Oblikovanje edukativnih aplikacija za osnovnu školu

Iviček, Ivona

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Teacher Education / Sveučilište u Zagrebu, Učiteljski fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:147:021304

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

University of Zagreb Faculty of Teacher Education - Digital repository

SVEUČILIŠTE U ZAGREBU
UČITELJSKI FAKULTET
ODSJEK ZA UČITELJSKE STUDIJE

Ivona Iviček

OBLIKOVANJE EDUKATIVNIH APLIKACIJA ZA OSNOVNU ŠKOLU

Diplomski rad

Čakovec, travanj 2021.

SVEUČILIŠTE U ZAGREBU
UČITELJSKI FAKULTET
ODSJEK ZA UČITELJSKE STUDIJE

Ivona Iviček

OBLIKOVANJE EDUKATIVNIH APLIKACIJA ZA OSNOVNU ŠKOLU

Diplomski rad

Mentor rada:
izv. prof. dr. sc. Predrag Oreški

Čakovec, 2021.

SADRŽAJ

1. UVOD 1

2. KURIKULUM NASTAVNOG PREDMETA INFORMATIKA (u nižim razredima) 3

2.1. Informacije i digitalna tehnologija 3

2.2. Računalno razmišljanje i programiranje 4

2.3. Digitalna pismenost i komunikacija 5

2.4. e-Društvo 5

3. PROGRAMIRANJE 6

4. SCRATCH 7

5. MICROSOFT VISUAL STUDIO 8

6. VISUAL BASIC 9

7. SOFTVERSKO INŽENJERSTVO 10

7.1. Aplikacije 11

7.2. Procesi razvoja softvera i softverski modeli 12

7.2.1. Model vodopada 13

7.2.2. Model evolucijskog razvoja 15

7.2.3. Model inkrementalnog razvoja 15

7.2.4. Model usmjeren na ponovnu upotrebu 16

8. OPIS PROGRAMA 17

8.1. Utvrđivanje zahtjeva i definicija 17

8.2. Oblikovanje sustava i softvera (dizajn) 17

8.3. Implementacija i testiranje modula (programiranje) 24

8.4. Testiranje sustava 31

8.5. Puštanje u rad i održavanje 31

9. ZAKLJUČAK 33

10. LITERATURA 34

11. PRILOZI I DODATCI 36

12. IZJAVA O IZVORNOSTI RADA 37

SAŽETAK

Edukativna aplikacija „Učimo informatiku - SCRATCH“ izrađena je prema Kurikulumu nastavnog predmeta informatika, po jednoj od četiri domene, Računalno razmišljanje i programiranje za drugi razred osnovne škole. Svrha izrade ove aplikacije je olakšati učenicima samostalno usvajanje novog nastavnog sadržaja u školi ili od kuće. Visual Studio 2019 omogućuje izradu aplikacija za radnu površinu, web, cloud i mobilne uređaje, a programski jezik Visual Basic omogućuje stvaranje korisnih programa i aplikacija poznavanjem samo nekoliko osnovnih naredbi. Softversko inženjerstvo se bavi svim aspektima proizvodnje softvera od ranih faza utvrđivanja zahtjeva do održavanja sustava nakon što je ušao u upotrebu, a model za softverski proces je idealiziran prikaz softverskog procesa, kojim se određuje poželjan način odvijanja i međusobnog povezivanja osnovnih aktivnosti. Ova edukativna aplikacija je napravljena u programskom jeziku Visual Basic i opisana kroz faze vodopadnog modela procesa programiranja, a učenicima omogućuje samostalnu obradu novog nastavnog sadržaja. Učenici drugog razreda osnovne škole uz pomoć ove aplikacije upoznat će i naučiti neke od osnova programiranja u Scratch-u na način da prvo ponove ono što bi trebali znati, kako bi usvojili novi nastavni sadržaj. Nakon ponavljanja slijedi učenje kroz nekoliko stranica (obrazaca) na kojima se mogu zadržati onoliko vremena koliko im je potrebno da bi sve usvojili, a na kraju svoje znanje utvrđuju kroz nekoliko različitih vrsta pitanja. Sadržaj aplikacije prati nastavne jedinice radnog udžbenika informatike za drugi razred osnovne škole, *„e-Svijet 2“* te uvodi učenike u svijet programiranja putem besplatnog vizualnog programskog jezika Scratch, koji je izrađen za sve, ali prvenstveno za djecu i učenike od 8 do 16 godina.

Ključne riječi: edukativna aplikacija, Scratch, programsko inženjerstvo, Visual Basic

SUMMARY

The educational application "Learning Informatics - SCRATCH" is made according to the Curriculum of the subject Informatics, in one of four domains, Computer Thinking and Programming for the second grade of primary school. The purpose of creating this application is to make it easier for students to independently adopt new teaching content at school or from home. Visual Studio 2019 allows you to create applications for desktop, web, cloud and mobile devices, and the programming language Visual Basic allows you to create useful programs and applications by knowing just a few basic commands. Software engineering deals with all aspects of software production from the early stages of requirements determination to system maintenance after it is put into use, and the software process model is an idealized representation of the software process, which determines the desired way of development and interconnection of basic activities. This educational application is made in the Visual Basic programming language and described through the stages of the waterfall model of the programming process, and allows students to independently process new teaching content. With the help of this application, second grade elementary school students will get to know and learn some of the basics of programming in Scratch by first repeating what they should know, in order to adopt new teaching content. Repetition is followed by learning through several pages (forms) on which they can stay as long as they need to adopt everything, and in the end they determine their knowledge through several different types of questions. The content of the application follows the teaching units of the working textbook of informatics for the second grade of primary school, "e-World 2" and introduces students to the world of programming through the free visual programming language Scratch, which is designed for everyone, but primarily for children and students aged 8 to 16.

Key words: educational application, Scratch, software engineering, Visual Basic

1. UVOD

Tehnologija je postala neizbježan dio svakodnevice. Sve više se koristi u školama i primjenjuje u nastavi. Škole su sve opremljenije digitalnom tehnologijom zbog uvođenja predmeta Informatike u niže razrede. Neke škole su osposobile informatičku učionicu s računalima, neke su dobile tablete za svakog učenika u razredu, a velika većina škola u razredima posjeduje barem jedno računalo i projektor. Pametne ploče još nisu toliko zastupljene, ali s vremenom ne sumnjam da će u potpunosti zamijeniti klasičnu zelenu ploču s kredom. Kada je učenik spriječen doći u školu, na neki način mora nadoknaditi propušteni nastavni sadržaj. Ova izrađena aplikacija mu to omogućuje. U bilo kojem trenutku može pristupiti programu, ponoviti, naučiti i utvrditi nastavni sadržaj samostalno od kuće. Aplikacija je jednostavna i namijenjena je učenicima drugog razreda osnovne škole za predmet Informatike i programiranje u Scratch-u. Učenici će pomoću nje moći ponoviti sadržaj bitan za svladavanje novog nastavnog sadržaja, naučit će nešto novo i moći to sve na kraju ponoviti kroz par pitanja.

U nastavku rada opisat će se i proći sve domene kurikuluma nastavnog predmeta informatike za niže razrede osnovne škole te što se u kojem razredu očekuje da će učenici savladati. Definirat će se programiranje i što je Scratch u kojem se djeca vjerojatno prvi put susreću s programiranjem. Nakon toga će biti opisan Microsoft Visual Studio s nekim od svojih značajki, kao i što omogućuje Visual Basic te kako izgleda. Definirat će se pojam softversko inženjerstvo, navesti neke faze i aktivnosti softverskog procesa. Bit će navedene i ukratko objašnjene neke od vrsta aplikacija te detaljnije opisani softverski modeli poput modela vodopada, modela evolucijskog razvoja, modela inkrementalnog razvoja i modela usmjerenog na ponovnu upotrebu. Nakon toga slijedi opis aplikacije po svakoj od faza vodopadnog modela po kojem je rađena. Razlog zbog kojeg se krenulo u izradu ove edukativne aplikacije je činjenica da smo svi u posljednjih godinu dana bili prisiljeni iskusiti online, tj. način rada i učenja od kuće. Smatram da smo na taj način rada bili nespremni jer nismo krenuli postupno, već „preko noći“. Mnogi se učitelji, profesori, nastavnici i ostali predavači sigurno nisu odmah uspjeli snaći i omogućiti učenicima dovoljno kvalitetno odrađen sat, obradu nastavne jedinice, povratnu informaciju o radu i ostalo što su u svakodnevnoj nastavi u školi imali. Zbog toga je cilj i svrha izrade ove aplikacije olakšati učenicima samostalno učenje novog nastavnog sadržaja od kuće i dodatno ih zainteresirati za programiranje i učenje

informatike. Aplikacija se kasnije može nadograditi i upotpuniti novim nastavnim cjelinama i jedinicama, dodatnim pitanjima i prikazom konačnog rezultata rješavanja kviza.

2. KURIKULUM NASTAVNOG PREDMETA INFORMATIKA (u nižim razredima)

Četiri su domene kojima se realiziraju ciljevi predmeta Informatika koja se održava 70 sati godišnje u svim razredima: Informacije i digitalna tehnologija, Računalno razmišljanje i programiranje, Digitalna pismenost i komunikacija, te e-Društvo. (MZO, 2018.)

Slika 1. Četiri domene kurikuluma. Izvor: (MZO, 2018.)

2.1. Informacije i digitalna tehnologija

U domeni Informacije i digitalna tehnologija važno je poznavati temeljne koncepte rada računala i pojedinih uređaja, obrasce pohrane podataka te obilježja i načine prijenosa digitalnih informacija kako bi se razvile sposobnosti odabira i uporabe primjerene digitalne tehnologije i programa za obradu i predstavljanje podataka.

Nakon prvog razreda učenik će moći prepoznati digitalnu tehnologiju, razlikovati oblike digitalnih sadržaja, uređaja i postupaka za njihovo stvaranje te će moći komunicirati s poznatim osobama uz pomoć učitelja u sigurnom digitalnom okruženju.

Nakon drugog razreda učenik objašnjava ulogu programa u uporabi računala te uz pomoć učitelja prepoznaje internet kao izvor nekih usluga i podataka te pretražuje preporučene sadržaje.

Nakon trećeg razreda učenik se koristi simbolima za prikazivanje podataka, objašnjava i analizira jednostavne hardverske/softverske probleme i poteškoće koji se mogu dogoditi tijekom njihove uporabe.

Nakon četvrtog razreda učenik će moći objašnjavati koncept računalne mreže, razlikovati mogućnosti koje one nude za komunikaciju i suradnju, analizirati čimbenike koji razlikuju ljude od strojeva te proučavati interakcije čovjek-stroj i moći će se koristiti simbolima za prikazivanje podataka, analizirati postupak prikazivanja te vrednovati njegovu učinkovitost. (MZO, 2018.)

2.2. Računalno razmišljanje i programiranje

Računalno razmišljanje temeljni je pristup kojim se razvija sposobnost rješavanja problema i programiranja. Pritom je naglasak na usvajanju procesa stvaranja aplikacije od početne ideje do konačnoga proizvoda, a ne isključivo na usvajanju sintakse i semantike programskoga jezika. Aktivnosti i sadržaji ishoda iz domene Računalno razmišljanje i programiranje razvijaju inovativnost, stvaralaštvo i poduzetnost te daju vrijedna znanja koja se mogu ugraditi u budući profesionalni život.

Nakon prvog razreda učenik može rješavati jednostavan logički zadatak te pratiti i prikazivati slijed koraka za rješavanje nekog jednostavnog zadatka.

Nakon drugog razreda učenik analizira niz uputa koje izvode jednostavan zadatak, a ako je potrebno ispravlja i pogrešan redoslijed te stvara niz uputa u kojemu upotrebljava ponavljanje.

