Medpredmetno povezovanje programiranja s strokovnim ali splošnim predmetom

Jan Robas

Šolski center Kranj
Slovenska cesta 27
4207 Cerklje na Gorenjskem

janrobas@gmail.com

**POVZETEK**

V tem prispevku raziskujemo možnosti za medpredmetno povezovanje pri strokovnem predmetu programiranje. Pri tem obravnavamo povezovanje med dvema strokovnima predmetoma ter med strokovnim in splošnim predmetom. Povezava programiranja z angleščino je pomembna, saj se pri stroki v prvi vrsti uporablja angleški jezik. Povezava z drugimi strokovnimi predmeti pa je pomembna, ker le tako znanje programiranja lahko resnično uporabimo za reševanje realnih problemov.

**Ključne besede**

Medpredmetno povezovanje, programiranje, angleščina, razvoj spletnih aplikacij.

**ABSTRACT**

In this paper, possibilities for cross-curricular integration with the programming course are explored. Options for integrating programming course with other professional subjects as well as with a general subject are explored in detail. Integration between programming and English is important because English is the primary language used in this field. Integration with other professional subjects is important because only such knowledge can be truly used to solve real-world problems.

**Keywords**

Cross-Curricular Integration, Programming, English, Web Application Development.

# UVOD

Današnji pristop k izobraževanju je tak, da je študij strogo razdeljen po predmetih. Učna snov je razdeljena na manjše dele oziroma predmete, pri čemer se vsak del načeloma izvaja neodvisno od drugega. Znotraj teh predmetov je snov razdeljena po urah. Pri tem je snov po urah znotraj enega predmeta ponavadi povezana in se logično nadaljuje oziroma nadgrajuje, učne snovi v različnih predmetih pa so morda sorodne, vendar so obravnavane ločeno. En predmet zato ponuja le del znanja, medtem ko reševanje realnega problema ponavadi terja celovito znanje, pridobljeno skozi različne predmete in izkušnje. Zaradi teh dejstev se poraja potreba po povezovanju predmetov med sabo.

# MEDPREDMETNO POVEZOVANJE

Za pridobitev celovitega znanja, ki je resnično uporabno, je potrebno obvladati več različnih disciplin. Za študijski program računalništva je smiselno, da se snov med predmeti nadgrajuje, tako da znanja, pridobljena pri enem predmetu, uporabljamo pri drugem [1]. Pri tem se lahko en predmet izvaja pred drugim in je uspešno opravljen izpit iz prvega predmeta predpogoj za pristop k drugemu predmetu. Primer je lahko predmet, kjer se študentje učijo razvoja mobilnih aplikacij. Predpogoj za ta predmet je lahko programiranje, saj razvoj mobilnih aplikacij terja znanje osnov programiranja. Ta predmeta sta lahko povezana tako, da se pri obeh uporablja isti programski jezik ali pa se pri predmetu razvoja mobilnih aplikacij na začetku razloži glavne razlike med že naučenim in novim programskim jezikom, ki ga bodo študentje uporabljali.

Druga možnost medpredmetnega povezovanja je povezovanje vsebin dveh predmetov, ki se izvajata vzporedno.

## Primer medpredmetnega povezovanja med programiranjem in razvojem spletnih aplikacij

Programiranje v prvem letniku je zelo splošen predmet, pri katerem se študenti seznanijo z osnovnimi pojmi in načinom razmišljanja, ki ga potrebujejo, da znajo rešiti osnovne programerske probleme. Reševanje osnovnih programerskih nalog brez povezovanja s širšim naborom znanj in realnim svetom samo po sebi ni zelo uporabno, vendar pa je obvladanje osnov predpogoj za nadaljnje učenje in delo. Ena od bolj konkretnih stvari, ki terja aplikacijo programerskega znanja, je razvoj spletnih aplikacij. Pri osnovah programiranja delamo s tekstovnimi izpisi v konzolo, pri čemer program poganjamo na klasičnih računalnikih, pri spletnih aplikacijah pa imamo grafičen uporabniški vmesnik, ki ga lahko uporabljamo tudi na mobilnih napravah, kot so telefoni in tablice.

