nart. Uporabo spletnih portalov bomo skušali umestiti v pouk OŠ in SŠ tako, da bo njihova uporaba najbolj koristna in smiselna.

## 2 UPORABA RAČUNALNIKA V IZOBRAŽEVANJU

Po modelu uporabe računalnika v izobraževanju, njegova uporaba pri učenju programiranja spada v primarno področje, saj sem prištevamo aktivnosti s katerimi želimo uporabnike seznaniti z *delovanjem* in uporabo računalnika oz. sodobno informacijsko komunikacijsko tehnologijo. Računalnik je tista učna vsebina, ki jo obravnavamo (**IKT**) [1]. Računalništvo v osnovi nastopa v dveh pomembnih področjih [2]:

- kot element splošne izobrazbe,
- kot element ožje strokovne - poklicne izobrazbe oz. usposabljanja.

V današnjem času se računalnik kot element splošne izobrazbe kaže kot velika potreba oz. se zdi znanje njegove uporabe samoumevno. Že pri najmlajših otrocih računalnik vzbuja zanimanje in interes. Računalnik je postal intelektualno orodje in pripomoček v vsaki sferi človekove dejavnosti in je prodrl tudi v šolo. Tako imenovana *računalniška pismenost* postaja nujna in zajema vse to kar bi človek moral znati o računalniku in to, kako je potrebno z njim delati, da bo uspešno živel v družbi, ki je osnovana na informacijah (informacijski družbi) [3].

Računalništvo se ne izvaja kot redni predmet, temveč poteka v

### 2.1 ZGODOVIN UPORABE RAČUNALNIKA V IZOBRAŽEVANJU RAČUNALNIŠTVA.

### 2.2 PREGLED APLIKACIJ V IZOBRAŽEVANJU RAČUNALNIŠTVA.

V katero kategorijo spadajo spletne strani za učenje programiranja? Klasifikacija!?

#### 2.2.1 SISATEMI CAI ( *ang Computer Assisted Instruction* )

Računalniško podprt sistem za učenje. -> Točna definicija gerlič.

Katere stvari mora nujno posedovati orodje za učenje novincev, da jim je kar se da v največjo pomoč?

Pomembna sta dva glavna cilja na katerih temelji učenje programerjev novincev:

1. Sistemi za učenje, pomagajo izključno učenju programiranja.
2. Sistemi za pomoč pri programiranju, ko želimo programiranje uporabiti za doogo nekega drugega cilja.

- Novice programming system and language taxonomy - tabela (str. 43).

Skupine programskih orodij, ki so v pomoč novincem so naslednje:

1. mikro svetovi,
2. vizualna okolja za programiranje,
3. okolje za izdelavo modela poteka,
4. okolje za izdelovanje objektov,
5. okolje za risanje in realizacijo algoritmov.

#### 2.2.2 SISTEMI CAA ( *ang. Computer Assisted assessment* ).

Računalniško podprti sistemi za vrednotenje znanja. // To nas toliko nebo zanimalo, razen če kjer najdem podporo za učitelje. -> Ima npr. CodeAcademy možnost sledenja napredka učencem in jih vidi učitelj.

### 2.3 PRIHODNOST IN SMER APLIKACIJ V IZOBRAŽEVANJU RAČUNALNIŠTVA.

Lahko pogledamo smernice in ugotovimo ali so se zares uresničile? Na hitrer pogled vse kaže da so se.

Spletna okolja za programiranje \* Tabele z kriterijami za ocenjevanje, zastarelih orodij! (str. 55)

## 3 UČENJE PROGRAMIRANJA

### 3.1 ZGODOVINA PROGRAMSKIH JEZIKOV V IZOBRAŽEVANJU

Uporaba računalništva v izobraževanju je bila deležna številnih sprememb. Sama uporaba računalnika v izobraževanju je tesno povezana z razvojem računalnikov. Začetno obdobje, 1960 letih prejšnjega stoletja so računalniki bili zelo dragi in veliki glavni računalniki ( *ang. mainframe* ), na njih se je učilo programiranja, a so se uporabljali tudi za druga področja. //-> Terminalska obdobje, Poglej gerliča. V tem obdobju se je za učenje programiranja

uporabljal **FORTTRAN** ali **assembler**. Programi so bili majhi in enostavi, zaradi fizičnih omejitev takratnega delovnega pomnilnika.