Nakon trećeg razreda učenik stvara program korištenjem vizualnoga okruženja u kojem se koristi slijedom koraka, ponavljanjem i odlukom te uz pomoć učitelja vrednuje svoje rješenje i slaže podatke na koristan način.

Nakon četvrtog razreda učenik može stvarati program korištenjem vizualnog okruženja u kojem koristi slijed, ponavljanje, odluku i ulazne vrijednosti te može rješavati složenije logičke zadatke s uporabom ili bez uporabe računala. (MZO, 2018.)

2.3. Digitalna pismenost i komunikacija

Domena Digitalna pismenost i komunikacija usko je povezana s ostalim domenama i daje temeljne digitalne kompetencije koje su neophodne za kvalitetnu primjenu tehnologije pri obavljanju svakodnevnih obveza, ali i za stjecanje kompetencija iz ostalih domena. Obilježje je te domene i razvijanje otvorenosti prema novim tehnološkim dostignućima u području informacijske i komunikacijske tehnologije.

Nakon prvog razreda učenik uz podršku učitelja vrlo jednostavnim radnjama izrađuje jednostavne digitalne sadržaje, koristi se predloženim programima i digitalnim obrazovnim sadržajima.

Nakon drugog razreda učenik prema savjetima učitelja odabire uređaj i program za jednostavne školske zadatke, uz podršku i pomoć učitelja izrađuje digitalne radove kombiniranjem različitih oblika sadržaja, surađuje i komunicira s poznatim osobama u sigurnome digitalnom okruženju.

Nakon završetka trećeg razreda učenik samostalno odabire uređaj i program iz skupa predloženih te procjenjuje načine njihove uporabe. Prema uputama izrađuje jednostavne digitalne radove, koristi se sigurnim digitalnim okruženjem za komunikaciju u suradničkim aktivnostima te razlikuje uloge i aktivnosti koje zahtijeva suradničko online okruženje.

Nakon četvrtog razreda učenik može odabrati prikladan program za zadani zadatak, preporučiti ga drugima te istraživati mogućnosti sličnih programa, može osmišljavati plan izrade digitalnoga rada, izrađivati i vrednovati rad te u suradničkome online okruženju zajednički planirati i ostvariti jednostavne ideje. (MZO, 2018.)

2.4. e-Društvo

Domena e-Društvo temelji se na činjenici da živimo u informacijskome društvu u kojemu je digitalna tehnologija neizostavni dio života. Teme kao što su područje sigurnosti na mreži, zaštita podataka, elektroničko nasilje i briga o svojemu digitalnom ugledu razvijaju potrebne vještine i stavove nužne za odgovorne, kompetentne, kreativne i pouzdane sudionike digitalnoga društva.

Nakon završenog prvog razreda učenik se pažljivo i odgovorno koristi informacijskom i komunikacijskom opremom i štiti svoje osobne podatke. Primjenjuje zdrave navike ponašanja tijekom rada na računalu i prihvaća preporuke o količini vremena provedenoga za računalom.

Nakon drugog razreda učenik prepoznaje i opisuje neke poslove koji se koriste informacijskom i komunikacijskom tehnologijom, koristi se e-uslugama u području odgoja i obrazovanja, analizira neke opasnosti koje mogu nastupiti pri uporabi računala i interneta te pravilno na njih reagira, odgovorno se ponaša pri korištenju sadržajima i uslugama na internetu radi zaštite osobnih podataka i digitalnoga ugleda.

Nakon trećeg razreda učenik primjenjuje preporuke o preraspodjeli vremena u kojemu se koristi digitalnom tehnologijom za učenje, komunikaciju i zabavu te primjenjuje zdrave navike. Primjereno reagira na svaku opasnost/neugodnost u digitalnome okruženju te štiti svoje i tuđe osobne podatke.

Nakon četvrtog razreda učenik istražuje ograničenja uporabe računalne tehnologije te primjenjuje upute za očuvanje zdravlja i sigurnost pri radu s računalom. Analizira široki spektar poslova koji zahtijevaju znanje ili uporabu informacijske i komunikacijske tehnologije.

(MZO, 2018.)

3. PROGRAMIRANJE

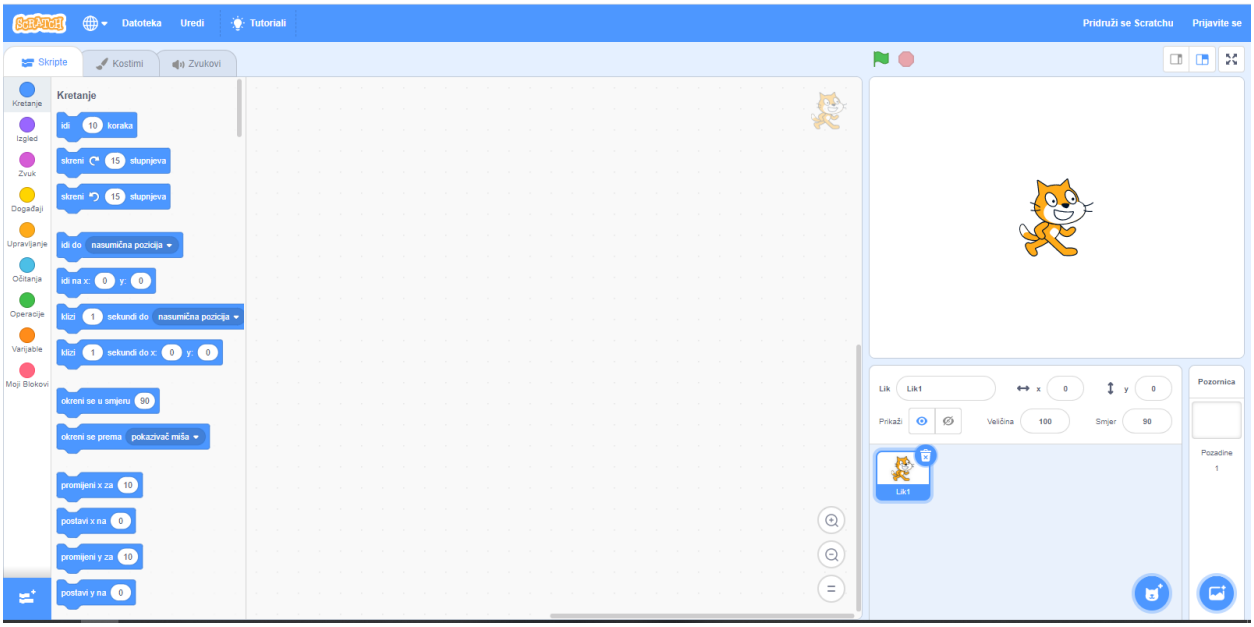
Programiranje u užem smislu predstavlja proces u kojem se pišu instrukcije za računalo kako bi uz pomoć njih ono moglo obavljati određene zadatke. Programiranje se može definirati i u širem smislu, a onda je to proces koji uključuje analizu problema i specificiranje zahtjeva, oblikovanje rješenja, kodiranje, testiranje, izradu programske i korisničke dokumentacije, održavanje, usavršavanje i usklađivanje s novim zahtjevima. (P. Oreški - Predavanje) Skup naredbi i uputa se najčešće naziva kod, a osoba koja piše kod, naziva se programer.

(Budojević i Kanić 2019.)

4. SCRATCH

Scratch je besplatni vizualni programski jezik izrađen za djecu i učenike od 8 do 16 godina, ali mogu ga i koriste ga ljudi svih dobnih skupina. Koristi se u više od 150 zemalja diljem svijeta i prilagođen je za više od 60 jezika. Milijuni ljudi koriste Scratch u raznim sredinama, u školama, domovima, muzejima, knjižnicama i raznim drugim mjestima. (scratch.mit.edu) Nastao je kao projekt Lifelong Kindergarten grupe koja je dio MIT (Massachusetts Institute of Technology) Media Lab-a. (Đurđević, 2013.) Scratch služi kako pomoć za razvijanje osnovnih vještina koje autori programa smatraju bitnim za 21. stoljeće, kao što su kreativno razmišljanje, jasna komunikacija, sistematična analiza, rješavanje problema i kontinuirano učenje, ali i da djecu na zabavan način uvedu u programiranje. Programskom sučelju se može pristupiti preko web preglednika ili se može preuzeti aplikacija na računalo koja može raditi bez mreže. (scratch.mit.edu)

U Scratchu se rabe naredbene strukture u obliku grafičkih programskih blokova, pa se na taj način eliminira mogućnost sintaksnih pogrešaka npr. kada se zaboravi staviti točka, zarez ili zagrade. (Wilson i Moffat, 2010.) Korisnik mišem povlači željene blokove u radni prostor koji se mogu prema obliku slagati kao „LEGO“ kockice ili dijelovi slagalice (ispod, iznad ili unutar bloka). Blokovi su podijeljeni u grupe ili kategorije različitih boja kako bi lakše pronašli željeni blok. (Blagus i sur. 2020., e-svijet 2) Na taj način dijete može bez napora samostalno izraditi programe, računalne igre, animacije, tekst, interaktivne priče i dr. Osim toga kada nauče raditi u Scratch-u lakše će razumjeti i naučiti teže programske jezike poput Python-a i Java-e. (Wilson i Moffat, 2010.)



Slika 2. SCRATCH. Izvor: (https://scratch.mit.edu/projects/editor/?tutorial=getStarted)

5. MICROSOFT VISUAL STUDIO

Visual Studio 2019 novo je izdanje Microsoftovog razvojnog okruženja (eng. Integrated Development Environment = IDE), koje omogućuje izradu aplikacija za radnu površinu, web, cloud i mobilne uređaje. (Del Sole, 2019)

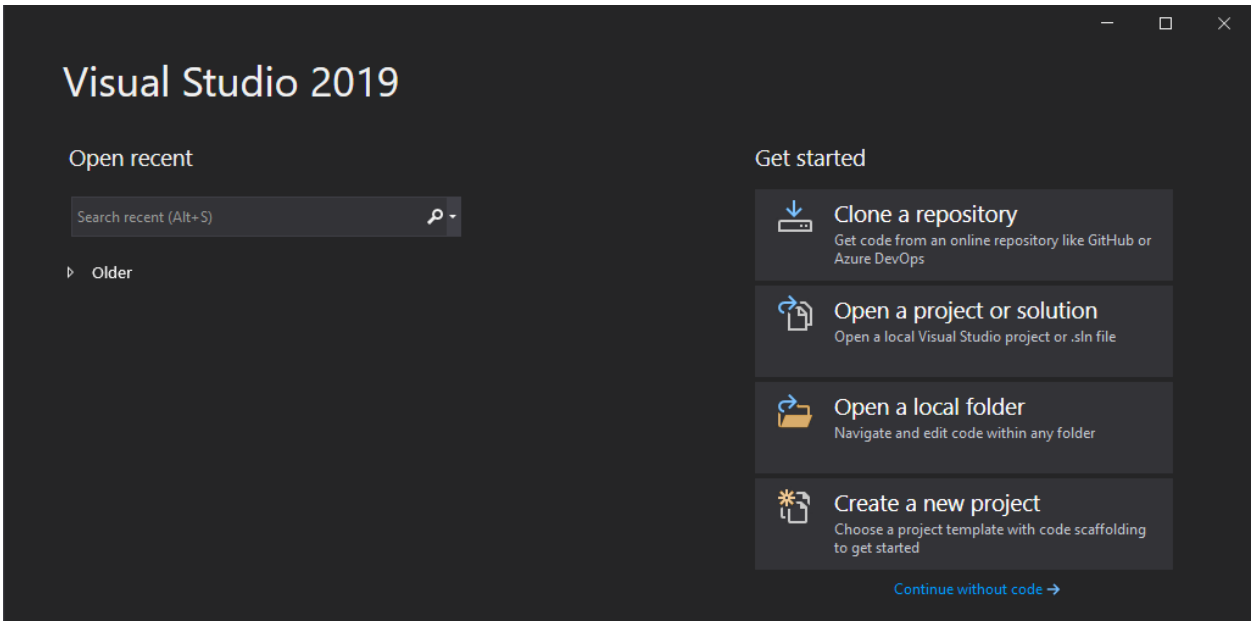
Neke od popularnih značajki Visual Studija koje pomažu boljem i produktivnijem razvoju softvera su:

Podcrtavanje pogreške (eng. squiggles) su valovita podcrtavanja koja upozoravaju na pogreške ili potencijalne probleme prilikom pisanja koda. Ovaj vizualni trag omogućava da se odmah riješi problem bez čekanja da se pogreška otkrije prilikom pokretanja programa. Ako se pokazivač zadrži iznad podcrtanog dijela koda, prikazat će se dodatne informacije o pogrešci. Na lijevoj margini može se pojaviti i žarulja s radnjama, poznata pod nazivom „brze radnje“ (eng. Quick Actions), kako bi se ispravila pogreška. Refaktoriranje uključuje operacije poput pametnog preimenovanja varijabli, izdvajanje jednog ili više redaka koda u novu metodu, promjenu redoslijeda parametara metode i još mnogo toga. IntelliSense je pojam za skup značajki koji

predviđa, predlaže, a u nekim slučajevima i napiše male dijelove koda.. Značajke IntelliSense razlikuju se ovisno o jeziku. Osim ovih postoje i mnoge druge značajke.

