

SPROVOĐENJE PRAKTIČNE NASTAVE PROGRAMIRANJA NA ELEKTROTEHNIČKOM FAKULTETU UNIVERZITETA U BEOGRADU U ONLINE OKRUŽENJU

Marko Mišić¹, Vladimir Jocović², Jovan Đukić³, Milana Prodanov⁴, Aleksa Srblijanović⁵,
Miloš Obradović⁶

^{1,2,3,4,5,6,7} Univerzitet u Beogradu, Elektrotehnički fakultet, Beograd, Srbija

¹marko.misic@etf.bg.ac.rs, ²jocke@etf.bg.ac.rs, ³dj@etf.bg.ac.rs, ⁴milana@etf.bg.ac.rs, ⁵aleksas@etf.bg.ac.rs,
⁶miobra@etf.bg.ac.rs

Kratak sadržaj: Praktična nastava iz oblasti programiranja se na Elektrotehničkom fakultetu Univerziteta u Beogradu u najvećem delu sprovodi kroz Praktikum iz programiranja 1 i Praktikum iz programiranja 2. Nastava se sastoji iz demonstracionih časova, domaćih zadataka i njihove odbrane u računarskim laboratorijama. Sve aktivnosti se sprovode za računarom. U okviru rada je prikazano sprovođenje ova dva predmeta u online okruženju korišćenjem Moodle platforme za elektronsko učenje i CodeRunner dodatka za automatizovano ocenjivanje programskog koda, što je izazvano aktuelnom pandemijom virusa COVID-19.

Ključne reči: programski jezik C, CodeRunner, praktična nastava programiranja, Moodle, Python

IMPLEMENTATION OF PRACTICAL PROGRAMMING EDUCATION AT THE SCHOOL OF ELECTRICAL ENGINEERING UNIVERSITY OF BELGRADE IN AN ONLINE ENVIRONMENT

Abstract: Practical classes in the field of programming at the School of Electrical Engineering, University of Belgrade are mostly conducted through Practicum in Programming 1 and Practicum in Programming 2. Practical work consists of demonstration classes, homework assignments and their defense in computer laboratories. All activities are carried out on the computer. The paper presents the implementation of these two courses in the online environment using the Moodle e-learning platform and the CodeRunner plugin for automated code evaluation, which is caused by the current pandemic of the COVID-19 virus.

Key Words: C programming language, CodeRunner, practical programming education, Moodle, Python

1. UVOD

Nastava programiranja na Elektrotehničkom fakultetu Univerziteta u Beogradu se izvodi na studijskim programima Elektrotehnika i računarstvo (ER) i Softversko inženjerstvo (SI) kroz obavezne predmete Programiranje 1 i Programiranje 2 na prvoj godini studija. U pitanju su masovni kursevi koje svake godine pohađa oko 900 studenata. Na predmetu Programiranje 1 se izučavaju simbolički mašinski jezik za školski računar – picoComputer (pC) i programski jezik Python, dok se na Programiranju 2 izučava programski jezik C, kao i određene teorijske teme iz oblasti programiranja i računarstva, poput elemenata algoritama i struktura podataka, arhitekture i organizacije računara, složenosti algoritam i sl. [1]

Oba predmeta imaju relativno ograničen fond časova za praktičan rad, pa stoga postoje predmeti Praktikum iz programiranja 1 (PP1) i Praktikum iz programiranja 2 (PP2) koji ne poseduju fond časova za predavanja i audiorne vežbe, već samo fond časova predviđen za praktičan rad i direktno se naslanjaju na dva pomenuta, obavezna predmeta. Na studijskom programu ER su ovi predmeti izborni i uobičajeno ih bira od 60 do 90 studenata, dok su na studijskom programu SI ovi predmeti obavezni i pohađa ih oko 200 studenata svake godine. Praktičan rad se na ovim predmetima već dugi niz godina sprovodi kroz demonstracione časove, domaće zadatke i odbranu domaćih zadataka u formi laboratorijskih vežbi. Glavne aktivnosti na predmetima tokom semestra čine dva demonstraciona časa i pet domaćih zadataka sa odbranama u računarskoj laboratoriji koji nose 70% ocene. U ispitnim rokovima se polaže ispit koji nosi 30% ocene. Tokom demonstracionih časova i laboratorijskih vežbi sa studentima interaguju predmetni nastavnici i saradnici, kao i studenti-demonstratori. Na taj način se, u živom kontaktu, sprovodi usvajanje praktičnih veština, novih koncepata, ali i ispitivanje teorijskih postavki, postavljanje novih zahteva za izmenu studentskih rešenja, otklanjanje grešaka i sl.

