

SVEUČILIŠTE U SPLITU  
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**Pristup ostvarivanju prilagodljivosti prema  
znanju u sustavima za upravljanje učenjem**

Ivan Kuvačić

Split, svibanj 2015.

SVEUČILIŠTE U SPLITU  
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

**Pristup ostvarivanju prilagodljivosti prema  
znanju u sustavima za upravljanje učenjem**

**Student:**

Ivan Kuvačić

**Mentor:**

doc.dr.sc. Ani Grubišić

Split, svibanj 2015.

## **Zahvala:**

*Prije svega ovaj rad posvećujem mojim roditeljima koji su me bezuvjetno podržavali i podupirali kroz moje školovanje. Njihova upornost i vjera u mene jednaka je njihovoj ljubavi prema meni. Na jednak način se zahvaljujem bratu koji me je na mnoge načine podupirao i pomagao u mom školovanju i razvoju kao osobi. Nadam se da ću im se jednog dana dostojno zahvaliti.*

*Kroz rad sam shvatio koliko mi je drago vrijeme provedeno na fakultetu, druženje s kolegama koje smatram prije svega prijateljima, i kako se ovaj rad približavao kraju to sam sve više cijenio njihovo društvo kroz ovo predivno razdoblje života. Zahvaljujem se i profesorima koji su mi omogućili steći nova znanja.*

*Rad na diplomskom radu je bio dug (sad mi je žao što nije bio i duži), ponekad težak ali nikad mukotrpan i to zahvaljujući mentorici prof.dr.sc Ani Grubišić. Kroz rad mi je pokazalo što znači biti mentor, uvijek je imala vjeru u mene, čak i kad ja nisam imao vjeru da ću uspjeti. Istaknuo bih njeno veliko stručno znanje, ali bih još više istaknuo njenu susretljivost i razumijevanje. Prilikom zajedničkog rada uvijek me dočekala s osmijehom i besprijeekorno mi pomogla u mom profesionalnom razvoju te joj se ovim putem iskreno zahvaljujem i divim.*

## Sadržaj

1. Uvod.....	3
2. E-učenje .....	4
2.1. Sustav za upravljanje učenjem.....	7
2.2. Sustav za upravljanje sadržajem .....	10
2.3. Sustav za upravljanje sadržajem učenja.....	11
3. Oblikovanje nastave u sustavu E-učenja.....	13
3.1. Oblikovanje nastave.....	13
3.1.1. Razvoj oblikovanja nastave .....	14
3.1.2. Oblikovanje nastavnog sadržaja u sustavu za upravljanje učenjem .....	18
3.1.3. Sustav Moodle .....	21
3.2. ADDIE model za oblikovanje nastavnih sadržaja .....	26
3.2.1. Analiza .....	27
3.2.2. Oblikovanje.....	29
3.2.3. Razvoj .....	30
3.2.4. Implementacija.....	31
3.2.5. Vrednovanje .....	31
4. Oblikovanje prilagodljivog modela courseware-a .....	33
4.1. Područno znanje „Programiranje 1“ .....	33
4.1.1. Mapa koncepata „Programiranje 1“.....	34
4.2. Prilagodljivi courseware u inteligentnim tutorskim sustavima.....	37
4.2.1. Prilagodljivi courseware AC – ware Tutoru .....	39
4.3. Oblikovanje courseware-a.....	40
4.3.1. Elementi za učenje .....	41
4.3.2. Strukture statičkog računalom oblikovanog nastavnog sadržaja za stereotipove .	52
4.3.3. Klasteri.....	62

4.3.4.	Kriteriji za napredovanje prema znanju .....	70
4.3.5.	Kriteriji za napredovanje prema znanju .....	73
5.	Vrednovanje modela .....	77
6.	Zaključak.....	78
	Literatura.....	80
	Sažetak .....	81
	Summary .....	82
	Skraćenice .....	83
	Prilozi.....	84
	Prilog A - Elementi računalom oblikovanog nastavnog sadržaja „Programiranje“ .....	84
	Prilog B – Mapa koncepata.....	93
	Prilog C – Izlazni upitnik.....	94

# 1. Uvod

Nagli razvoj informacijske i komunikacijske tehnologije (engl. *Information and Communication Technology - ICT*), kao i povećana količina informacija dovela je do sve veće upotrebe računala u svim područjima ljudske djelatnosti, pa tako i u obrazovanju. Učitelj se prije smatrao glavnim izvorom informacija, a danas njegova uloga je znatno promijenjena s novom generacijom učenika koji su odrasli u znatno drugačijem vremenu sveprisutne računalne tehnologije.