Visual Studio podržava različite programske jezike, C#, C++, F#, TypeScript, JavaScript, Python , Query Language, Visual Basic i XAML, a podrška za ostale programske jezike (Ruby, SQL, HTML, CSS i druge) dostupna je instalacijom jezičnih servisa.

(docs.microsoft.com - visualstudio)



Slika 3. Visual Studio 2019

6. VISUAL BASIC

Programski jezik Visual Basic razvila je tvrtka Microsoft, a zamišljen je kao alat za brzu izradu programa. Riječ „Visual“ odnosi se na metodu korištenu za stvaranje grafičkog korisničkog sučelja (eng. Graphical User Interface = GUI), a riječ „Basic“ označuje programski jezik BASIC (eng. Beginners All-Purpose Symbolic Instruction Code). Sadrži više stotina izraza, naredbi i funkcija, od kojih je najveći dio izravno povezan sa Windows grafičkim sučeljem. Omogućuje stvaranje korisnih programa i aplikacija poznavanjem samo nekoliko osnovnih naredbi. (Microsoft Press, 1999.) Kod programskog jezika Basic proces programiranja se svodio na pisanje koda, tj. svaki objekt koji se pojavio na zaslonu, morao se posebno programirati u kodu pa je to stvaralo

veliki broj kodnih linija koje su se ponavljale. Kako bi se taj posao olakšao, objektno orijentirani programeri olakšali su taj posao izradom određenog broja objekata, koji su se do tada najviše koristili u programiranju. Tako je objekt postao centralni dio programiranja, a kod dio programa koji dodatno opisuje i međusobno povezuje objekte. (Damjanović i Katanić, 2014.)

Visual Basic je objektno orijentiran, a programiranje se izvršava događajima, tj. ako imamo objekt Gumb (eng. Button) koji sadrži metodu za ispis „Pozdrav!“, samo prilikom pritiska na taj gumb će se na ekranu ispisati „Pozdrav!“. Objekti imaju tri skupine značajki: Svojstva (Properties) kojima se može mijenjati njihova svojstva, (npr. boja pozadine, slova, lokacija, vrsta slova,...); Metode (Methods) su potprogrami primjenjivi na objektima (vrše neku akciju s objektom); Događaji (Events) su također potprogrami, ali koji opisuju događaje nad objektima poput klika (Click), dvostrukog klika (DBL Click), pomaka miša (Mouse Move) i sl.. Objekte možemo umetnuti po principu dovuci i ostavi (eng. Drag’n’Drop) tako da ih dovučemo i ostavimo na radnoj površini/formi (eng. Form), a prilikom dvostrukog klika na objekt otvara se prozor za pisanje koda gdje se upisuju metode koje će se izvršiti kada se klikne ili napravi određena radnja na tom objektu. Svaki objekt se može promijeniti željene parametre i izgled u izborniku svojstva (eng. Properties).

(Microsoft Press, 1999.)

7. SOFTVERSKO INŽENJERSTVO

Softversko inženjerstvo (eng. Software Engineering) je disciplina koja se bavi modelima, metodama i alatima koji su potrebni da bi na što učinkovitiji način mogli proizvoditi što kvalitetnije softverske produkte. Ne postoje univerzalne metode i tehnike softverskog inženjerstva koje su prikladne za sve sustave i sve tvrtke. Ujedno je i sustavni pristup proizvodnji softvera koji uzima u obzir praktična pitanja troškova, rasporeda i pouzdanosti, kao i potrebe kupaca i proizvođača softvera. Počelo se razvijati krajem 60-ih godina 20. stoljeća pojavom računala treće generacije kada se stvorila potreba za složenijim softverom.

(Manger, 2016.)

Softversko inženjerstvo se bavi svim aspektima proizvodnje softvera od ranih faza utvrđivanja zahtjeva do održavanja sustava nakon što je ušao u upotrebu. U ovoj su definiciji dvije ključne fraze:

1. Inženjerska disciplina gdje inženjeri čine da stvari funkcioniraju. Primjenjuju teorije, metode, i alate tamo gdje su prikladni. Međutim, koriste ih selektivno i uvijek pokušavaju otkriti rješenja za probleme čak i kad nema primjenjivih teorija i metoda.

2. Svi aspekti proizvodnje softvera uključuju i aktivnosti kao što je upravljanje softverskim projektima i razvoj alata, metoda, i teorije za podršku proizvodnje softvera.

Softverski proces je slijed aktivnosti koji vodi do izrade softverskog proizvoda. Četiri su osnovne zajedničke aktivnosti svim softverskim procesima.

1. Utvrđivanje zahtjeva (specifikacija softvera) gdje kupci i inženjeri definiraju softver koji treba proizvesti i ograničenja u njegovom radu.

2. Razvoj softvera (implementacija), gdje je softver dizajniran i programiran.

3. Provjera valjanosti softvera (verifikacija i validacija), gdje se softver provjerava da li je to što kupac zahtijeva.

4. Evolucija (održavanje) softvera, gdje je softver modificiran tako da odražava promjenu kupca i zahtjeve tržišta.

(Sommerville, 2016.)

7.1. Aplikacije

Možda najznačajniji čimbenik u određivanju metoda i tehnika softverskog inženjerstva su načini razvijanja aplikacija. Postoji mnogo različitih vrsta aplikacija, uključujući:

1. Samostalne aplikacije koje se izvode na lokalnom računalu. Sadrže svu potrebnu funkcionalnost i ne trebaju biti povezana s mrežom.

2. Interaktivne aplikacije temeljene na transakcijama su aplikacije koje se izvršavaju na udaljenom računalu i kojima korisnici pristupaju s vlastitog računala ili terminala. Tu spadaju web aplikacije

poput aplikacija za e-trgovinu gdje se može komunicirati s udaljenim sustavom za kupnju robe i usluga te za posebne usluge zasnovane na „cloud-u“, poput razmjene pošte i fotografija.

3. Ugrađeni upravljački sustavi su softverski upravljački sustavi koji kontroliraju i upravljaju hardverskim uređajima. Primjeri ugrađenih sustava uključuju softver u mobitelu (mobitel), softver u mikrovalnoj pećnici za kontrolu postupka kuhanja i drugi.

4. Sustavi serijske obrade su poslovni sustavi koji su dizajnirani da obrađuju podatke u velikim skupinama. Oni obrađuju velik broj pojedinačnih ulaza da stvore odgovarajuće izlaze. Primjeri ovog sustava uključuje periodične sustave za naplatu, kao što su telefonski sustavi za naplatu ili sustavi za isplatu plaća.

5. Sustavi za zabavu su sustavi koji su prvenstveno za osobnu upotrebu i koji su namijenjeni zabavi korisnika. Većina tih sustava su igre raznih vrsta.

6. Sustavi za modeliranje i simulaciju su sustavi koje su razvili znanstvenici i inženjeri za modeliranje fizičkih procesa ili situacija, koje uključuju mnoge, odvojene, interaktivne objekte.

7. Sustavi za prikupljanje podataka su sustavi koji prikupljaju podatke iz svog okruženja pomoću skupa senzora koji šalju te podatke drugim sustavima na obradu. Softver mora komunicirati sa senzorima i često se instalira u „neprijateljskom“ okruženju poput motora ili na udaljenom mjestu.

8. Sustavi sustava su sustavi koji se sastoje od niza drugih softverskih sustava. Neki od njih mogu biti generički softverski proizvodi, kao što je program proračunskih tablica.

Granice između ovih vrsta sustava su nejasne. Za svaku vrstu sustava koriste se različite tehnike softverskog inženjerstva jer softver ima sasvim različite karakteristike.

(Sommerville, 2016.)

7.2. Procesi razvoja softvera i softverski modeli

Softverski proces je skup aktivnosti i pripadnih rezultata čiji je cilj razvoj i evolucija softvera. Model procesa razvoja softvera je idealiziran prikaz softverskog procesa, kojim se određuje poželjan način odvijanja i međusobnog povezivanja osnovnih aktivnosti.

Modeli se razlikuju po načinu odvijanja i međusobnog povezivanja osnovnih aktivnosti, a zajednička osobina svih modela za softverski proces je da se oni u većoj ili manjoj mjeri zasnivaju na otprilike istim osnovnim aktivnostima, a to su:

- utvrđivanje zahtjeva = analiziraju se zahtjevi korisnika i utvrđuje se što softver treba raditi.
- oblikovanje = oblikuje se građa sustava, način rada njegovih dijelova te sučelje između dijelova, tj. projektira se rješenje koje određuje kako će softver raditi.
- implementacija = oblikovano rješenje realizira se uz pomoć raspoloživih programskih jezika i alata.
- verifikacija i validacija = provjerava se radi li softver prema specifikaciji (radi li ono što korisnik želi).
- evolucija (održavanje)= nakon uvođenja u upotrebu, softver se dalje popravlja, mijenja i nadograđuje, u skladu s promijenjenim potrebama korisnika.

(Manger, 2016.)

7.2.1. Model vodopada

Model vodopada (eng. Waterfall model) nastao je u ranim 70-im godinama 20. stoljeća, kao neposredna analogija s procesima iz drugih inženjerskih struka. Model je dobio ime zbog oblika dijagrama. Aktivnosti se odvijaju kao faze slijedno jedna iza druge, a svaka faza daje rezultat koji „teče“ po vodopadu i predstavlja polazište za iduću fazu.