V 1970 so na trg prišli manjši računalniki, ki so bili tudi cenejši in zmoglivejši. V tem času pride v ospredje strukturirano programiranje. Najpopularnejši programski jezik je bil **PASCAL**.

V 1980 so se prvič pojavili samostojni osebni računalniki. Programski jeziki v tem obdobju so bili strukturirani in močnega tipa ( *ang. strong type.* ). Med te spada **Ada**, **Modul 2**, **ML** in **naj omenimo še Prolog**. V naslednjem desetletji, 1990 so v ospredje prišli objektno orjentirani programski jeziki, kot sta **JAVA** in **C#** [8].

Metode poučevanja računalništva so se pravtako spreminjale. 1960 so računalnike uporabljali samo za poučevanje programiranja. Povdarek pri predmetih programiranja je bil predvsem na detaljih zmožnosti programskega jezika. Programiranje je bilo omejeno le na reševanje enostavnih primerov in povdarek ni bil na reševanju problemov na splošno.

V 1970 je reševanje problemov in abstrakcija podatkov postala glavni in najpomembnejši del vseh programerskih predmetov, kar velja še danes. Programi so postali večji, bolj interaktivni in spremenil se je vnos podatkov z tekstovnega v grafičnega. Vsebina predmetov računalništva se je hitro razširjala, kakor so se množili številni programski jeziki [8].

### 3.2 KAJ JE PROGRAMIRANJE?

### 3.3 PROGRAMSKE PARADIGME

Paradigma je način kako obravnavamo in gledamo na stvari, je okvir v katerem leži naša interpretacija realnosti sveta. Paradigma najpogosteje pomeni vzorec delovanja v znanstvenem ali drugem raziskovanju. Izraz -programske paradigme- je več pomenka, ki povzema mentalne procese, strategije reševanja problemov, povezave med različnimi paradigmami, programske jezike, stil programiranja in še več (Wikipedia: Paradigma).

Povemo lahko, da je programiranja, hevristična paradigma za algoritme ki rešujejo probleme. Programski jezik je način za izražanje programske paradigme.

Programske paradigme so hevristike, ki se uporabljajo za reševanje problemov. Programska paradigma analizira problem, čez specifičen pogled in na ta način formulira rešitev za dani problem, ki ga razdeli na manjše dele med katerimi definira razmerja.

Programske paradigme so na primer –proceduralno, objektno orientirano, funkcijsko, logično in istočasno programiranje–.



### 3.4 PROBLEMATIKA ZAČETKOV UČENJA PROGRAMIRANJA

Programiranje je veščina, ki potrebuje veliko vaje, ugotavljajo avtorji v članku [5]. Študenti pridobijo znanje programiranja z veliko programiranja oz. pisanjem kode. Praktični del je zelo pomemben za proces učenja programiranja. [5].

Nekatere osnove težave, katere srečajo programerji novinci [8]:

1. Inštalacija in nastavitve okolja za programiranje.
2. Uporaba urejevalnika besedil.
3. Razumevanje napisanih nalog oz. problemov in uporabe sintakse programskega jezika pri pisanju programske kode.
4. Razumevanje napak prevajalnika.
5. Razhroščevanje.

V preteklosti je bilo razvitih mnogo orodij, ki so nastala ravno z raziskovanja učenja programiranja, vendar mnoga od teh zahtevajo, da študenti pišejo celotne programe od začetka do konca.

Tudi začetniki, ki uspešno premagajo začetne ovire in se lotijo takojšnjega programiranja, imajo zelo slabo napisano in konstruirano programsko kodo. Pomagati novincem, pistati kvalitetno programsko kodo je časovno zelo zahtevno opravilo.

Težave programiranja se stopnjujejo ko se za učenje programiranja uporabljajo Objektno-orjentirani programski jeziki, sej ti zahtevajo visoko stopnjo abstraktnega razumevanja programskih konceptov in so načrtovani predvsem za zahtevne programerje.