Međutim, zdravstveno-epidemiološka situacija izazvana pandemijom virusa COVID-19 i prelazak na nastavu na daljinu u martu 2020. godine su usloveli značajne promene u načinu sprovođenja aktivnosti tokom semestra, ali i ispita. Demonstracioni časovi su sprovedeni kroz snimljeni video materijal, dok su domaći zadaci rađeni, predavani i ocenjivani putem odgovarajućih testova korišćenjem Moodle platforme za elektronsko učenje sa kojom nastavnici i saradnici na predmetu već imaju određenog iskustva [8], kao i sa CodeRunner dodatkom za Moodle za automatizovano testiranje programskog koda [9].

Svaki test je obuhvatao automatsko ocenjivanje programskog koda putem CodeRunner dodatka korišćenjem javnih i tajnih testova, nekoliko pitanja iz relevantnog gradiva i u vezi sa domaćim zadatkom, kao i kratak programski problem u vezi sa obrađivanom tematikom koji su studenti rešavali samo prilikom predaje. Delimično su sprovedeni po uzoru na primere iz otvorene literature [10], kao i iz iskustava nastavnika sa predmeta stečenih tokom sprovođenja eksperimentalne, specijalne grupe za izučavanje programskog jezika Python koja je prethodila uvođenju u nastavu ovog programskog jezika [6][7].

S obzirom na skoro kompletno sprovođenje domaćih zadataka izvedeno na daljinu, pojačana je kontrola samostalnosti rada studenata korišćenjem sistema za detekciju sličnosti u programskom kodu sa čime autori takođe imaju prethodna iskustva [12][13], a na oba predmeta je sprovedena i bar jedna odbrana uživo u modifikovanom obliku u odnosu na raniji način. U okviru rada će detaljnije biti prikazano prethodno stanje i način održavanja nastave na oba predmeta PP1 i PP2 na daljinu u zimskom semestru školske 2020/2021. godine i letnjem semestru 2019/2020. Biće izvršena i preliminarna evaluacija uspeha studenata kroz nov način polaganja u odnosu na prethodne školske godine.

2. ORGANIZACIJA PRAKTIČNE NASTAVE U ONLINE OKRUŽENJU

Praktični rad na predmetima PP1 i PP2 sprovodi se kroz samostalnu izradu 5 domaćih zadataka u toku semestra i jednog većeg ispitnog zadatka, koji se može raditi u svakom ispitnom roku. Domaći zadaci su tematski orijentisani ka određenom delu gradiva, koje se u nedeljama do trenutka odbrane konkretnog domaćeg zadatka obrađuje na obaveznim predmetima Programiranje 1 i Programiranje 2. Svaki naredni domaći zadatak zahteva od studenta savladavanje koncepata predstavljenih prethodnim domaćim zadacima i na praktičan način utvrđuje znanje stečeno na obaveznim predmetima Programiranje 1 i Programiranje 2, koje ovi praktikumi prate. Tema prvog domaćeg zadatka na predmetu PP1 su linearno složeni problemi na simboličkom mašinskom jeziku za računar pC. Drugi domaći zadatak se tiče linearno složenih problema nad jednodimenzionim kolekcijama podataka na programskom jeziku Python. Treći domaći zadatak ilustruje rad sa višedimenzionim strukturama podataka, a četvrti domaći zadatak rad sa stringovima na programskom jeziku Python. Peti domaći zadatak demonstrira rad sa tokovima podataka (fajlovima), regularnim izrazima i obradom izuzetaka na programskom jeziku Python. Ispitni zadatak ima cilj da obuhvati modularne probleme predstavljene pojedinačnim domaćim zadacima na programskom jeziku Python. Od studenata se zahteva da na licu mesta izvrše dekomponovanje problema predstavljenog specifikacijom i inkrementalno izgrade svoje rešenje, uz testiranje i verifikaciju koda nakon svakog realizovanog potproblema. Ovo se postiže time što se studentima priznaju samo delovi koda koji poštuju specifikaciju i koji vrše traženu obradu u skladu sa specifikacijom. Na taj način se demotivise pisanje velike količine programskog koda koji ne izvršava ono čemu je i namenjen.