Glavne faze modela vodopada izravno odražavaju temeljni razvoj aktivnosti:

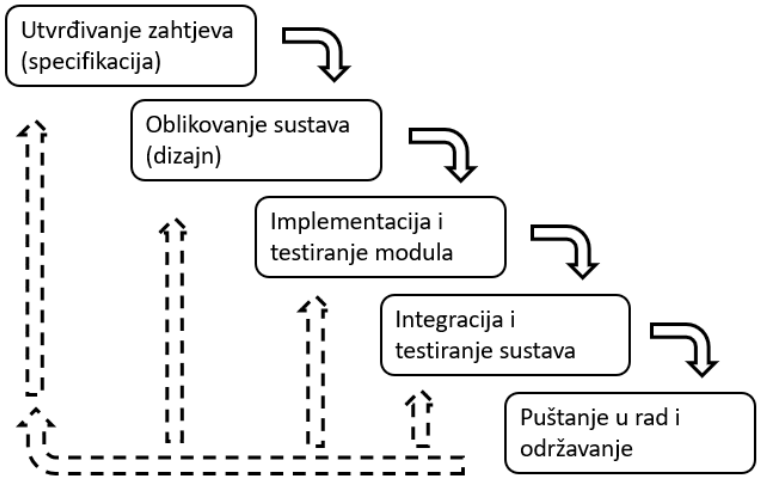
1. Utvrđivanje zahtjeva - usluge sustava, ograničenja i ciljevi utvrđuju se i detaljno definiraju konzultacijama s korisnicima i služe kao specifikacija sustava.

2. Oblikovanje sustava i softvera (dizajn) - dizajn softvera uključuje prepoznavanje i opisivanje temeljnih apstrakcija softverskog sustava i njihove odnose.

3. Implementacija i testiranje modula (programiranje) - tijekom ove faze realizira se dizajn softvera kao skup programa ili programskih cjelina. Jedinstveno testiranje uključuje provjeru zadovoljava li svaka jedinica svoje specifikacije.

4. Integracija i testiranje sustava - pojedinačne programske cjeline ili programi integrirani su i testirani kao cjelovit sustav kako bi se osiguralo da su zahtjevi softvera ispunjeni. Nakon testiranja, softverski sustav se isporučuje klijentu.

5. Puštanje u rad i održavanje - sustav se instalira i stavlja u praktičnu upotrebu. Održavanje uključuje ispravljanje pogrešaka koje ranije nisu bile otkrivene, poboljšanje primjene sistemskih jedinica i poboljšanje usluga sustava kako se otkrivaju novi zahtjevi.



Slika 4. Vodopadni model. Izvor: (Manger, 2016.)

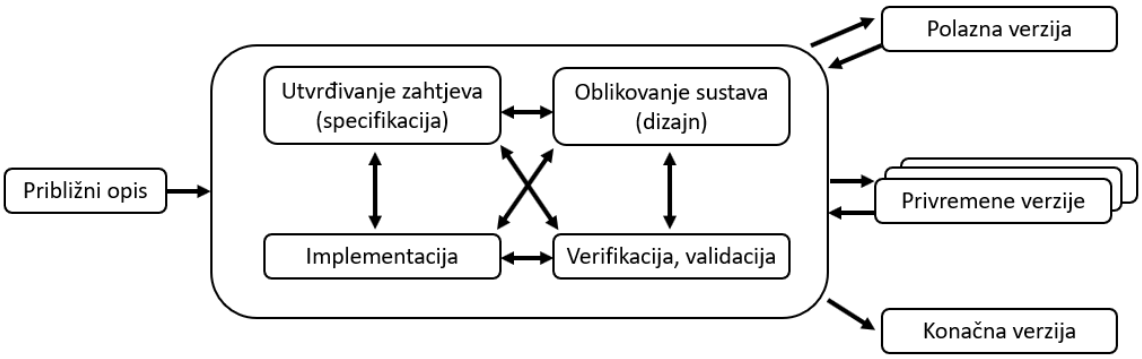
U principu, rezultat svake faze je jedan ili više odobrenih dokumenata. Sljedeća faza ne bi trebala započeti dok prethodna faza ne završi, ali u praksi se ove faze preklapaju. Tijekom dizajna, identificiraju se problemi sa zahtjevima. Tijekom kodiranja, nalaze se problemi s dizajnom i tako dalje. Proces softvera nije jednostavan linearni model, ali uključuje povratne informacije iz jedne u drugu fazu.

Model slapa omogućuje detaljno planiranje cijelog softverskog procesa i podjelu poslova na velik broj suradnika te se lako može utvrditi stanje u kojem se proces trenutno nalazi. Problem ovog modela je što je faze teško razdvojiti pa dolazi do naknadnog otkrivanja pogrešaka i nepoželjnog vraćanja u prethodne faze. Osim toga, proces je spor pa se može dogoditi da u trenutku puštanja u rad, sustav već bude neažuran i zastario.

(Sommerville, 2016.), (Manger, 2016.)

7.2.2. Model evolucijskog razvoja

Model evolucijskog razvoja (eng. Evolutionary development) nastao je kao protuteža vodopadnom modelu. Ovdje se na osnovi približnog opisa problema razvija polazna verzija sustava koja se pokazuje korisniku. Na temelju korisnikovih primjedbi, polazna verzija se poboljšava i ponovo pokazuje korisniku dok se ne dobije konačna verzija sustava. Unutar svakog ponavljanja/obnavljanja osnovne se aktivnosti obavljaju istovremeno i ne daju se razdvojiti.



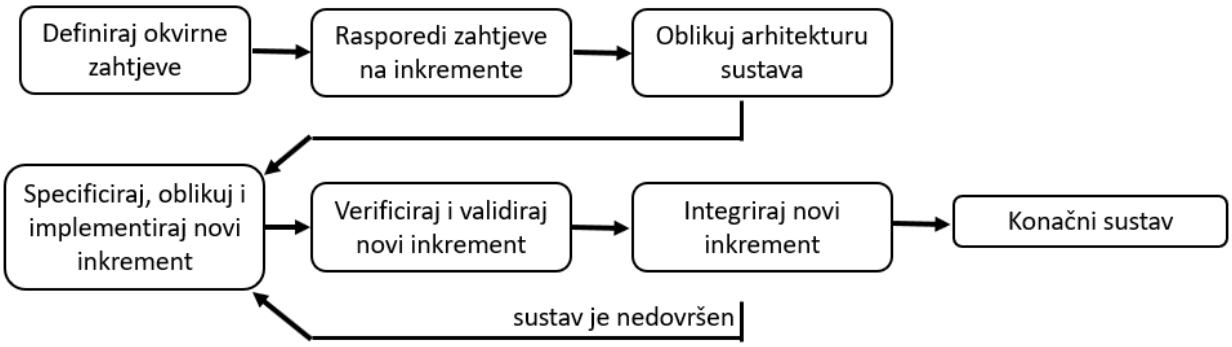
Slika 5. Model evolucijskog razvoja. Izvor: (Manger, 2016.)

(Manger, 2016.)

7.2.3. Model inkrementalnog razvoja

Model inkrementalnog razvoja (eng. Incremental development) sličan je modelu evolucijskog razvoja, a može se shvatiti kao hibrid vodopadnog i evolucijskog modela razvoja. Sustav se i ovdje razvija nizom ponavljanja i promjena, ali umjesto promjene već realiziranog dijela sustava, ovdje se dodaje sasvim novi dio - inkrement.

(Manger, 2016.)



Slika 6. Model inkrementalnog razvoja. Izvor: (Manger, 2016.)

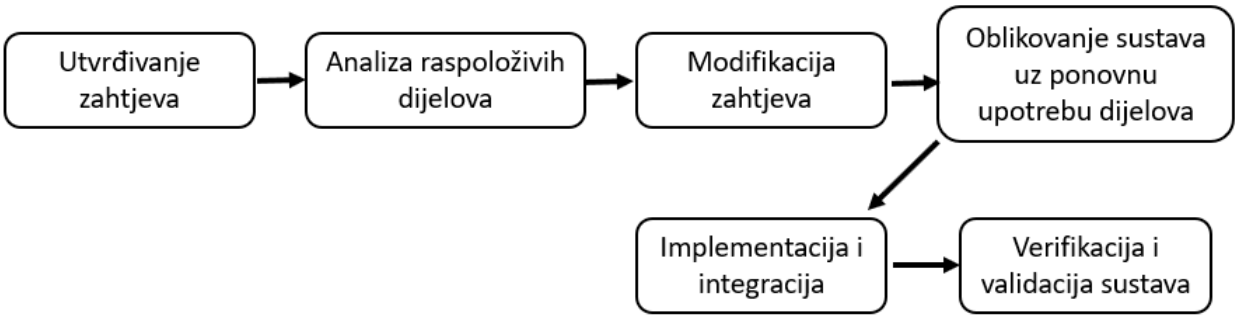
Rijetko se unaprijed radi cjelovito rješenje problema, ali prema rješenju se dolazi nizom koraka te vraćanjem unatrag kad se shvati da je napravljena pogreška. Postupni razvoj softvera jeftiniji je i lakši za izradu promjene u softveru kako se razvija. Korisnik može procijeniti sustav u relativno ranoj fazi razvoja da bi vidio isporučuje li ono što se traži. Ako ne, tada mora promijeniti samo trenutni inkrement, a moguće i definirati nove funkcionalnosti za kasnije promjene.

(Sommerville, 2016.)

7.2.4. Model usmjeren na ponovnu upotrebu

Model usmjeren na ponovnu upotrebu (eng. Reuse-oriented development) polazi od pretpostavke da već postoje gotove i upotrebljive softverske cjeline. Ljudi koji rade na projektu znaju za dizajn ili kod koji je sličan onome koji se traži pa ih prema potrebi modificiraju i uključuju u svoj sustav. U 21. stoljeću procesi razvoja softvera usmjereni na ponovnu upotrebu postojećeg softvera postali su široko korišteni jer je broj gotovih rješenja sve veći, a korisnici imaju sve manje vremena za čekanje rješenja.

Početna faza utvrđivanja zahtjeva i faza validacije su usporedive s drugim softverskim procesima, međufaze u procesu ponovne upotrebe su različite.



Slika 7. Model usmjeren na ponovnu upotrebu. Izvor: (Manger, 2016.)

Prednost modela usmjerenog na ponovnu upotrebu je da smanjuje količinu softvera koju treba razviti, a samim time se smanjuje vrijeme, trošak i rizik te se oslonac stavlja na provjerene i dobro testirane dijelove softvera koji dovode do brže isporuke softvera. Međutim, zbog kompromisa može se dogoditi da sustav neće udovoljiti stvarnim potrebama korisnika.

(Sommerville, 2016.), (Manger, 2016.)

8. OPIS PROGRAMA

Aplikacija „Učimo informatiku - SCRATCH“ edukativni je program namijenjen učenicima drugog razreda osnovne škole koji će ih uvesti u osnove programiranja. Uz pomoć programa moći će samostalno usvojiti novi nastavni sadržaj, ponoviti sadržaj bitan za razumijevanje novog nastavnog sadržaja te će sve što su naučili moći utvrditi kroz kratki kviz. Program je rađen u alatu Microsoft Visual Studio 2019, u programskom jeziku Visual BASIC, vodopadnim modelom procesa programiranja. Sadržaj prati nastavne jedinice po udžbeniku Informatike za drugi razred osnovne škole - uvod u programiranje. Nastavna jedinica je osmišljena prateći ishode iz domene Kurikuluma „Računalno razmišljanje i programiranje“ za drugi razred osnovne škole.