Prednosti koje nosi novi način učenja su zabavni, dinamični, multimedijски i interaktivni nastavni sadržaji koji za razliku od klasičnih udžbenika nude učenje bilo kada i bilo gdje. Na temelju takvih potreba i zahtjeva razvio se novi način učenja, e-učenje, koje predstavlja presjek dvaju svjetova i to: svijeta informacijske i komunikacijske tehnologije i svijeta obrazovanja [Stankov, 2010]. E-učenje je nova paradigma, orijentirana prema učeniku (engl. *learner – centered paradigm*), a predstavlja proces obrazovanja uz pomoć multimedije i Interneta, te na taj način poboljšava kvalitetu procesa i ishoda obrazovanja.

Cilj ovog rada je istražiti modele prilagodljivosti u sustavima e-učenja, te na osnovu tih saznanja razviti i implementirati nastavni sadržaj na sustavu e-učenja, Moodle, koji bi imao karakteristike prilagodljivosti. Moodle je našao veliku primjenu u obrazovnim ustanovama ali kao sustav e-učenja nije „fleksibilan“ kao neki drugi sustavi. Ovaj rad pokušava iskoristiti funkcionalnosti sustava Moodle u cilju ostvarivanja prilagodljivosti nastavnog sadržaja učeniku, i to prema znanju.

Za rad je bilo potrebno istražiti i proučiti neke od sustava e-učenja s karakteristikama prilagodljivosti, odrediti modele oblikovanja nastavnog sadržaja, oblikovati nastavni sadržaj te ga vrednovati.



- E-učenje putem interneta omogućava interakciju između nastavnika i učenika, te učenika međusobno.
- Mogućnost integracije i pristupa drugim izvorima bitnima za gradivo koje se podučava.

Nedostaci [Wikipedia, E-učenje] koji se javljaju u E-učenju su:

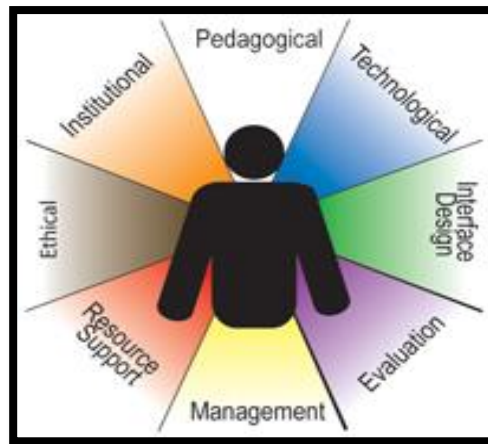
- E-učenje zahtijeva od učenika određena znanja i vještine kako bi se mogli njime koristiti.
- Veća odgovornost učenika jer učenik sam određuje kada će pristupiti učenju. Dakle ovo može biti prednost, ali i mana, ovisno o navikama učenika.
- Sustav zahtijeva određenu tehničku podršku i stalan razvoj u skladu s razvojem tehnologije kojom se koristi.

E-učenje nije alat, već vrsta komunikacijskog kanala kroz koji se odvija učenje. Poput komunikacije licem u lice, tiska, telefona, televizije ili audio-video sustava.

Khan [Khan, 2012] uvodi okvir okruženja E-učenja (engl. *framework for e-learning*) te smatra da ono mora odgovoriti na pitanje „*Što je potrebno da se osigura najbolje i najfleksibilnije obrazovno okruženje za učenike u svijetu?*“. Khan smatra da projektiranje, razvoj, implementacija i evaluacija predstavljaju otvorene, fleksibilne i raspodijeljene sustave učenja koji zahtijevaju promišljenu analizu i istragu o tome kako koristiti atribute i sredstva interneta i digitalne tehnologije u suradnji s obrazovnim načelima dizajna i pitanjima važnim za razne dimenzije online obrazovnog okruženja.

Okvir E-učenja pruža smjernice za planiranje i izradu materijala za E-učenja, organiziranje sredstava za okruženje E-učenja, dizajniranje raspodijeljenih sustava učenja, ocjenjivanje tečajeva i sustava E-učenja, te za projektiranje i ocjenjivanje okruženja za učenje.





