

UNIVERZA V LJUBLJANI  
FAKULTETA ZA MATEMATIKO IN FIZIKO

Matematika – 2. stopnja

in

INSTITUT “JOŽEF STEFAN”  
LJUBLJANA

Petra Poklukar

**Poročilo o praktičnem usposabljanju**

Mentor: prof. dr. Franc Novak

Ljubljana, 2015

## ZAHVALA

Rada bi se zahvalila svojemu mentorju, prof. dr. Francu Novaku za zanimiv izbor projekta ter prijaznost, potrpežljivost in usmerjanje med samim usposabljanjem. Zahvaljujem se tudi odsečni tajnici Jolandi Jakofčič za urejanje vseh potrebnih uradnih listin in ostalim zaposlenim za dobro vzdušje.

## KAZALO

1. Uvod	4
2. Delo na praktičnem usposabljanju	4
2.1. Naprava za sledenje pogleda	4
2.2. Razvoj programske opreme	5
3. Pridobljena znanja	8
4. Priloge k poročilu	8
Literatura	8

## 1. UVOD

Institut “Jožef Stefan”, s sedežem v Ljubljani, izvaja raziskave in razvoj tehnologij na področju naravoslovnih in tehniških znanosti ter znanosti o življenju. Pod vodstvom direktorja prof. dr. Jadrana Lenarčiča intenzivno sodeluje pri spodbujanju tehnološkega in gospodarskega razvoja v Sloveniji, kar se kaže v obliki raziskav ter šolanja kadrov. V vseh svojih letih delovanja so v financiranje sprejeli že več kot tisoč mladih raziskovalcev, prav tako pa so podelili številne stipendije, večinoma študentom računalništva, elektrotehnike, kemije in kemijske tehnologije, fizike in matematike. Institut sodeluje tako s številnimi slovenskimi tehnološkimi centri, centri odličnosti in fakultetami, kot tudi z raznimi tujimi ustanovami. Ima okoli tisoč zaposlenih, od tega pa je že skoraj polovica doktorjev znanosti. Pohvali se lahko z objavami mnogih znanstvenih člankov, monografij, patentov in tudi opravljenimi doktorskimi ter magistrskimi deli. Organizacijsko je razdeljen na štiri večje raziskovalne skupine: fizika, kemija in biokemija, elektronika in informacijske tehnologije ter jedrska tehnika in energetika, vsaka od teh pa se deli še na manjše raziskovalne odseke, katerih skupno število je 28. Eden izmed odsekov skupine za elektroniko in informacijske tehnologije je tudi odsek za Računalniške sisteme E7, kjer je potekalo moje praktično usposabljanje. Odsečne raziskave so usmerjene v področje razvoja naprednih računalniških struktur in zmogljivih algoritmov za obdelavo večjih količin podatkov ter sistemov za učinkovito interakcijo med človekom in računalnikom. Ukvarjajo se tudi s samopopravljivimi in samoorganizirajočimi sistemi, modeliranjem in optimiziranjem kompleksnih, dinamičnih in nedeterminističnih sistemov. Aplikacije, ki jih razvijajo v okviru svojih raziskav, so uporabljene na področjih bioinformatike, zdravja, medicine, proizvodnje, transporta, energetike in okoljske vzdržnosti. Kolektiv sestavlja 11 zaposlenih in 14 (zunanjih) sodelavcev. Odsek se lahko pohvali z visokim nivojem znanja z omenjenih raziskovalnih področij ter številnimi objavljenimi strokovnimi članki, prispevki v konferenčnih zbornikih in raziskovalnimi projekti.

## 2. DELO NA PRAKTIČNEM USPOSABLJANJU

Kot rečeno, je eno izmed področij odsečnih raziskav tudi interakcija med človekom in računalnikom, s katerim se aktivno ukvarja tudi moj mentor prof. dr. Franc Novak. Po njegovem predlogu sem v okviru praktičnega usposabljanja razvila programsko opremo naprave za sledenje pogleda (ang. *eyetracker*).