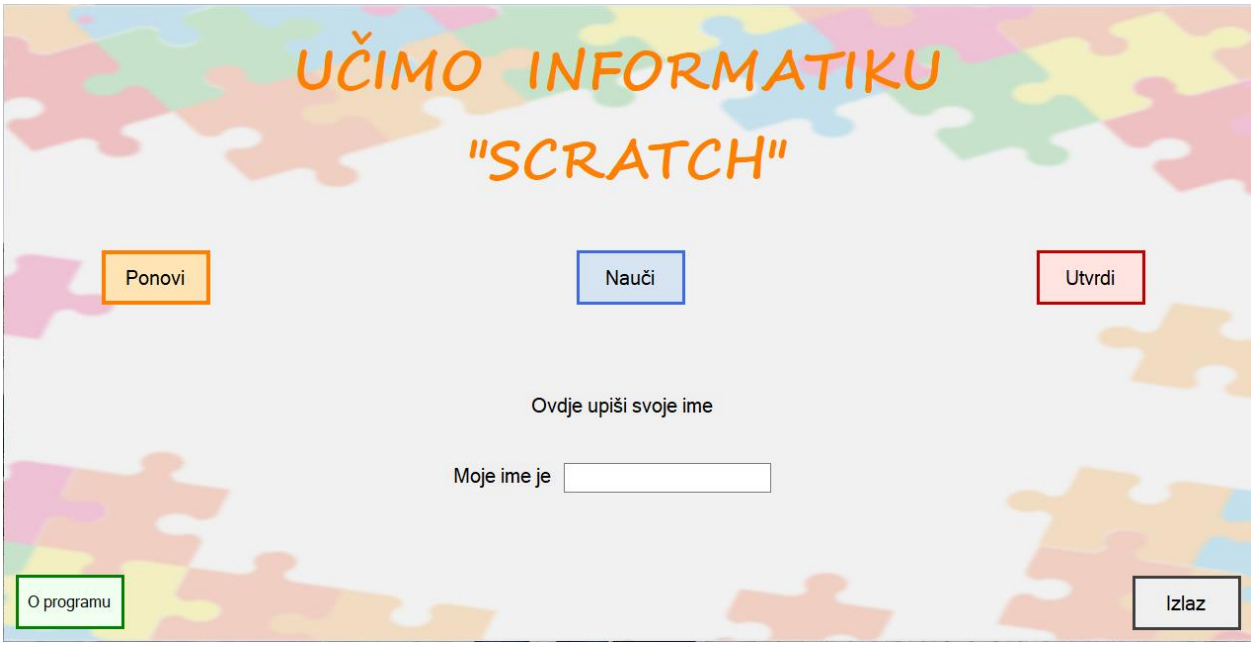
8.1. Utvrđivanje zahtjeva i definicija

Nakon što smo svi bili prisiljeni iskusiti online način rada u školama u zadnjih godinu dana odlučila sam osmisliti i realizirati aplikaciju/program koji će učenicima pomoći da samostalno usvoje novi nastavni sadržaj u školi ili od kuće. Program im omogućuje da kroz povezivanje pojmova ponove ono što im je potrebno iz prethodne nastavne jedinice kako bi usvojili novi nastavni sadržaj. U ovom dijelu se nalazi fotografija s označenim dijelovima koje kroz padajući izbornik učenici moraju odrediti što oni predstavljaju. U dijelu gdje će spoznavati nove činjenice, pojmove i definicije, mogu individualno odabrati tempo rada i zadržati se na pojedinom dijelu onoliko vremena koliko im je potrebno. Na prvom obrascu bit će prikazane boje i naziv grupa odnosno kategorija koje se nalaze u Scratch-u te ono što svaka od njih sadržava. Drugi obrazac će opisivati i prikazivati izgled blokova te način njihovog međusobnog spajanja. Kada su usvojili sve što se nalazi na obrascu (Form) mogu krenuti na sljedeći. Na kraju sve što su naučili provjerit će kratkim kvizom koji sadrži različite vrste zadataka. Na početnoj stranici se nalazi polje u koje mogu upisati svoje ime te izbornik pomoću kojeg mogu odabrati koji dio žele proći ili vidjeti detalje programa. U bilo kojem dijelu ili fazi učenja mogu se vratiti na početnu stranicu ili izaći iz programa.

8.2. Oblikovanje sustava i softvera (dizajn)

Nakon otvaranja nove forme/obrasca (Windows Forms App (.NET)) prema ideji sam preko alatne trake (Toolbox) dodala pet gumbi (Button), tri oznake (Label) i jedan tekstualni okvir

(Textbox). Svaki gumb sam imenovala s obzirom na ono što će se pomoću njega otvarati (O programu, Ponovi, Nauči, Utvrdi, Izlaz). Label sam koristila za pisanje teksta, naslova, tj. imena programa, a tekstualni okvir za mogućnost upisivanja korisničkog imena koje će se ponekad pojavljivati u programu. Svaki obrazac sadrži Label na vrhu prozora s naslovom koji predstavlja gdje se korisnik nalazi.



Slika 8. Početni obrazac

Gumb *„O programu“* otvara obrazac u kojem se nalaze Label koji sadrže tekst te dva gumba, od kojih jedan otvara početni obrazac, a drugi izlazi iz programa.

Slika 9. Obrazac "O programu"

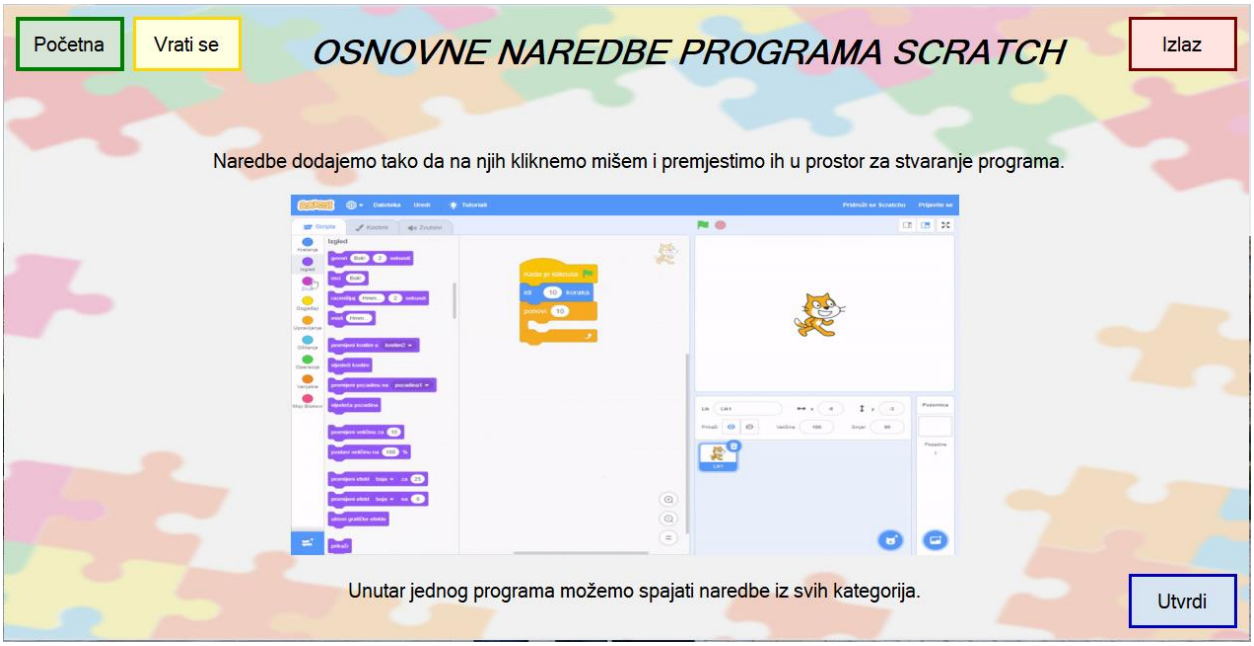
Pritiskom na prvi gumb *„Ponovi“* na početnom obrascu otvara se novi obrazac koji sadrži Label s tekstom, gumbe za početni obrazac, gumb za izlazak iz programa, gumb za provjeru točnosti riješenog zadatka, PictureBox koji sadrži sliku te padajući izbornik (ComboBox) s ponuđenim slovima koje treba označiti. Kada korisnik označi sve odgovore otvara se MessageBox koji provjerava točnost odgovora i tek kada je zadatak točno riješen otvara se novi MessageBox koji pita korisnika želi li naučiti nešto novo ili ga vraća na početni obrazac.

Slika 10. Obrazac "Ponovi"

Gumb *„Nauči“* s početnog obrasca otvara novu formu koja sadrži gumb za povratak na početni obrazac, gumb za izlaz iz programa te gumb za otvaranje novog obrasca koji se sadržajem nastavlja na ovaj. Osim tri gumba, ovaj obrazac još sadrži nekoliko Label-a i nekoliko PictureBox-a sa slikama onoga što tekst definira. Otvaranjem sljedećeg obrasca prikazuju se identične kontrole, različitih svojstva i sadržaja, ali i dodatni (četvrti) gumb kojim se može vratiti na prethodni obrazac. Pritiskom na gumb *„Dalje“* na ovom obrascu otvara se posljednji obrazac u kojem se usvaja novi nastavni sadržaj. Ovaj obrazac sadrži gumbe kod kojih je jedina promjena što gumb za dalje predstavlja odlazak na obrazac u kojem će se sve naučeno utvrditi. Uz gumbe, sadrži i Label-e te jedan PictureBox koji sadrži sliku u GIF formatu pa se prilikom otvaranja tog obrasca ta slika ponaša kao video u petlji.

Slika 11. Obrazac "Nauči"

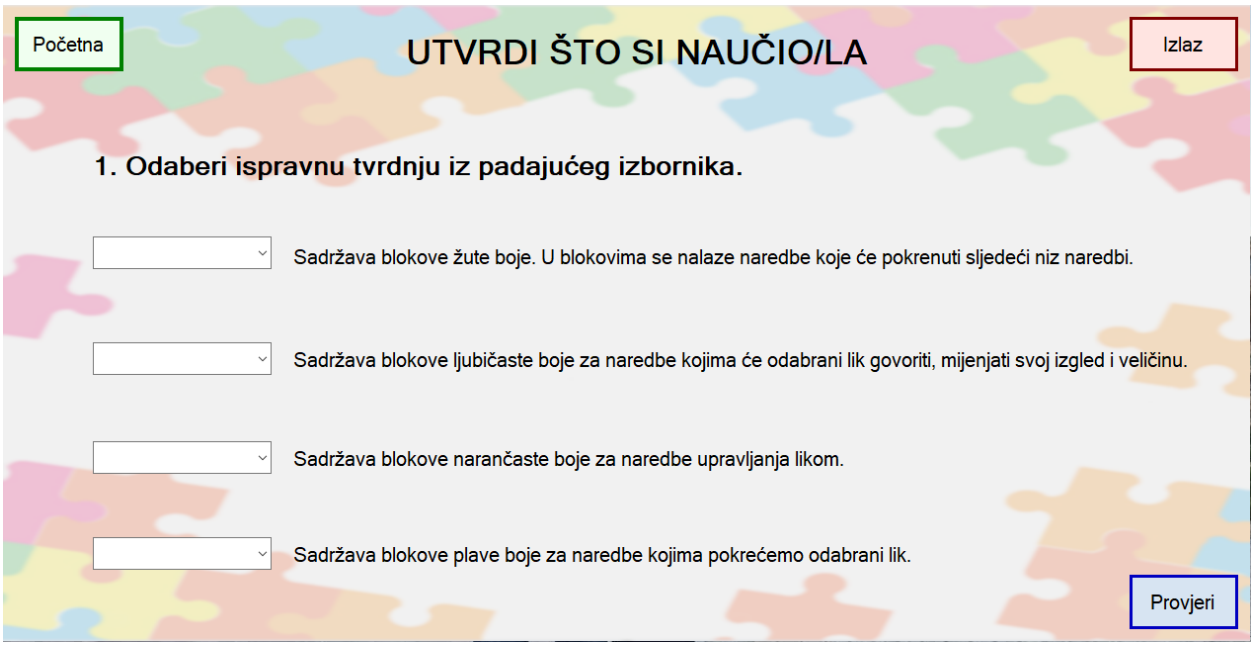
Slika 12. Obrazac "Nauči2"