Slika 2.2 Okruženje E-učenja [Khan, 2012]

Okruženje E-učenja po Khan-u formiraju pedagogija, tehnologija, korisničko sučelje, procjenjivanje, poslovanje, online podrška, etika i institucija zajedno sa atributima koje svaka od njih ima:

- *Pedagogija* predstavlja dimenziju koja povezuje e-učenje s procesom učenja i poučavanja. Temeljena je na analizi sadržaja, analizi cilja, analizi medija koji će se primjenjivati, pristup oblikovanja nastavnih sadržaja te medijima okruženja e-učenja. Metode mogu biti različite, kao što su: simulacije, rasprave u grupi, demonstracije, vježbe i ponavljanje te različiti načini interakcije
- *Tehnologija* predstavlja tehnologijsku infrastrukturu e-učenja. Najprije se odnosi na planiranje programske i tehničke podrške
- *Oblikovanje sučelja* je vidljivi dio sustava e-učenja koji se prezentira učeniku. Potrebno je postići da učenik navigacijom kroz sustav pristupa intuitivno te je potrebno prije toga napraviti test upotrebljivosti (engl. *usability*)
- *Vrednovanje* se vrši nad učenikom, tj. učinkom uspješnosti onog koji uči i poučava te vrednovanje okruženja učenja
- *Upravljanje* se odnosi na raspodjelu informacija koje su ključne za učenje
- *On-line podrška* predstavlja interaktivnu podršku koja objašnjava i potiče značaj i svrhu okruženja u kojem se obavlja učenje.
- *Etika* razmatra e-učenje prema socijalnom aspektu, te zemljopisne različitosti i različitosti onoga koji se poučava.

- *Dimenzija institucije* predstavlja administrativne poslove onog tko se poučava u odnosu na e-učenje

## 2.1. Sustav za upravljanje učenjem

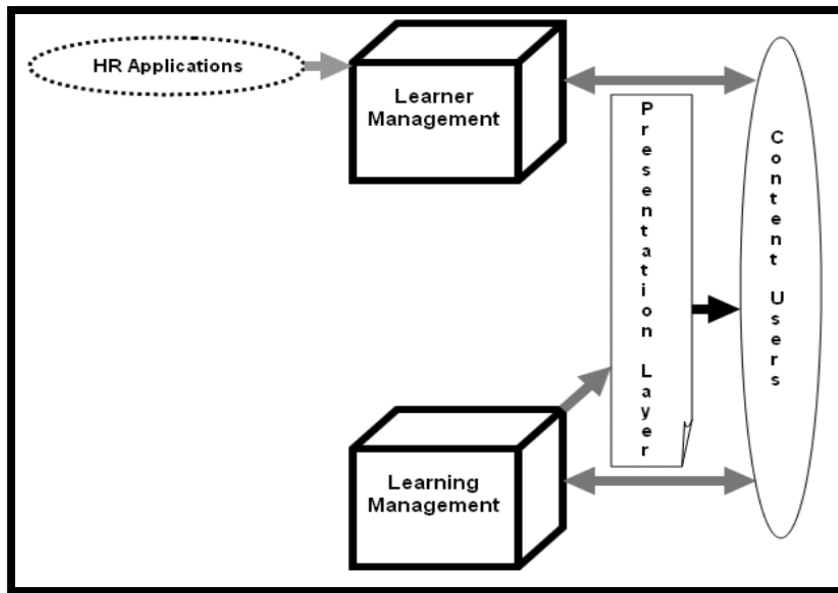
Sustav za upravljanje učenjem (engl. *Learning Management Systems - LMS*) [Nichani, 2001] predstavlja programsku podršku koja omogućuje korisniku potpuno administriranje i kontrolu procesa učenja i poučavanja. Sustav za upravljanje učenjem objedinjuje sve alate koji su potrebni za implementiranje "on-line razreda": izradu i dostavljanje sadržaja za učenje, kvizove i testove, komunikaciju, bilježenje podataka o studentima (engl. *logs*),...