**2.1. Naprava za sledenje pogleda.** Sledenje očem nam je omogočila naprava Tobii EyeX Controller, ki obsega komplet projektorjev, treh kamer, algoritmov za procesiranje slike in matematičnih izračunov za določitev položaja oči. To napravi omogoča, da z visoko stopnjo natančnosti lahko determinira, kam na zaslonu uporabnik gleda, položaj uporabnikovih oči, samo prisotnost uporabnika ter velikost njegovih zenic. Uporaba sledenja očem je prisotna na različnih področjih. V psihologiji se uporablja za raziskovanje delovanja vizualnega spomina, branja in razumevanja jezika, v medicini pa za raziskovanje možganske aktivnosti. V računalništvu se uporablja za testiranje učinkovitosti različnih uporabniških vmesnikov in omogoča prostoročno komunikacijo med človekom in računalnikom. Vse bolj se uporablja tudi v marketingu, kjer s sledenjem premikom oči ocenjujejo uspešnost reklamnih akcij, merijo učinkovitost grafičnega oblikovanja oglasov, spremljajo vidljivost reklam na

spletnih straneh itd. Več o napravi si lahko bralec prebere na uradni spletni strani proizvajalca [6].

Motivacija za razvoj programske opreme na praktičnem usposabljanju je bila predvsem analiza spletnih strani.

**2.2. Razvoj programske opreme.** Po dogovoru z mentorjem, sem programsko opremo pisala v objektno usmerjenem programskem jeziku *C#* v programu Visual Studio 2013. V nadaljevanju bodo izrazi povezani z programsko kodo zaradi boljše razumljivosti večinoma ostali neprevedeni v angleškem jeziku.

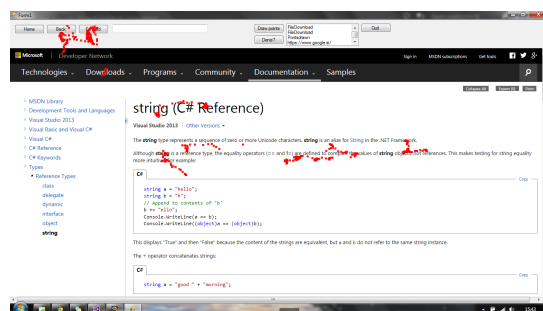
Zaradi pomanjkanja predznanja programskega jezika *C#*, sem najprej opravila spletni tečaj "Programming with *C#*" v okviru 60 ur, ki je bil ponujen na spletni strani [1] (potrdilo o opravljenem tečaju je dodan kot priloga k poročilu). Tečaj je obsegal 12 modulov, kjer je posamezen modul vključeval tekočo snov, primere, samopreverjanja, naloge za vajo ter domačo nalogo, ki jo je bilo potrebno oddati v ocenjevanje. Obseg snovi v posameznih moduli:

- Modul 1:** Obsega uvod v programski jezik *C#* in možna orodja za pisanje programske kode v *C#*. Predstavljeni so podatkovni tipi, izjave, operatorji in možne pretvorbe med podatkovnimi tipi. Navedene so rezervirane ključne besede in pa tako imenovani *identifiers* - elementi programa. V *C#* so to *variables, methods, classes* in *namespaces*.
- Modul 2:** Predstavitev *if* in *switch* stavka ter zank *for*, *for each*, *while* in *do*.
- Modul 3:** Deklaracija in klic metode ter uporaba opcijskih in poimenovanih parametrov. Predstavljene so tudi izjeme in z njimi povezane naslednje tematike: *exception handling, try/catch/finally block* in *exception throwing*.
- Modul 4:** Predstavljene so definicije in uporabe kompleksnejših podatkovnih struktur, kot so *single* in *multidimensional arrays, enum structure* ter *struct* in *property constructs*.
- Modul 5:** Deklaracija ter uporaba identifikatorja *class* in njegovih članov: *methods, fields, properties* in *events*. Predstavljeni so tudi tako imenovani *access modifiers, partial classes, anonymous* in *static classes, static methods* ter *class constructors*.
- Modul 6:** Modul obravnava naslednje teme: *inheritance, abstract* in *sealed classes* ter *interfaces*. Predstavi tudi življenski cikel objekta (*object lifecycle*), tako imenovani *garbage collector* in opiše, kako pravilno implementirati *Dispose pattern*.
- Modul 7:** Predstavitev *C#* zbirk, ki se delijo na dve skupini: *standard C# collections* in *specialited collections*. Pri tem prva skupina vsebuje na primer *ArrayList, Queue, Stack, Hashtable, ...*, druga pa na primer *ListDictionary, StringCollection, BitVector32, ...*. Modul predstavi tudi uporabo omenjenih zbir ter vpelje lambda izraze.
- Modul 8:** Obravnavane teme so: *creating, using and constraining generics, using generic list collections* in *collection interfaces*.
- Modul 9:** Implementacija dogodkov in delegatov (*events and delegates*) ter njihova uporaba: *raising and subscribing to the events*.
- Modul 10:** Uporaba tako imenovanega *Language Integrated Query (LINQ)*, ki vključuje teme, kot so *selecting, ordering, groupping* in *joining data*.
- Modul 11:** Glavna tema modula je *Multitasking and Parallel programming*, obravnava pa: *multitasking, creating tasks, controlling tasks execution, returning*