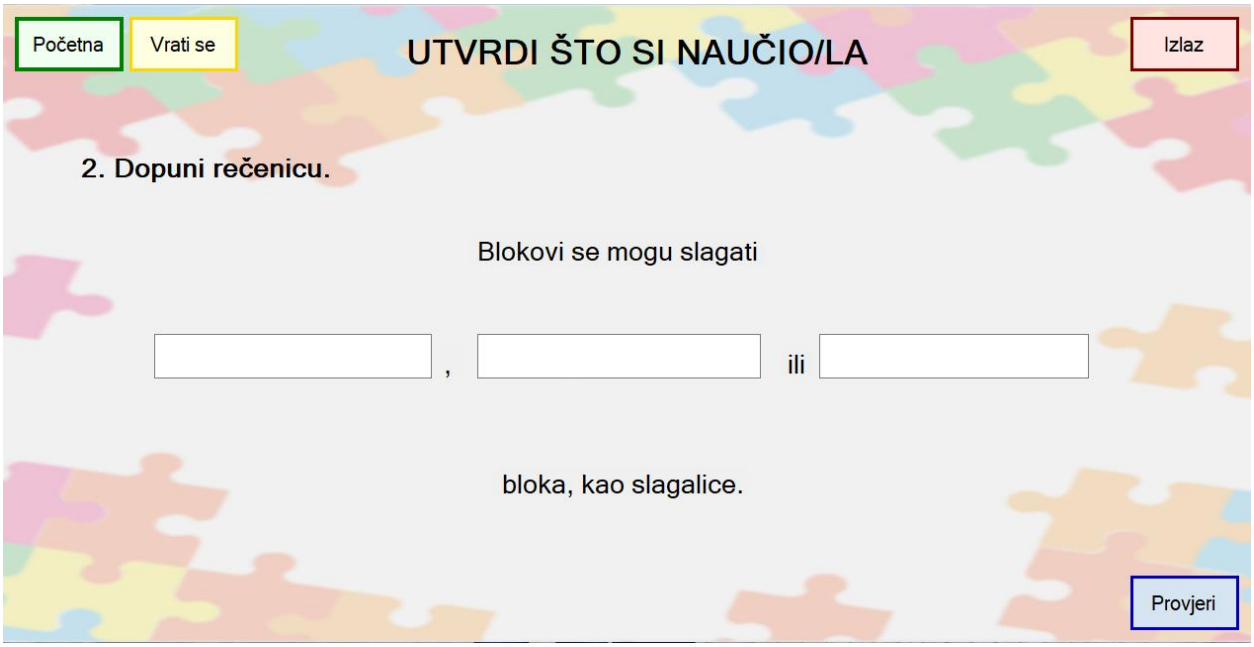
Slika 13. Obrazac "Nauči3"

Posljednji gumb na početnom obrascu *„Utvrdi“* otvara novi obrazac za ponavljanje naučenog sadržaja koji se također sastoji od više obrazaca. Prvi obrazac u ovom dijelu sadrži pitanje i ponuđene odgovore koji su napisani u Label-u (kontroli) te četiri ComboBox-a (padajući izbornik) s ponuđenim odgovorima koje treba točno označiti. Nakon označenih odgovora otvara se MessageBox koji pogrešnim odgovorom vraća na pokušaj, a točnim odgovorom otvara novi obrazac s novim pitanjem. Kao i prethodni obrasci i ovaj ima gumbe pomoću kojih se mogu vratiti na početni obrazac, izaći iz programa te provjeriti točnost riješenog zadatka. Drugi (Utvrdi) obrazac osim povratka na početni obrazac i izlaska iz programa nudi i gumb za povratak na prethodno pitanje. Sadrži uputu za rješavanje ovog pitanja i dio teksta u Label-u, a ostatak teksta od tri riječi mora upisati korisnik u svaki pojedini TextBox. Nakon pravilno upisanih riječi (neovisno o redoslijedu) i pritiskom na gumb *„Provjeri“* otvara se MessageBox koji prikazuje rezultat i pomoću kojeg se otvara novi obrazac koji također sadrži Label s tekstom pitanja. U ovom pitanju korisnik mora točno označiti jednu od četiri napisane tvrdnje u CheckBox-u koji sadrže prazno polje te napisane tvrdnje. Kada je točna tvrdnja označena otvara se MessageBox s rezultatom i novi obrazac s posljednjim pitanjem. Posljednji obrazac više nema gumb za povratak na prethodno pitanje, ali ima gumb kojim korisnik može poništiti pokušaj rješavanja ovog pitanja. Tekstom u Label-u je zadana rečenica po kojoj treba pravilnim redoslijedom posložiti slike koje

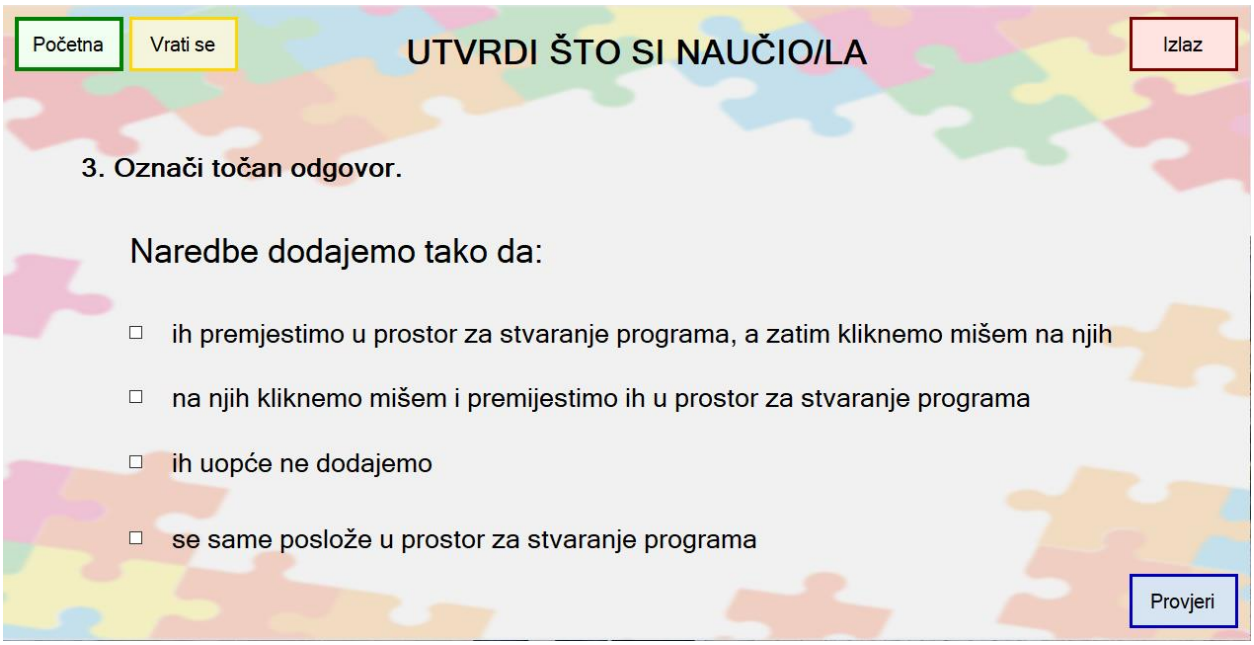
se nalaze u PictureBox-u. Jednom kada se dovuku na mjesto označeno PictureBox-om ne mogu se više pomicati, zato se ovdje nalazi gumb *„Poništi“* u slučaju pogrešnog rasporeda slika.



Slika 14. Obrazac "Utvrdi"



Slika 15. Obrazac "Utvrdi2"



Slika 16. Obrazac "Utvrdi3"

Slika 17. Obrazac "Utvrdi4"

8.3. Implementacija i testiranje modula (programiranje)

Nakon što su postavljene sve kontrole koje su potrebne, za svaku od njih je potrebno napisati kod s obzirom na ono što će svaka od njih raditi.

```
0 references
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Provjeribtn.Click
    If ComboBox1.Text = "D" And ComboBox2.Text = "E" And ComboBox3.Text = "C" And ComboBox4.Text = "F" And ComboBox5.Text = "B" And ComboBox6.Text = "A" Then
        MessageBox.Show("Bravo " & Početna.TextBox1.Text & "! Sve je točno!", "Rezultat")
        Dim rezultat As DialogResult = MessageBox.Show("Želiš li naučiti nešto novo?", "Idemo dalje?", MessageBoxButtons.YesNo)
            If rezultat = DialogResult.Yes Then
                Me.Hide()
                Nauči.Show()
            Else
                Me.Hide()
                Početna.Show()
            End If

        Else
            MessageBox.Show(Početna.TextBox1.Text & " nije sve točno, moraš još malo ponoviti...", "Rezultat")
    End If
End Sub
```

Slika 18. Obrazac "Ponovi"- kod za gumb "Provjeri"

Na obrascu *„Ponovi“* nalazi se gumb *„Provjeri“* koji provjerava je li korisnik točno riješio zadatak. U slučaju da je i samo jedan, ako ne i više, od ponuđenih odgovora iz padajućeg izbornika pogrešno označen pojavljuje se poruka (MessageBox) koja ispisuje ime korisnika koje je ranije upisao na početnom obrascu te tekstualna poruka da nije sve točno označeno. Trenutno odabrani odgovori se ne brišu pa korisnik ne mora rješavati sve ispočetka, nego može promijeniti samo određeni odgovor. Ako je korisnik pravilno označio sva polja iz padajućih izbornika pojavljuje se poruka (MessageBox) s pohvalom i imenom korisnika. Ova poruka ima ponuđen samo gumb *„OK“* koji prilikom pritiska otvara novu poruku s pitanjem, želi li korisnik sada naučiti nešto novo ili ne, pa tako ova poruka nudi dva gumba za odabir, *„DA“* ili *„NE“*. Pritiskom na gumb *„NE“* trenutni obrazac *„Ponovi“* se zatvara i program se vraća na početni obrazac, na kojem se ponovno može odabrati koji dio se želi proći. Pritiskom na gumb *„DA“* otvara se obrazac *„Nauči“*, a trenutni obrazac se zatvara.

```
0 references
Private Sub Provjeribtn_Click(sender As Object, e As EventArgs) Handles Provjeribtn.Click
    If ComboBox1.Text = "DOGAĐAJI" And ComboBox2.Text = "IZGLED" And ComboBox3.Text = "UPRAVLJANJE" And ComboBox4.Text = "KRETANJE" Then
        Dim rezultat As DialogResult = MessageBox.Show("Bravo " & Početna.TextBox1.Text & "! Idemo na drugo pitanje. :)", "Rezultat")
        If rezultat = DialogResult.OK Then
            Dim obrisi = New Utvrdi2
            obrisi.Show()
            Me.Close()
        End If
    Else
        MessageBox.Show(Početna.TextBox1.Text & " nije sve točno, pokušaj još jednom...", "Rezultat")
        Dim obrisi = New Utvrdi
        obrisi.Show()
        Me.Close()
    End If
End Sub
```

Slika 19. Obrazac "Utvrdi"- kod za gumb "Provjeri"