Klasična spletna stran je statična, kar pomeni, da vsi obiskovalci vidijo enako vsebino in se le-ta ne spremeni, dokler lastnik spletne strani ročno ne spremeni vsebine in jo naloži na strežnik. Spletna aplikacija je spletna stran, katere vsebina se ob obisku sproti sestavi s pomočjo programa, ki teče na strežniku. Poleg tega lahko v vsebini vključuje tudi kodo, ki jo ob obisku spletne aplikacije požene odjemalec (brskalnik). Spletne aplikacije dandanes v veliki meri nadomeščajo namizne in mobilne aplikacije. Nekateri menijo, da bo v bližnji prihodnosti večina aplikacij temeljilo na spletnih tehnologijah [2]. S spletnimi tehnologijami lahko v današnjem času implementiramo vse vrste aplikacij, od sistema za vodenje knjižnice do računalniške igre. Znanje s področja razvoja spletnih aplikacij je torej zelo uporabno.

Spletna aplikacija se izvaja na strežniku in odjemalcu. Na strani strežnika imamo na voljo različne programske jezike (C#, PHP, Java, Python), na strani odjemalca pa ponavadi uporabljamo jezik JavaScript, ki ga moderni spletni brskalniki podpirajo. Razvoj spletnih aplikacij poleg programiranja vključuje oblikovanje in pisanje slogovnih predlog v jeziku CSS, uporabo označevalnega jezika HTML ter načrtovanje in uporabo podatkovnih baz. Ker gre za zelo obsežno snov in nadgradnjo znanja o programiranju, je znanje osnov programiranja ključnega pomena. Na sliki 1 lahko vidimo primer dela kode za izpis imen v konzolo in na sliki 2 primer kode za vključitev seznama z imeni v spletno aplikacijo. Oba primera sta napisana v jeziku C#, vendar pa je v primeru spletne aplikacije uporabljen tudi označevalni jezik HTML in jezik CSS. Pri obeh primerih je uporabljena zanka “foreach”, ki se jo študentje naučijo pri programiranju. Gre za poenostavljen primer.

Slika 1: Izpis imen študentov v konzolo.



Slika 2: Točkovni seznam z imeni študentov v modri barvi.



Ker je ustvarjanje programov, ki se zaganjajo v konzoli, za študente lahko suhoparno, študentom že pri programiranju na kratko predstavimo, kaj omogoča programiranje v povezavi s spletnimi tehnologijami in jih tako bolj navdušimo nad predmetom. Tistim, ki jih to zanima, lahko ponudimo dodatne naloge v povezavi s spletnimi tehnologijami ter jih tako pripravimo na snov izbirnega predmeta, ki jo bodo jemali po programiranju. Druga pomembna stvar je uskladitev kurikula tako, da je pri razvoju spletnih aplikacij na strani strežnika uporabljen isti ali vsaj čimbolj podoben programski jezik kot pri programiranju. S tem pri predmetu iz razvoja spletnih aplikacij prihranimo čas, ki bi ga sicer porabili za učenje osnov v novem programskem jeziku. Namesto tega se osredotočimo na ostala pomembna znanja, ki so specifična za predmet in tako pokrijemo večji del kompleksne snovi razvoja spletnih aplikacij.

## Primer medpredmetnega povezovanja med programiranjem in angleščino

Pri programiranju je angleščina prvi jezik. Ukazi in imena metod v standardnih knjižnicah so pri vseh priljubljenih programskih jezikih v angleškem jeziku. Prav tako so ponavadi v angleškem jeziku dokumentacija in orodja za pisanje programske kode (slika 3).

Slika 3: Opis metode standardne knjižnice znotraj razvojnega okolja Visual Studio.



Pri programiranju je pomembno razvijanje kompetenc skozi družbeni splet oziroma tako imenovani splet 2.0 [3].