Pojava pandemije virusa COVID-19 uslovlila je značajne promene u sprovođenju organizacije kursa, imajući u vidu da su se u regularnom režimu rada odbrane domaćih zadataka sprovodile u laboratorijskim učionicama uz prisustvo predmetnih asistenata i studenata-demonstratora i živu interakciju studenata sa njima. Odlučeno je da kurs zadrži svoju organizaciju u pogledu broja i tema domaćih zadataka, ali je tranzicija ka *online* rešenjima bila neminovna. Doneta je odluka o migraciji kursa na Moodle platformu za elektronsko učenje sa kojom nastavnici i saradnici na predmetu već imaju određenog iskustva sa predmeta iz oblasti algoritama [8] i predmeta Programiranje 2[9], kao i za inkorporaciju dodatka za automatizovano testiranje programskog koda – CodeRunner [2].

Moodle platforma za elektronsko učenje obezbeđuje pravljenje tema u okviru kursa te je za svaki domaći zadatak izvršeno kreiranje po jedne teme radi lakše organizacije kursa. Takođe, obezbeđuje dodavanje raznovrsnih resursa (audio datoteka, video datoteka, dokumenata itd.) čime je obezbeđeno koncizno prezentovanje postavke studentima za svaki domaći zadatak. U okviru svakog domaćeg zadatka postoji nekoliko tematskih problema tako da se studentu na osnovu njegovog broja indeksa i godine upisa dodeli određena varijana zadatka. Svaki zadatak se modeluje barem jednim testom, pri čemu su testovi za izradu podeljeni u dve kategorije, a to su javni i tajni testovi.

Za svaki problem u okviru jednog domaćeg zadatka potrebno je napraviti po jedan javni i tajni test. Javni testovi sastoje se isključivo iz CodeRunner pitanja, pri čemu se takvo pitanje sastoji iz jednog ili više test primera, koji testiraju različite aspekte studentskog rešenja. Javni testovi su studentima na raspolaganju u periodu od 7 do 10 dana od datuma objave domaćeg zadatka. Svrha javnih testova je da izvrše evaluaciju studentskog programskog koda, čime se izbegava situacija u kojoj je rešenje studenta možda konceptualno dobro, ali ne ispunjava svoju svrhu. Studenti mogu da testiraju svoje rešenje proizvoljan broj puta, pri čemu dobijaju povratne informacije o sintaksoj i semantičkoj korektnosti njihovog rešenja, o eventualnom lošem rukovođenju resursima, kao i o eventualnoj prevelikoj složenosti njihovog rešenja. Tajni testovi sastoje se od većeg broja tipova pitanja koji se razlikuju, pri čemu je barem jedno CodeRunner tipa. Tajni testovi su studentima na raspolaganju u periodu onlajn odbrane domaćeg zadatka koja uobičajeno trajala četiri sata. Sami studenti su imali uobičajeno jedan sat da završe jednom započeti test. Studenti nisu upoznati sa ulaznim podacima tajnih test primera kojima se vrši testiranje njihovog rešenja, ali dobijaju povratne informacije koje im omogućavaju da lakše utvrde sintaksnu ili semantičku neispravnost u svom kodu. Ideja tajnih testova je da spreči studente da svoje rešenje prilagode specifičnim ulaznim podacima i na taj način se od studenta zahteva da napišu rešenje koje rešava opšti problem, a ne samo njegove specifične situacije.