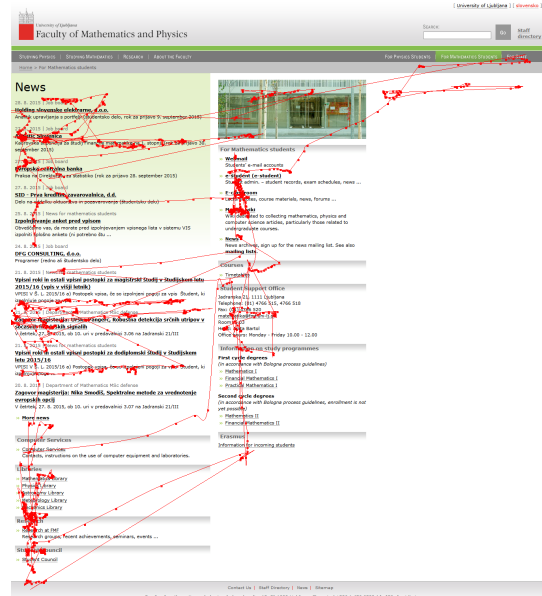
values from a task, canceling long running tasks, running tasks in parallel, using parallel LINQ, linking and nesting tasks, handling task exceptions.

**Modul 12:** Obravnava *Asynchronous programming*, ki vključuje naslednja podpodglavja: *using the Dispatcher, using Async and Await, awaitable methods and callbacks, APM operations, handling exceptions from awaitable methods.*

Po končanem spletnem tečaju sem se lotila pisanja programske opreme za napravo Tobii EyeX Controller. Nastali sta dve glavni verziji, ki se razlikujeta po izhodnem tipu aplikacije. Prva verzija je izhodnega tipa *Windows application*, druga verzija, ki je tudi nadgradnja prve, pa je tipa *Console application*.



(1.a) Testna fotografije verzije tipa *windows application*, ki zajame sliko zaslona, vključno z orodno vrstico operacijskega sistema.



(1.b) Testna fotografija verzije tipa *console application*, ki zajame le sliko celotne spletne strani brez orodne vrstice operacijskega sistema.

SLIKA 1. Testni fotografiji obeh verzij.

**Windows application:** Aplikacija najprej odpre okno, ki je v zgornjem delu opremljeno z navigacijskimi gumbi *Home*, *Back*, *Forward*, *Search* ter tekstovnim poljem, kamor uporabnik lahko vpiše URL naslov, v spodnjem delu okna pa se odpre brskalnik, ki se navigira na zahtevani začetni URL naslov. S pomočjo vgrajene programske opreme se najprej vzpostavi in zažene *eye-tracker*. Nato se naročimo na tok podatkov o smeri pogleda, ki je bil prav tako že implementiran s strani podjetja Tobii. Ob vsakem preteku določene časovne enote torej dobimo koordinate točke na zaslonu, ki jo je uporabnik gledal ob tem času. Zatem se v brskalniku naročimo na naslednje dogodke: *DocumentCompleted*, *Navigating*, *CanGoBackChanged* ter *CanGoForwardChanged*. Če se sproži dogodek *CanGoBackChanged* pomeni, da je v brskalni zgodovini na voljo prejšnja stran in zato takrat aktiviramo navigacijski gumb *Back*. Podobno aktiviramo navigacijski gumb *Forward*, ko se sproži dogodek *CanGoForwardChanged*. Ko uporabnik brska po spletu,