Gumb *„Utvrdi“* otvara četiri pitanja, tj. obrasca, jedno po jedno, nakon točno odgovorenog pitanja. Na prvom obrascu *„Utvrdi“* nalazi se vrsta pitanja kao i u obrascu *„Ponovi“* - padajući izbornik. Razlika kod ovog padajućeg izbornika je u sadržaju ponuđenih opcija. Ovdje su ponuđeni pojmovi koji moraju biti označeni s obzirom na definiciju koja je uz svaki od njih napisana. Kao i

u prošlom obrascu i ovdje se nalazi gumb *„Provjeri“* koji prilikom pritiska provjerava točnost označenih pojmova. Ukoliko korisnik nije točno označio pojmove, pritiskom na gumb *„Provjeri“* otvorit će se poruka koja sadrži ime korisnika i uputu kako zadatak nije točno riješen te savjet neka pokuša ponovo. U ovom prikazu poruke, pritiskom na gumb *„OK“* označeni pojmovi ne ostaju zabilježeni već se brišu i ponavlja se zadatak s praznim poljima padajućeg izbornika. Kada su svi pojmovi određenog padajućeg izbornika točno odabrani, pritiskom na gumb *„Provjeri“* pojavljuje se poruka s čestitkom korisniku i prelazak na drugo pitanje. Iako su svi odgovori točno označeni, kada se pritisne gumb *„OK“* brišu se svi odgovori, ostaju prazna polja i otvara se novi obrazac *„Utvrdi“* s drugim pitanjem.

```
0 references
Private Sub Provjeribtn_Click(sender As Object, e As EventArgs) Handles Provjeribtn.Click
    If (TextBox1.Text.ToLower = "ispod" Or TextBox1.Text.ToLower = "iznad" Or TextBox1.Text.ToLower = "unutar") And
    (TextBox2.Text.ToLower = "ispod" Or TextBox2.Text.ToLower = "iznad" Or TextBox2.Text.ToLower = "unutar") And
    (TextBox3.Text.ToLower = "ispod" Or TextBox3.Text.ToLower = "iznad" Or TextBox3.Text.ToLower = "unutar") Then
        Dim rezultat As DialogResult = MessageBox.Show("Bravo " & Početna.TextBox1.Text & "! Idemo na treće pitanje. :)", "Rezultat")
        If rezultat = DialogResult.OK Then
            Dim obrisi = New Utvrdi3
            obrisi.Show()
            Me.Close()
        End If
    Else
        MessageBox.Show(Početna.TextBox1.Text & " nije točno, pokušaj još jednom...", "Rezultat")
        Dim obrisi = New Utvrdi2
        obrisi.Show()
        Me.Close()
    End If
End Sub
```

Slika 20. Obrazac "Utvrdi2"- kod za gumb "Provjeri"

Drugo pitanje nalazi se na obrascu *„Utvrdi2“*, pitanje je za nadopunjavanje rečenice pojmovima. Pritiskom na gumb *„Provjeri“* program provjerava je li korisnik točnim riječima nadopunio rečenicu. Uvjet koji je postavljen je da moraju biti upisane točne riječi, ali nije bitan redoslijed upisivanja. Ako neka od riječi nije točna ili dobro upisana program će prikazati poruku da zadatak nije točno riješen, poništit će, tj. obrisati trenutno upisane odgovore, ponovo otvoriti obrazac *„Utvrdi2“* te na taj način omogućiti korisniku ponovni pokušaj rješavanja. Tek kada su sve riječi točno upisane, otvara se poruka koja čestita korisniku i najavljuje novo pitanje. Prilikom pritiska na gumb *„OK“* u prikazanoj poruci, otvara se novi obrazac s pitanjem.

```
0 references
Private Sub Provjeribtn_Click(sender As Object, e As EventArgs) Handles Provjeribtn.Click
    If CheckBox2.Checked And CheckBox1.Checked = False And CheckBox3.Checked = False And CheckBox4.Checked = False Then
        Dim rezultat As DialogResult = MessageBox.Show("Bravo " & Početna.TextBox1.Text & "! Idemo na zadnje pitanje. :)", "Rezultat")
        If rezultat = DialogResult.OK Then
            Dim obrisi = New Utvrdi4
            obrisi.Show()
            Me.Close()
        End If
    Else
        MessageBox.Show(Početna.TextBox1.Text & " nije točno označeno, pokušaj još jednom...", "Rezultat")
        Dim obrisi = New Utvrdi3
        obrisi.Show()
        Me.Close()
    End If

End Sub
```

Slika 21. Obrazac "Utvrdi3"- kod za gumb "Provjeri"

Obrazac *„Utvrdi3“* otvara treće pitanje u kojemu korisnik mora označiti prazno polje ispred tvrdnje koju smatra točnom. Ispred svake tvrdnje se nalazi prazno polje/kvadratić koje se prilikom pritiska mišem na tvrdnju ili polje, ispuni/označi kvačicom (CheckBox). Ako je označno više polja ili pogrešna tvrdnja prilikom pritiska na gumb *„Provjeri“* program šalje poruku korisniku kako nije točno označio i upućuje ga na ponovni pokušaj. Kao i u prethodnom obrascu, trenutno označno polje ili polja se brišu i korisnik može početi s ponovnim pokušajem rješavanja. Tek kada je označeno polje ispred točne tvrdnje pojavljuje se poruka s pohvalom i uputom za prelazak na sljedeće, posljednje pitanje. Kada se pritisne gumb *„OK“* na prikazanoj poruci, označeno polje se briše i otvara se novi obrazac s pitanjem.

```
3     Dim m_MouseIsDown1 As Boolean
4     Dim m_MouseIsDown2 As Boolean
5     Dim m_MouseIsDown3 As Boolean
6     Dim blok1 As String
7     Dim blok2 As String
8     Dim blok3 As String
9     Dim polazni_blok As String
10
```

Slika 22. Obrazac "Utvrdi4"- kontrolne varijable

Za posljednje pitanje koje se nalazi na obrascu *„Utvrdi4“*, gdje korisnik mora slike posložiti u za to predviđene okvire pravilnim redoslijedom. Na početku koda obrasca *„Utvrdi4“* su definirane kontrolne varijable (vrijednosni tip Boolean) koje služe za prepoznavanje je li koji od blokova slika (PictureBox) s desne strane označen. Blokovima s lijeve strane, varijable (blok1, blok2 i blok3) predstavljaju dodjelu naziva slika s desne strane koju je korisnik odabrao.

```
0 references
Private Sub idi10koraka_MouseDown(sender As Object, e As MouseEventArgs) Handles idi10koraka.MouseDown
    polazni_blok = "idi10koraka"
    If Not idi10koraka.Image Is Nothing Then
        m_MouseIsDown1 = True
    End If
End Sub

0 references
Private Sub idi10koraka_MouseMove(sender As Object, e As MouseEventArgs) Handles idi10koraka.MouseMove
    If m_MouseIsDown1 Then
        idi10koraka.DoDragDrop(idi10koraka.Image, DragDropEffects.Move)
    End If
    m_MouseIsDown1 = False
End Sub

0 references
Private Sub prvopolje_DragDrop(sender As Object, e As DragEventArgs) Handles prvopolje.DragDrop
    blok1 = polazni_blok
    prvopolje.Image = e.Data.GetData(DataFormats.Bitmap)
    If m_MouseIsDown1 Then
        idi10koraka.Image = Nothing
    ElseIf m_MouseIsDown2 Then
        cekaj5s.Image = Nothing
    Else
        recibok.Image = Nothing
    End If
End Sub

0 references
Private Sub prvopolje_DragEnter(sender As Object, e As DragEventArgs) Handles prvopolje.DragEnter
    If e.Data.GetDataPresent(DataFormats.Bitmap) Then
        e.Effect = DragDropEffects.Move
    Else
        e.Effect = DragDropEffects.None
    End If
End Sub
```

Slika 23. Obrazac "Utvrdi4"- kod za raspoređivanje slika 1

Ovisno o tome koju sliku s desne strane, tj. polje sa slikom korisnik odabere i pritisne, varijabla *„polazni_blok“* poprima njegovu vrijednost - naziv. Tek kada korisnik odabere polje sa slikom i drži pritisnutu lijevu tipku miša (funkcija MouseDown) na toj slici, program dopušta ispuštanje, tj. premještanje odabrane slike (funkcija *DoDragDrop*). Bilo koja od tri slike s desne strane se može premjestiti u bilo koje predviđeno polje (blok) s lijeve strane. Kada se slika ispusti, ostaje s lijeve i više nije vidljiva s desne strane. Jednom premještena slika, više se ne može premještati, osim ako se ne pritisne gumb *„Poništi“*. Ovaj dio koda isti je za svaku od tri ponuđene slike za premještanje i za sva tri polja gdje se te slike mogu ispustiti.

```
0 references
Private Sub cekaj5s_MouseDown(sender As Object, e As MouseEventArgs) Handles cekaj5s.MouseDown
    polazni_blok = "cekaj5s"
    If Not cekaj5s.Image Is Nothing Then
        m_MouseIsDown2 = True
    End If
End Sub

0 references
Private Sub cekaj5s_MouseMove(sender As Object, e As MouseEventArgs) Handles cekaj5s.MouseMove
    If m_MouseIsDown2 Then
        cekaj5s.DoDragDrop(cekaj5s.Image, DragDropEffects.Move)
    End If
    m_MouseIsDown2 = False
End Sub

0 references
Private Sub drugopolje_DragDrop(sender As Object, e As DragEventArgs) Handles drugopolje.DragDrop
    blok2 = polazni_blok
    drugopolje.Image = e.Data.GetData(DataFormats.Bitmap)
    If m_MouseIsDown1 Then
        idi10koraka.Image = Nothing
    ElseIf m_MouseIsDown2 Then
        cekaj5s.Image = Nothing
    Else
        recibok.Image = Nothing
    End If
End Sub

0 references
Private Sub drugopolje_DragEnter(sender As Object, e As DragEventArgs) Handles drugopolje.DragEnter
    If e.Data.GetDataPresent(DataFormats.Bitmap) Then
        e.Effect = DragDropEffects.Move
    Else
        e.Effect = DragDropEffects.None
    End If
End Sub
```

Slika 24. Obrazac "Utvrdi4"- kod za raspoređivanje slika 2

```
0 references
Private Sub recibok_MouseDown(sender As Object, e As MouseEventArgs) Handles recibok.MouseDown
    polazni_blok = "recibok"
    If Not recibok.Image Is Nothing Then
        m_MouseIsDown3 = True
    End If
End Sub

0 references
Private Sub recibok_MouseMove(sender As Object, e As MouseEventArgs) Handles recibok.MouseMove
    If m_MouseIsDown3 Then
        recibok.DoDragDrop(recibok.Image, DragDropEffects.Move)
    End If
    m_MouseIsDown3 = False
End Sub

0 references
Private Sub trecepolje_DragDrop(sender As Object, e As DragEventArgs) Handles trecepolje.DragDrop
    blok3 = polazni_blok
    trecepolje.Image = e.Data.GetData(DataFormats.Bitmap)
    If m_MouseIsDown1 Then
        idi10koraka.Image = Nothing
    ElseIf m_MouseIsDown2 Then
        cekaj5s.Image = Nothing
    Else
        recibok.Image = Nothing
    End If
End Sub

0 references
Private Sub trecepolje_DragEnter(sender As Object, e As DragEventArgs) Handles trecepolje.DragEnter
    If e.Data.GetDataPresent(DataFormats.Bitmap) Then
        e.Effect = DragDropEffects.Move
    Else
        e.Effect = DragDropEffects.None
    End If
End Sub
```

Slika 25. Obrazac "Utvrdi4"- kod za raspoređivanje slika 3

```
0 references
Private Sub Provjeribtn_Click(sender As Object, e As EventArgs) Handles Provjeribtn.Click
    If blok1 = "idi10koraka" And blok2 = "cekaj5s" And blok3 = "recibok" Then
        MessageBox.Show("Bravo " & Početna.TextBox1.Text & "!", "Rezultat")
        Dim rezultat As DialogResult = MessageBox.Show("SVE SMO PROŠLI! SADA MOŽEŠ IĆI PROGRAMIRATI U SCRATCH-u ILI SE MOŽEŠ IĆI IGRATI :)", "KRAJ!")
        Me.Hide()
        Početna.Show()
    Else
        MessageBox.Show("Krivo! Pokušaj ponovno!")
    End If
End Sub

0 references
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim ponisti = New Utvrdi4
    ponisti.Show()
    Me.Close()
End Sub
```

Slika 26. Obrazac "Utvrdi4"- kod za gumb "Provjeri" i „Poništi“

Kada se sve slike poslože u predviđena polja, korisnik svoje rješenje provjerava pritiskom na gumb *„Provjeri“* koji otvara tekst poruke ovisno o točnosti riješenog zadatka. Ako slike nisu poredane pravilnim redoslijedom prikazuje se poruka koja upućuje korisnika da je pogriješio i

neka pokuša ponovno. Ponovni pokušaj može ostvariti pritiskom na gumb *„Poništi“* koji poništava trenutno posložene slike, tj. ponovno otvara trenutni obrazac s četvrtim pitanjem *„Utvrdi4“* i vraća slike na početnu poziciju za ponovno preslagivanje. Tek kada su sve slike posložene pravilnim redoslijedom prikazuje se poruka *„Bravo“* i korisničko ime s jednim gumbom *„OK“* koji prilikom pritiska otvara sljedeću poruku s pozdravnom porukom za kraj kviza i programa. Ova posljednja poruka također sadrži samo gumb *„OK“* koji kada se pritisne otvara, tj. vraća korisnika na početni obrazac gdje opet može odabrati koji dio želi proći, ponoviti ili izaći iz programa.