Testove na Moodle platformi za elektronsko učenje sačinjavaju pitanja različite vrste. Najčešće korišćeni tip pitanja je obeležavanje tačnih odgovora, dopisivanje odgovora, spajanje odgovora iz dva bazena odgovora, kao i pisanje programskog koda i evaluaciju istog uz dodatne povratne informacije. Poslednje je omogućeno prethodnopomenutim

dodatkom CodeRunner za Moodle platformu. Veliki broj vrsta pitanja koje Moodle podržava, oblasti predviđenih programom predmeta i varijanti domaćih zadataka stvara pravu kombinatornu eksploziju pitanja u banci pitanja te se kvalitetna organizacija pitanja po kategorijama u okviru banke pitanja sama nameće. Kategorije pitanja u banci pitanja organizovane su po nivoima, pri čemu se na spoljašnjem nivou nalaze oblasti grupisane svakim od 5 domaćih zadataka. U okviru svake od tih 5 kategorija postoje kategorije pitanja sa javnim i tajnim testovima. Takođe, u svakoj od 5 kategorija koje predstavljaju domaće zadatke nalaze se i kategorije za oblasti koje svaki od 5 pojedinačnih zadataka grupiše. U okviru kategorija koje predstavljaju oblasti nalazi se još jedan nivo kategorija koje čine pitanja različitog tipa (npr. strogo teorijsko, strogo praktično, dopisivanje ili objašnjavanje delova koda). Za svaki tip pitanja postoji predefinisano šablonsko pitanje na osnovu koga se na jednostavan način pravi novo konkretno pitanje.

Iako Moodle platforma podržava mnoge tipove pitanja, neke aspekte ispitivanja studenata koje se sprovodi uživo nije moguće verno predstaviti nijednim tipom pitanja. To se najviše odnosi na "živu" komunikaciju nastavnika i saradnika sa studentima, tokom i nakon izrade rešenja. Poboljšanje epidemioloških mera omogućilo je sprovođenje ispita u laboratorijskim salama sa polovičnim kapacitetima. Ispitni zadatak obuhvata celokupno gradivo ispita i realizovan je u vidu testa, koji se sastoji od pojedinačnih programskih zadataka koji se tiču specifikacije jednog većeg problema. Test, koji je dostupan u određenom vremenskom periodu, započinje sam student, dok se test automatski završava nakon određenog vremena. Neminovno je da postoji zavisnost između pojedinačnih potproblema jednog složenog problema koji se rešava. Da se ne bi dešavale situacije u kojima student ne može da napreduje dalje od rešavanja nekog potproblema i samim tim bude ograničen u rešavanju sledećih potproblema koji (potencijalno) zavise od onih koje student ne može da reši, predmetni asistenti obezbeđuju programski kod koji se korektno izvršava, tako da potproblem koji student rešava i koji zavisi od prethodnih potproblema ne koristi realizacije potproblema studenta, već korektna rešenja predmetnih asistenata. Ispitivanje je značajno ubrzano i praktično obezbeđuje brzu verifikaciju rešenja studenta na arbitrarnom primeru koji vrši testiranje datog rešenja. Sa druge strane, omogućena je interakcija i diskusija oko načina realizacije problema i koje nije moguće ispitati napisanim test programom.

3. SKRIPTA ZA AUTOMATSKO OCENJIVANJE ZADATAKA

Važan segment u sprovođenju praktične nastave u *online* okruženju je činilo CodeRunner okruženje za automatizovano testiranje programskog koda [2] na Moodle platformi za elektronsko učenje. Inicijalno okruženje koje se instalira u vidu dodatka za Moodle platformu se sastoji od skripte na programskom jeziku Python koja procesira programski kod koji student unosi kao svoj odgovor. Skripta je sastavni deo odgovarajućeg tipa pitanja na Moodle platformi i može se menjati. Iako je dosta dobro napisano, samo okruženje je dosta izmenjeno od strane nastavnog osoblja, kako bi se bolje prilagodilo za testiranje koda napisanog na programskim jezicima Python i C.

Skripta je izmenjena tako da podržava sledeće dodatne funkcionalnosti: sakrivanje određenih testova (tajni testovi), raspoređivanje poena po celinama prilikom ocenjivanja, rad sa tekstualnim i binarnim datotekama, praćenje otvorenih datoteka, prosleđivanje argumenata komandne linije, upravljanje memorijom na programskom jeziku C, kako bi se signaliziralo eventualno curenje memorije, kao i sa tim povezano modelovanje otkaza. Takođe, za određene formate kraćih programskih pitanja je omogućeno zadavanje obaveznih ključnih reči u kodu rešenja, kao i zadavanje obaveznog ili zabranjenog koda.