### 3.5 PREDLAGANE REŠITVE

Pri samem vadenju programiranja je pomembno, da ob težavah, novinci dobijo čimprajšen odziv mentorja. V velikih razredih se to izkaže za zelo zahtevno.

### 3.6 OSNOVNI KONCEPTI PROGRAMIRANJA

### 3.7 PROGRAMIRANJE V OŠ

### 3.8 PROGRAMIRANJE V SŠ

## 4 SPLETNI PORTALI ZA UČENJE PROGRAMIRANJA

V začetku nas bo zanimalo kaj so spletni portali za učenje. Spoznali bomo, da poznamo različne kategorije spletnih portalov za posredovanje različnega znanja in veščin. Zanimali nas bodo predvsem spletni portali, ki učijo znanje programiranja.

Tradicionalni spletni portali v izobraževanju, kot so **moodle**, nikoli niso popolnoma izkoristili zmožnosti uporabe, ki jih ponujajo nove internetne in komunikacijske tehnologije. Večinoma so se uporabljale le kot podaljšana roka obstoječim metodam poučevanja. Uporabljale so se za objavo gradiv in spletno prijavo za oddajo nalog. Takšni sistemi ne zagotavljajo izboljšav kvalitete poučevanja programiranja [5].

Poglejmo primer spletnega portala, ki ga je izdelal avtor [8], in ima naslednje elemente.

1. Spletni portal za programiranja, ki omogoča naloge tipa "Zapolni prazna mesta".
2. Ogrodje za analizo, ki preverja kvaliteto in pravilnost, nalog, tipa "Zapolni prazna mesta".
3. Avtomatski sistem za dajanje povratnih informacij, ki sporoča prilagojena sporočila prevajalnika in formalni odziv študentom in njihovim mentorjem. Poročilo vsebuje kvaliteto napisanega programa, strukturo in pravilnost glede na programsko analizo.

### 4.1 PREDNOSTI SPLETNIH PORTALOV ZA UČENJE PROGRAMIRANJA.

Ena od prednosti dela z takšnim sistemom je ta, da novinci niso odvisni od mentorjevih uradnih govorilnih ur, pravtako tako lahko naloge opravljajo kadar koli [8].

### 4.2 PRIMER SISTEMSKE ARHITEKTURE SPLETNEGA PORTALA ZA UČENJE PROGRAMIRANJA

Primer systemske arhitekture kot so si zamislili avtorji [5]. Slika .. opis slike.

#### 4.2.1 ANALIZA PROGRAMSKE KODE

Dober odziv spletnega portala mora dati poročilo o pravilnosti programa in o kvaliteti [8].

Ogrodje (ang. framework) za analizo programske kode naj bi vsebovalo:

- Sintaktično ali semantično opozarjanje na napake ali napake kompilera. //To ima vgrajeno veliko spletnih mest.
- odziv na kvaliteto in pravilnost programske kode //Ali ga sistem nima ali je ta pomnankljiv. //Zgornje pomaga predvsem slabim učencem. //Večina sistemov izvaja statično analizo programske kode in tako ni v pomoč kakšne kvalitete je ta koda.
- Formalni odziv učitelja oz. komunikacija med učiteljem in učencem.

## 5 METODE IN STRATEGIJE PRI UPORABI SPLETNIH PORTALOV

Primer strategij in metod spletnega portala za učenje jave [8]:

- Scaffolding -> Gradnja študentovega znanja pri katerem pomaga mentorja, z svojim znanjem in izkušnjami.
- Bloomova taskonomija. Zakaj je pomembno vključevanje Bloomove taksonomije in kako jo vključujemo.
- Konstruktivizem: Aktivnost študentov pri gradnji znanja. Učenje z eksperimentiranjem. Problemski pristop.