Eden od načinov razvijanja programerskih kompetenc je preko družbenih omrežij in spletnih forumov. Eden od najbolj priljubljenih strokovnih družbenih omrežij je Stack Overflow. Stack Overflow je stran z vprašanji in odgovori, ki je specifično namenjena programerjem. Entuziasti in profesionalni programerji na tem omrežju najdejo rešitve za svoje programerske probleme tako, da bodisi najdejo odgovore na že zastavljena vprašanja bodisi postavijo svoje vprašanje. Ostali programerji lahko dobra vprašanja in dobre odgovore tudi ocenijo s plusom ali minusom. Za dobra vprašanja in odgovore tako dobiš točke. Šele ko imaš dovolj točk, se tvoje ocene pri drugih vprašanjih in odgovorih štejejo. Stack Overflow zahteva uporabo angleščine [4].

Zaradi tega tako v profesionalnem okolju kot pri izvajanju predmeta programiranje večkrat beremo dokumentacijo, odgovore na vprašanja in vodnike v angleškem jeziku. Pri tem je uporabljenih veliko strokovnih izrazov, ki so specifični za računalništvo. Včasih je težko najti ustrezen prevod teh izrazov v slovenščino, poleg tega pa je zaradi prej omenjenih dejstev pomembno, da študentje vedo strokovne izraze v angleščini. Čeprav je literatura za predmet v slovenskem jeziku, študentje dodatno razlago snovi iščejo tudi s pomočjo spletnih iskalnikov, kot naprimer Google, in pridobijo vsebino snovi z dodatnimi razlagami in primeri v angleščini. Viri na internetu so nepogrešljivi tudi pri nadaljnjem samostojnem izpopolnjevanju znanja, ki presega snov predmeta. Perspektivne študente, ki si želijo naučiti več, napotimo na spletne strani z angleško vsebino in jim jo pomagamo razumeti. Pomembno je, da pri tem sodeluje tudi učitelj angleščine tako, da se pri angleščini študentje spoznajo z nekaterimi strokovnimi izrazi. Poleg tega študentom, ki jim gre angleščina slabše, dodatno pomaga ter učitelja programiranja opozori o posebnih primerih, naprimer če ima kdo z razumevanjem angleščine težave.

# ZAKLJUČEK

Pri programiranju je medpredmetno povezovanje z drugimi vejami računalniške stroke in angleščine nepogrešljivo. Ko programiranje povežemo s snovjo ostalih računalniških predmetov, lahko dosežemo višji cilj. Z dobro načrtovanim medpredmetnim povezovanjem lahko bolje dosegamo cilje pri vseh predmetih, ki jih povezujemo. Poleg tega študenta naučimo teoretična znanja povezovati tudi v praksi, kar je ključni cilj kurikularnega povezovanja. Kljub vsemu pa medpredmetno povezovanje terja od učiteljev dodaten čas in napor. Zato je še posebej pomembno, da imamo pri tem dobro strategijo. Menimo, da z dobro strategijo in načrtom pozitivne stvari pretehtajo negativne.

# REFERENCE

1. Hadfield, S., Schweitzer, D., Gibson, D., Fagin, B., Carlisle, M., Boleng, J. and Bibighaus, D., 2011, October. Defining, integrating, and assessing a purposeful progression of cross-curricular initiatives into a computer science program. In 2011 Frontiers in Education Conference (FIE) (pp. T3J-1). IEEE.
2. Taivalsaari, A. and Mikkonen, T., 2011, August. The web as an application platform: The saga continues. In 2011 37th EUROMICRO Conference on Software Engineering and Advanced Applications (pp. 170-174). IEEE.
3. Rico, M., Coppens, J., Ferreira, P., Sánchez, H. and Agudo, J.E., 2013. Everything matters: Development of cross-curricular competences in engineering through web 2.0 social objects. In Ubiquitous and mobile learning in the digital age (pp. 139-157). Springer, New York, NY.
4. Stack Overflow, “Non-English Question Policy”. [Online]. Dostopno <https://stackoverflow.blog/2009/07/23/non-english-question-policy/>. [Povzeto 1. 9. 2019]