8.4. Testiranje sustava

Nakon izrade programa, cijeli kod prolazi kroz testiranje i provjerava pune li se vrijednosti varijabli pravilno prilikom pritiska na određene objekte (otvaranje forme, upisivanje teksta, odabir iz padajućeg izbornika i dr.). Aplikacija je testirana od strane autorice nakon izrade svakog pojedinog obrasca te ista još nije puštena u rad krajnjem korisniku kome je namijenjena. Ukoliko se prikazuje greška (Bug) program upućuje na dio koda gdje se ona nalazi i gdje program ne radi kako treba. Neke od pogrešaka su bile logičke, npr. prilikom pritiska na određeni gumb otvorio se „pogrešan“ obrazac ili se uopće ništa nije otvorilo. Još jedna od logičkih pogrešaka bila je da pritisak na gumb *„Provjeri“* nije pravilno provjerio točnost zadatka, pustio je korisnika na sljedeće pitanje, a zadatak nije bio točno riješen. Ove greške riješene su na način da se pronađe i promijeni dio koda koji treba otvarati određeni obrazac, a za gumb *„Provjeri“* je bilo potrebno promijeniti uvjet (*If, Else*) koji korisnik treba zadovoljiti kako bi mogao ići na sljedeće pitanje. Osim logičkih, pojavljivale su se i neke sintaksne pogreške prilikom samog pisanja programa. Alat Visual Studio automatski podcrtava sintaksne pogreške koje je potrebno zamijeniti kako bi se program pokrenuo, pa je samim time olakšan njihov pronalazak. Neke od tih pogrešaka su bile, izostavljanje zareza, navodnih znakova, upisivanje varijable koja nije prethodno definirana te pogrešno napisane riječi.

8.5. Puštanje u rad i održavanje

Aplikaciju je jednostavno prilagoditi novim zahtjevima korisnika. Ako dođe do potrebe, mogu se izmijeniti načini pitanja-odgovora. Na početnom obrascu se može napraviti veći i sadržajniji izbornik s više nastavnih jedinica koje su međusobno povezane. Nastavne jedinice mogu biti povezane u nastavne cjeline, u ovom slučaju za učenje informatike u drugom razredu osnovne škole. Svaka od nastavnih jedinica tražit će znanje prethodne nastavne jedinice, a na kraju

svake može biti kratki kviz za ponavljanje naučenog. U sklopu svake pojedine nastavne cjeline na kraju se može nalaziti kviz koji obuhvaća sve jedinice pojedine cjeline. U budućnosti se može dodati uspješnost riješenosti kviza i prikaz ukupnog rezultata što nastavnicima može poslužiti kao provjera znanja i povratna informacija o usvojenosti nastavnog sadržaja. Tada bi ostavila mogućnost povratka na prethodno pitanje, ali bi ostao zabilježen odgovor, ne bi se brisao prilikom prelaska na sljedeće pitanje, kako bi se na kraju mogao vidjeti ukupan rezultat. Osim rezultata i novog nastavnog sadržaja, mogu se dodati i nove opcije, poput kalendara, brojača vremena za rješavanje zadataka i dr. Program se također može prilagoditi za bilo koji drugi predmet i razred u razrednoj nastavi.

9. ZAKLJUČAK

Uz besplatni alat Microsoft Visual Studio 2019 i programski jezik Visual BASIC autorica ovog programa uspješno je napravila edukativnu aplikaciju „Učimo informatiku - SCRATCH“ za učenike drugog razreda osnovne škole. Prvi susret s alatom Microsoft Visual Studio 2019 i programskim jezikom Visual BASIC zahtjeva dodatno proučavanje literature. Uz Microsoft dokumentaciju i podršku, video materijale i ostale internet stranice puno je lakše i jednostavnije razumjeti način rada u Visual Studio alatu. Visual BASIC, također, ima velik broj primjera i stručne literature pomoću koje je upoznavanje i učenje ovog programskog jezika uvelike olakšano. Zbog toga je idealno za početnike i one koji žele napraviti svoju prvu aplikaciju. Zbog unaprijed osmišljene ideje, dizajna, načina rada, mogućnosti aplikacije i osjećaja potrebe za izradom takve aplikacije, „Učimo informatiku - SCRATCH“ rađena je prema vodopadnom modelu procesa razvoja softvera. Upravo takav model idealan je za detaljno planiranje izrade programa te njenu realizaciju i puštanje u rad. Ova aplikacija će im omogućiti da u bilo kojem trenutku samostalno usvoje novi nastavni sadržaj. Uvest će ih u početke programiranja i pripremiti za daljnje učenje informatike. Tijekom izrade ove aplikacije autorica je stekla mnoga nova znanja, iskusila programiranje i izradu vlastite aplikacije. Naišla je na probleme i komplikacije, ali uz pomoć literature i mentora kojemu je neizmjerno zahvalna na pomoći, sve je uspješno napravila. Nakon završetka izrade programa osjećala se ponosno i sretno, znajući da je izradila aplikaciju koja uvijek može biti korisna učenicima, ali i koju ću moći primijeniti u svom budućem zanimanju, bilo kao učiteljica razredne nastave ili učiteljica informatike u osnovnoj školi. Izrada aplikacije s ovom nastavnom jedinicom motivirala ju je za izradu i nadogradnju dodatnih nastavnih jedinica i kviza ili ispita znanja kojim će se obrađeni nastavni sadržaj moći provjeriti i dobiti povratna informacija o usvojenosti znanja učenika.

10. LITERATURA

Blagus, J., Ljubić Klemše, N., Flisar Odorčić, A., Mihočka, N. i Ružić, I. (2020). *e-SVIJET 2, radni udžbenik informatike u drugom razredu osnovne škole.* Školska knjiga, d.d.

Budojević, A. i Kanić, A. (2019). *Istražite SCRATCH 3.0 Radni priručnik za rano učenje programiranja*. Školska knjiga, d.d.

Damjanović, S., Katanić, P. (2014). *Programski jezik Visual Basic zbirka zadataka*. Fakultet poslovne ekonomije, Bijeljina

Del Sole, A. (2019). *Visual Studio 2019 Succinctly.* Syncfusion.

Đurđević, I. (2013). *Procjene studenata učiteljskog studija o tri računalna programa namijenjena malim početnicima u programiranju.* [Stručni rad]. Zavod za znanstveni i umjetnički rad u Požegi. Preuzeto: 10.04.2021.

Halvorson, M. (2013). *Microsoft Visual Basic 2103 Step by Step*. Microsoft Press

Halvorson, M. (2010). *Visual Basic 2010*. Microsoft Press

Latinović, T. (2007). *Osnove Programiranja (Visual Basic)*. Besjeda

Leach, R. J. (2016). *Introduction to Software Engineering Second Edition*. CRC Press

Manger, R. (2016). *Softversko inženjerstvo.* Element.

Marji, M. (2014). *Learn to program with Scratch*. No Starch Press

Microsoft Press, (1999). *Microsoft Visual Basic 6.0, Vodič za programere.* Znak 2000.

Ministarstvo znanosti i obrazovanja [MZO]. (2018). *Kurikulum nastavnog predmeta Informatika za osnovne škole i gimnazije.* https://skolazazivot.hr/wp-content/uploads/2020/06/INF_kurikulum.pdf

Oreški, P. *Programiranje* (predavanja)

SCRATCH. *About Scratch.* https://scratch.mit.edu/about

Selmanović, N., Delalić, S. (2019). *Uvod u programiranje - Elementi teorije sa zbirkom riješenih zadataka*. Univerzitet u Sarajevu - Prirodno-matematički fakultet

Sheldon, B., Hollis, B., Windsor, R., McCarter, D., Hillar, G.C., Herman, T. (2012). *Visual Basic 2012 and .NET 4.5 Programming*. John Wiley & Sons, Inc.

Sommerville, I. (2016). *Software Engineering (10th ed.).* Pearson.

Tutorialspoint, Simple easy learning. (2014). *Software Engineering*.

Voon Kiong, L. (2017). *Visual Basic 2017 Made Easy*.

Welcome to the Visual Studio IDE. Visual Studio Docs. Microsoft. https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019

Wilson, A. i Moffat, D. C. (2010). *Evaluating Scratch to introduce younger schoolchildren to programming.*
https://scratched.gse.harvard.edu/sites/default/files/wilson-moffat-ppig2010-final.pdf

11. PRILOZI I DODATCI

Slika 1. Četiri domene kurikuluma. Izvor: (MZO, 2018.) 3
Slika 2. SCRATCH. Izvor: (https://scratch.mit.edu/projects/editor/?tutorial=getStarted) 8
Slika 3. Visual Studio 2019 9
Slika 4. Vodopadni model. Izvor: (Manger, 2016.) 14
Slika 5. Model evolucijskog razvoja. Izvor: (Manger, 2016.) 15
Slika 6. Model inkrementalnog razvoja. Izvor: (Manger, 2016.) 15
Slika 7. Model usmjeren na ponovnu upotrebu. Izvor: (Manger, 2016.) 16
Slika 8. Početni obrazac 18
Slika 9. Obrazac "O programu" 19
Slika 10. Obrazac "Ponovi" 20
Slika 11. Obrazac "Nauči" 21
Slika 12. Obrazac "Nauči2" 21
Slika 13. Obrazac "Nauči3" 22
Slika 14. Obrazac "Utvrdi" 23
Slika 15. Obrazac "Utvrdi2" 23
Slika 16. Obrazac "Utvrdi3" 24
Slika 17. Obrazac "Utvrdi4" 24
Slika 18. Obrazac "Ponovi"- kod za gumb "Provjeri" 25
Slika 19. Obrazac "Utvrdi"- kod za gumb "Provjeri" 25
Slika 20. Obrazac "Utvrdi2"- kod za gumb "Provjeri" 26
Slika 21. Obrazac "Utvrdi3"- kod za gumb "Provjeri" 27
Slika 22. Obrazac "Utvrdi4"- kontrolne varijable 27
Slika 23. Obrazac "Utvrdi4"- kod za raspoređivanje slika 1 28
Slika 24. Obrazac "Utvrdi4"- kod za raspoređivanje slika 2 29
Slika 25. Obrazac "Utvrdi4"- kod za raspoređivanje slika 3 30
Slika 26. Obrazac "Utvrdi4"- kod za gumb "Provjeri" i „Poništi“ 30

12. IZJAVA O IZVORNOSTI RADA

Izjavljujem da je moj diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristila drugim izvorima osim onih koji su u njemu navedeni.

_____

(vlastoručni potpis studenta)