se koordinate dobljenih točk shranjujejo v seznam *points*. Implementirana je štoparica, ki vsakih 5 sekund točke seznama *points* izriše na zaslonsko sliko (ang. *screenshot*) in seznam sprazni. Spremembo URL naslova javi dogodek *Navigating*, ki se sproži, ko se brskalnik navigira na novo spletno stran. Takrat izrišemo točke iz seznama *points* na zaslonsko sliko prejšnje strani in seznam spraznimo. Ko se sproži *DocumentCompleted* dogodek pomeni, da se je nova spletna stran naložila in zato ponovno zajamemo sliko zaslona. Potrebno je omeniti, da ta verzija aplikacije zajame le vidno sliko zaslona skupaj z orodno vrstico operacijskega sistema, in ne slike celotnega dokumenta (spletne strani). Zato tudi ne upošteva morebitnih premikov po spletni strani. Testna fotografija je vidna na sliki (1.a).

**Console application:** Pri tej aplikaciji imata ključno vlogo knjižnici *Selenium* ([5]) in *PhantomJS* ([4]). *Selenium* je orodje, ki omogoča avtomatizacijo brskalnikov. Primarno je namenjen avtomatizaciji spletnih aplikacij za testiranje, vendar se uporablja na različnih področjih. Razvitih je že več projektov, ki omogočajo različne pristope. Knjižnica *PhantomJS* pa je orodje, ki se večinoma uporablja za nevidno/brezglavo testiranje spletnih aplikacij, za zajemanje zaslonskih slik, avtomatizacijo spletnih strani ter za nadzorovanje omrežja. Oboje sem uporabila pri implementaciji *eyetracker* aplikacije, ki sestoji iz razredov *Program* in *SnapshotGenerator* (za razliko od *Windows application*, ki sestoji iz enega samega razreda). Slednji poskrbi za izris točk in daljic (ali krivulje) na ustrezno zaslonsko sliko. Razred ima implementiran konstruktor s tremi parametri:

- URL naslov spletne strani, na katero želimo izrisati točke,
- seznam točk, ki jih želimo izrisati,
- pot do direktorija, kjer želimo shraniti ustrezno zaslonsko sliko.

Metoda, ki vrne datoteko z izrisanimi točkami in daljicami, v ozadju najprej odpre brskalnik, ki je uporabniku neviden. Brskalnik nato navigira na ustrezen URL naslov in s pomočjo *PhantomJSDriver*-ja zajame sliko celotnega dokumenta (spletne strani), ki jo shrani v PNG datoteko. Iz nje nato ustvari *Bitmap* objekt, na katerega najprej izrišemo zelene točke, nato pa ga shranimo nazaj v PNG datoteko. Razred *Program* pa skrbi za del aplikacije, ki je viden uporabniku. Enako kot pri *windows application*, se s pomočjo vgrajene programske opreme vzpostavi in zažene *eyetracker*, zatem pa se naročimo še na tok podatkov o smeri pogleda. S pomočjo *Selenium* knjižnice se odpre Chrome brskalnik, ki je viden uporabniku. Ko le ta brska po spletu, se koordinate dobljenih točk shranjujejo v seznam *points*. Na vsake 100 milisekunde se preveri, ali se je spremenil URL naslov v vidnem brskalniku. Če se je, se ustvari nov objekt tipa *SnapshotGenerator* z starim URL naslovom, seznamom *points*, ter potjo do zelenega direktorija, nato pa na njem pokličemo zgoraj opisano metodo za izris točk. Nazadnje izpraznimo seznam *points*, ter osvežimo zadnji URL naslov. Potrebno je še omeniti, da aplikacija upošteva pomike po posamezni spletni strani. Preden koordinate točke, dobljene iz zgoraj omenjenega toka, dodamo v seznam *points*, s pomočjo *javascript*-a ([2]) ugotovimo, kje na strani se uporabnik nahaja in nato prvotnim koordinatam prištejemo ustrezni vrednosti. Celotna koda vključuje *try/catch/finally* bloke, kjer ulovimo morebitne izjeme

in se s tem izognemu sesutju programa. Na sliki (1.b) je prikazana testna fotografija aplikacije.