4. ANALIZA USPEHA STUDENATA

Promene u načinu držanja nastave na predmetima PP1 i PP2 značajno utiču na sticanje poena i formiranje ocene studenata iz ovih predmeta. Kako bi se video uticaj ovih promena, prikupljeni su i analizirani uspesi studenata iz prethodne tri školske godine, a rezultati analize su dati u Tabeli 1 i Tabeli 2.

U letenjem semstru 2019/2020, odbrane prva četiri domaća zadatka na predmetu PP2 su organizovane na daljinu, dok je samo odbrana petog domaćeg zadatka organizovana na standardan način. Na osnovu analize rezultata predmeta koja pokazuje nešto lošiji učinak studenata na predmetu uprkos istom obimu gradiva kao i prethodnih školskih godina i jako dobrom uspehu istih studenata na predmetu PP1 iste godine, može se zaključiti da epidemiološka situacija prouzrokovana virusom COVID-19 značajno utiče na redovan rad i uspeh studenata. Međutim, dalja analiza rezultata pokazuje da su se studenti znatno bolje snašli na predmetu PP1 u 2020/2021 godini, iako se on izvodio u sličnom formatu kao predmet PP2 prethodne školske godine. Bolji uspeh je svakako uzrokovan i samim programskim jezikom Python koji se izučava na PP1 i koji se značajno lakše usvaja nego programski jezik C, što govore i naša pređašnja iskustva [6][7]. Na predmetu je odbrana prvog i petog domaćeg zadatka organizovana uživo u prostorijama fakulteta, dok je odbrana ostalih domaćih zadataka odžana na daljinu. Bolji rezultati studenata na predmetu PP1 u 2020/2021 godini u odnosu na rezultate studenata na predmetu PP2 u 2019/2020 godini govore da je konstatan rad, u poslednjih godinu dana, na poboljšanju nastavnih materijala koji su dostupni studentima, upoznavanja studenata sa radom na Moodle platformi, kao i rad na skripti za automatsko ocenjivanje pomogli studentima da bolje savladaju ovaj predmet uprkos teškim uslovima za rad u kojima se svi nalaze prouzrokovanih pandemijom virusa COVID-19. Treba napomenuti i da su ocene na predmetu PP1 u školskoj 2020/2021 godini prikazane samo nakon januarskog ispitnog roka i da se očekuje da određeni broj studenata popravi svoje ocene do kraja školske godine.

Tabela 1. Pregled raspodele ocena iz predmeta PP1 i PP2 u poslednje tri školske godine

Predmet	Praktikum iz programiranja 1						Praktikum iz programiranja 2					
	2018/2019		2019/2020		2020/2021		2017/2018		2018/2019		2019/2020	
Šk. godina	Br.	%	Br.	%	Br.	%	Br.	%	Br.	%	Br.	%
5	23	10.00	22	11.68	14	5.60	16	6.96	22	9.09	45	17.05
6	31	7.73	17	15.74	21	8.40	36	15.65	38	15.70	34	12.88
7	24	13.64	30	12.18	43	17.20	17	7.39	20	8.26	35	13.26
8	31	10.45	23	15.74	42	16.80	17	7.39	22	9.09	38	14.39
9	23	10.91	24	11.68	53	21.20	22	9.57	22	9.09	31	11.74
10	65	47.27	104	32.99	77	30.80	122	53.04	118	48.76	81	30.68
Ukupno	197	100	220	100	250	100	230	100	242	100	264	100

Tabela 2. Pregled prosečnog broja poena osvojenih na domaćim zadacima iz predmeta PP1 i PP2 u poslednje tri školske godine

Predmet	Praktikum iz programiranja 1			Praktikum iz programiranja 2		
	2018/2019	2019/2020	2020/2021	2017/2018	2018/2019	2019/2020
DZ1	6.90	6.88	6.61	6.89	6.95	6.33
DZ2	6.75	6.89	6.59	6.77	6.82	6.58
DZ3	6.77	6.84	6.41	6.78	6.58	5.66
DZ4	5.99	6.77	6.12	6.61	6.74	5.86
DZ5	6.43	6.69	5.51	6.54	6.53	5.69
Prosek	6.57	6.81	6.25	6.72	6.72	6.02