Sustav za upravljanje učenjem (engl. *LMS- Learning management system*) je programska podrška koja automatizira administriranje procesa učenja i poučavanja [Stankov, 2010.]. Takav sustav karakterizira veliki broj komponenti za učenje koji su standardizirane. Neke od tih komponenti su: registracija učenika/učitelja, praćenje napretka u učenju, pristup nastavnim sadržajima, provjera znanja, dodjeljivanje određenih uloga – razina pristupa za pojedinog korisnika, te sigurnost podataka.

LMS sustavi na različite načine spremaju sadržaje koji se uče, dodatne informacije potrebne za učenje, kao i informacije o studentima. Funkcije LMS-a se mogu podijeliti u administratorske i nastavne, a samo sučelje u autorsko, koje služi za nastavnike koji generiraju sadržaj, te korisničko, koje služi učenicima za korištenje sustava.

Obično LMS ne uključuje u svojoj konfiguraciji autorske alate za stvaranje nastavnog sadržaja. Prodavači LMS sustava obično nude i dodatne alate za stvaranje nastavnog sadržaja. Ponovna upotrebljivost je na razini cijelog tečaja (jedan tečaj može se isporučiti većem broju učenika, a omogućeno je i praćenje postignuća).

Slika 2.3 prikazuje odnos komponenti koje čine sustav za upravljanje učenjem. LMS ima mogućnost vođenja učenika i njihovih zapisa, kao i mogućnost upravljanja procesom učenja. U okviru LMS-a, korisnici su u interakciji sa učeničkim podacima i podacima koji se koriste za upravljanje učenjem. Od velike pomoći učenicima u organizaciji i upravljanju znanjem su i programi ljudskih resursa (engl. *Human Resource - HR*).



Slika 2.3 LMS komponente [Irlbeck & Mowat, 2005]

Administratorske funkcije LMS-a obuhvaćaju sve evidencije podataka nužnih za funkcioniranje samog sustava, a koje nisu izravno vezane uz procese prijenosa znanja, i to evidenciju predmeta i korisnika sustava (učenika, nastavnika), prijavu korisnika, kreiranje dozvola i korisničkih grupa, te izvještavanje o napretku, statusu i rezultatima učenja samih korisnika.

LMS sustav koristi se za praćenje tijeka E-učenja dijeljenjem nastavnog sadržaja, a najvažnije nastavne funkcije su:

- Sadržaji za učenje kreiraju se u obliku lekcija i modula
- Navigacija služi za vizualnu pomoć pri kretanju lekcijama (gumbi standardiziranog dizajna)
- Provjera znanja putem testova i kvizova
- Komunikacija putem računala dozvoljava studentima i nastavnicima međusobnu komunikaciju koja može biti privatna i javna, te sinkrona i asinkrona
- Autorski alati koji omogućavaju pohranjivanje sadržaja za učenje na www serveru i njihovog odgovarajućeg povezivanja, te stvaranja testova i diskusija. Potrebno je naglasiti da ipak svi LMS sustavi ne omogućavaju potpunu autorsku okolinu.

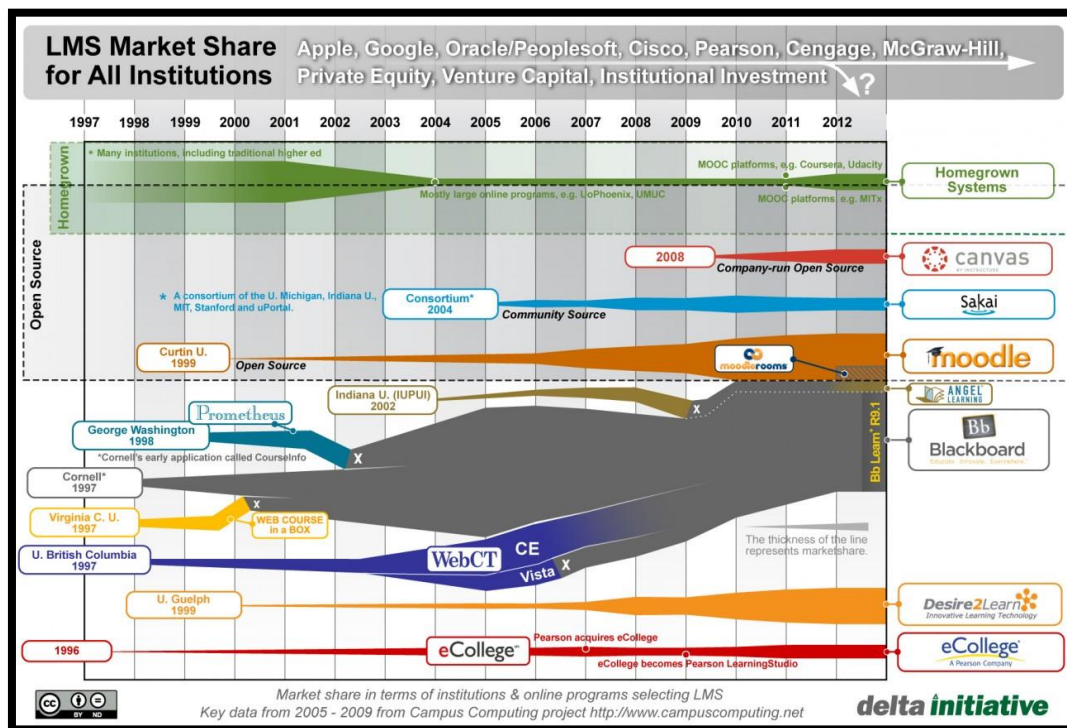
Koristeći ovakve sustave, učenik se može uključiti u proces nastave kada mu to odgovara, može odabrati modul za učenje i pristupiti nastavnim sadržajima. LMS kontrolira cjelokupni proces učenja na način da pohranjuje sve podatke vezane za cjelokupne aktivnosti učenika na