**Komentar:** Pri analizi spletnih strani želimo uporabnika postaviti v njemu čimbolj domače okolje, zato je *console application* primernejša verzija. *Selenium* knjižnica poleg Chrome brskalnika omogoča uporabo tudi drugih brskalnikov, kot na primer Firefox, Safari, InternetExplorer, ... Naslednja faza pri razvoju programske opreme za Tobii EyeX Controller napravo je vzpostavitev podatkovne baze. Ta je namreč nujno potrebna, če želimo izvajati naprednejše analize spletnih strani, kot na primer tako imenovane *gaze plot* ali *heat map* fotografije. Tudi iz tega stališča je primernejša *Console application*, saj je implementirana tako, da omogoča preprost vnos podatkov v podatkovno bazo. Pred izrisom točk v razredu *SnapshotGenerator* se namreč generira JSON datoteka ([3]), ki vsebuje URL naslov spletne strani ter koordinate vseh točk, ki pripadajo danemu URL naslovu, in se jo da zlahka uvoziti v podatkovno bazo.

### 3. PRIDOBLEJENA ZNANJA

Na praktičnem usposabljanju sem pridobila predvsem dobro poznavanje programskega jezika *C#* ter spletnih aplikacij. Pri učenju jezika *C#* mi je koristilo poznavanje programskega jezika Python, ki sem ga pridobila v času študija, vendar pa Python spada tako med objektno kot tudi funkcijsko usmerjene programske jezike. Slednje sem med študijem že spoznala (Haskell, SML, Racket), s praktičnim usposabljanjem pa sem tako razširila tudi svoje znanje objektno usmerjenih programskih jezikov. Spoznala sem kar nekaj novih konceptov, najpomembnejši med njimi pa so obravnavanje izjem, dogodkov ter tako imenovani področji *multitasking* in *parallel programming*. Vse omenjene teme so bile ključne v razvoju opisane programske opreme. Poleg samega jezika, sem se naučila uporabljati program Visual Studio, ki je nasploh zelo razširjen pri razvoju različnih aplikacij. Pridobila sem znanje prej nepoznanih knjižnic Selenium ter PhantomJS in s tem tudi širši vpogled v spletne aplikacije ter njihovo testiranje. S seznanitvijo in razumevanjem delovanja naprave za sledenje očem sem spoznala tudi del širokega področja interakcije med človekom in računalnikom. Vso omenjeno pridobljeno znanje mi bo zelo koristilo tako na letošnji študijski izmenjavi, saj sem si izbrala predmet, ki zahteva znanje vsaj enega objektno usmerjenega programskega jezika, kot navsezadnje tudi pri kasnejši zaposlitvi.

### 4. PRILOGE K POROČILU

- Certifikat o opravljenem spletnem tečaju "Programming with *C#*".
- Potrdilo o opravljenem praktičnem izobraževanju.

### LITERATURA

- [1] *edX*, [ogled 30. 8. 2015], dostopno na <https://www.edx.org/>.
- [2] *Javascript*, [ogled 30. 8. 2015], dostopno na <https://www.javascript.com/>.
- [3] *JSON v programskem jeziku C#*, [ogled 30. 8. 2015], dostopno na [https://msdn.microsoft.com/en-us/library/cc197957\(v=vs.95\).aspx](https://msdn.microsoft.com/en-us/library/cc197957(v=vs.95).aspx).
- [4] *PhantomJS*, [ogled 30. 8. 2015], dostopno na <http://phantomjs.org/>.
- [5] *Selenium*, [ogled 30. 8. 2015], dostopno na <http://www.seleniumhq.org/>.
- [6] *Tobii EyeX Controller*, [ogled 30. 8. 2015], dostopno na <http://www.tobii.com/en/eye-experience/>.