Kaj od katerih metod predstavlja v uporabi spletnega portala ...:

- Spletni portal -> Scaffolding + Bloom
- Naloge narejene tako, da podpirajo konstruktivno metodo -> problemski pristop

### 5.1 UČNE STRATEGIJE

#### 5.1.1 MODEL AKTIVNEGA UČENJA

Pri **aktivnem učenju** je to, da učenci z lastno aktivnostjo ugotovijo, sami za sebe, kako nekaj deluje. Sami si morajo izmisliti primere, preiskusiti lastne veščine in reševati naloge, ki so jih

že ali jih še podo spoznali. Učenje je aktivno usvajanje, je gradnja idej in znanja. Za učenje mora biti posameznik aktivno vključen v gradnjo svojih lastnih mentalnih modelov.

Model aktivnega učenja je sestavljen s štirih korakov.

1. **Sprožilec** Je predstavljena naloga kot izziv za uvod v novo tematiko. //Gerlič -> Motivacija.
2. **Aktivnost** Študenti izvajajo aktivnost, ki jim je bila predstavljena v sprožilcu. Ta kora je lahko kratek ali lahko zavzame večju del učne ure. To je odvisno od vrste sprožilca in izobraževalnih ciljev.
3. **Diskusija** sledi po koncu aktivnosti, kjer se zbere zeloten razred, ne glede na obliko dela. V temo koraku študenti izpopolnijo koncepte in ideje, kod del konstruktivnega učnega procesa.
4. **Povzetek** je lahko izračen v različnih oblikah, kot so zaogrožene definicije, lahko so miselni vzorci ali povezav med temami, ki so jih obravnavali študenti in med drugimi temami, ki se navezujejo nanje.

Ko se ta model izkaže za primerne, ga lahko uporabimo v številnih učnih urah v različnih variacijah.

#### 5.1.2 UČENJE NA DALJAVO

### 5.2 TIPI NALOG

#### 5.2.1 ZAPOLNI PRAZNA MESTA

Tip nalog začenniku ponuja ogrodje programa, del programske kode, na katerem dijak usvoji novo znanje in/ali lahko uporablja že pridobljeno znanje.

## 6 KATEGORIZIRANJE SPLETNIH PORTALOV

### 6.1 VRSTA VSEBINE

Po hitrem pregledu izbranih spletnih portalov lahko ugotovimo, da je

## 6.2 PROGRAMSKI JEZIKI

# 7 OVREDNOTENJE IZBRANIH SPLETNIH PORTALOV IN NJIHOVE POSEBNOSTI

## 7.1 DOLOČITEV KRITERIJEV

# 8 MOŽNI NAČINI UPORABE SPLETNIH PORTALOV PRI PUKU

## LITERATURA IN VIRI

- [1] Martina Fefer, *Uporaba informacijske-komunikacijske tehnologije v osnovnih šolah s prilagojenim programom*, Univerza v Mariboru - Fakulteta za naravoslovje in matematiko, Maribor, 1999. Pridobljeno 4.4. 2016, iz <http://student.pfmb.uni-mb.si/~dgunze/diplomske/d2/s6.html>.
- [2] Gerlič, Ivan, *Sodobna informacijska tehnologija v izobraževanju*, DZS, Ljubljana, 2000.
- [3] Klemenčič M., *Uporaba računalniškega programa "Postani matematični mojster" pri pouku matematike.*, Pedagoška fakulteta, Ljubljana, 2011.
- [4] Anthony Robins, Janet Rountree, and Nathan Rountree, "Learning and Teaching Programming: A Review and Discussion" v *Computer Science education*, vol 13, No. 2, 2003, pp. 137 - 172.
- [5] S.C. Ng, S.O Choy, R. Kwan, S.F. Chan, "A Web-Based Environment to Improve Teaching and Learning of Computer Programming in Distance Education", *ICWL'05 Proceedings of the 4th international conference on Advances in Web-Based Learning*, 2005
- [6] L. Ma, J. D. Ferguson, M. Roper, I. Ross, M. Wood, "A web-based learning model for improving programming students' mental models", v *Proceedings of the 9th annual conference of the subject centre for information and computer sciences*, HE Academy, 2008 pp. 88-94.
- [7] O. Hazzan, T. Lapidot, N. Ragonis, *Guide to Teaching Computer Science*, Springer, 2011.
- [8] Nghi Truong, *A web-based programming environment for novice programmers*, Queensland University of Technology, Australia, 2007.