S obzirom da se usled pandemije virusa COVID-19 predaja i odbrana velikog broja domaćih zadataka izvedena na daljinu, pojačana je kontrola samostalnosti rada studenata korišćenjem sistema za detekciju sličnosti u programskom kodu. Uprkos povećanom broju studenata na predmetima i radu na daljinu, nisu uočena velika odustapanja u pokušaju plagiranja domaćih zadataka u odnosu na prethodne godine. Pretpostavlja se da ovaj podatak proizilazi iz činjenice da predmet nije obavezan za sve studente Elektrotehničkog fakulteta već da studenti koji su odabrali da ga prate žele da bolje savladaju gradivo koje se radi na obaveznim predmetima Programiranje 1 i Programiranje 2. Sa druge strane, studenti koji redovno odraduju sve obaveze predviđene predmetima PP1 i PP2 nemaju poteškoća sa polaganjem predmeta, pa shvataju da im nije u interesu bilo kakav pokušaj plagiranja domaćih zadataka na pomenutim predmetima.

5. ZAKLJUČAK

Zahvaljujući dostupnim platformama za elektronsko učenje i komunikaciju, kao i sistemima za automatizovano ocenjivanje programskog koda, sprovođenje praktične nastave iz oblasti programiranja na Elektrotehničkom fakultetu Univerziteta u Beogradu je izvedeno na izuzetno zadovoljavajući način. To pokazuje i uspeh studenata koji ne odstupa mnogo od prethodnih školskih godina, iako je online način ispitivanja relativno strožiji za studente, jer je ocenjivanje egzaktno. Utisak nastavnog osoblja je da su studenti morali da ulože dodatni napor kako bi prilagodili svoja rešenja traženim zahtevima i time dodatno unapredili svoje programerske veštine. Sa druge strane, sa stanovišta nastavnog osoblja, najveći nedostatak sprovođenja praktične nastave u *online* okruženju je izuzetno veliko vreme potrebno za pripremu odgovarajućih materijala i koje je ponekad i do tri puta duže u odnosu na vreme koje je ulagano tokom ranijih godina. Ipak, veći deo materijala koji je napravljen se može ponovo upotrebiti uz relativno male izmene, tako da uloženi napor svakako nije uzaludan.

6. LITERATURA

- [1] M. Mišić, M. Lazić, and J. Protić, *A software tool that helps teachers in handling, processing and understanding the results of massive exams*, Proceedings of the Fifth Balkan Conference in Informatics, ACM 2012, pp. 259-262.
- [2] R. Lobb and J. Harlow, *Coderunner: A tool for assessing computer programming skills*, ACM Inroads, vol. 7, no. 1, pp. 47-51, 2016.
- [3] D. Draskovic, M. Misic, Z. Stanisavljevic, *Transition from traditional to LMS supported examining: A case study in computer engineering*, Computer Applications in Engineering Education, Vol. 24, No. 5, Sep, 2016., pp. 775-786
- [4] V. Jocović, J. Đukić, M. Mišić, *First Experiences with Moodle and Coderunner Platforms in Programming Course*, Proceedings of the Tenth International Conference on e-Learning, Belgrade Metropolitan University, Belgrade, Sep, 2019., pp. 81-86
- [5] A. Radenski, *Python First: A lab-based digital introduction to computer science*, ACM SIGCSE Bulletin, vol 38, no. 3, ACM, 2006., pp. 197-201.
- [6] N. Vitorović, J. Protić, *Eksperimentalno uvođenje programskog jezika Python kao prvog programskog jezika za studente elektrotehnike i računarstva*, XX skup Trendovi razvoja: "Razvojni potencijal visokog obrazovanja", Kopaonik, 24. - 27. 02. 2014.
- [7] M. Mišić, S. Delčev, T. Šekularac, *Naučiti prvu Python*, Primena slobodnog softvera i otvorenog hardvera, Univerzitet u Beogradu - Elektrotehnički fakultet, Beograd, Oct, 2018., pp. 28-31
- [8] M. Mišić, Ž. Šuštran, J. Protić, *A Comparison of Software Tools for Plagiarism Detection in Programming Assignments*, International Journal of Engineering Education, Vol. 32, No. 2, pp. 738-748, Apr, 2016.
- [9] M. Mišić, J. Protić, M. Tomašević, *Improving source code plagiarism detection: Lessons learned*, 25th Telecommunication Forum (TELFOR), November, 2017. IEEE, pp. 1-8.