sustavu u svoju bazu podataka. Zbog toga učenik može dobiti kvalitetnu povratnu informaciju u pravo vrijeme, te istu tu informaciju učitelj koristi za praćenje napretka učenika. Tom činjenicom postupak učenja i poučavanja se znatno poboljšava.

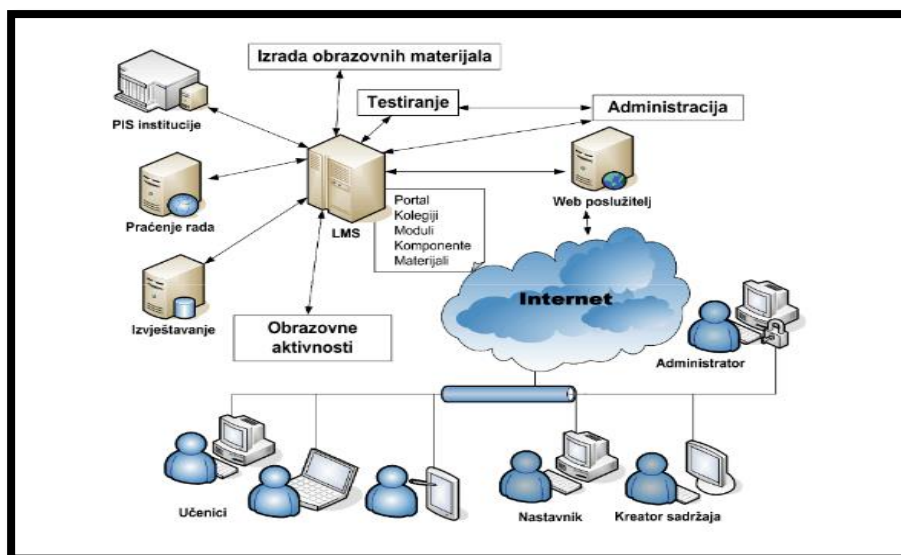
LMS se sastoji od nekoliko modula, a moguće ga je grupirati u tri razine (Slika 2.5):

- Osnovna razina sadrži prijavu, autorizaciju, dodjelu prava i statistiku;
- Sadržajna razina sadrži upravljanje sadržajem, praćenje i testovi;
- Razina učenja i komunikacije sadrži lekcije, novosti, forum, e-mail, chat.

U Svijetu postoji više od 70 LMS sustava koji su koriste u obrazovanju. Neki od tih sustava su komercijalni a neki besplatni i svi imaju razvijenu podršku zajednice. Iako neki imaju prednosti u odnosu na druge o tome sada nećemo pisati jer to nije centar ovoga rada. Na sljedećoj slici smo prikazali neke od najpoznatijih LMS sustava koja prikazuje njihov razvoj do 2012. godine.



Slika 2.4 Razvoj najpoznatijih LMS sustava [Phil Hill, 2012]



Slika 2.5 Primjer LMS modela [Nataša Hoić-Božić, 2009]

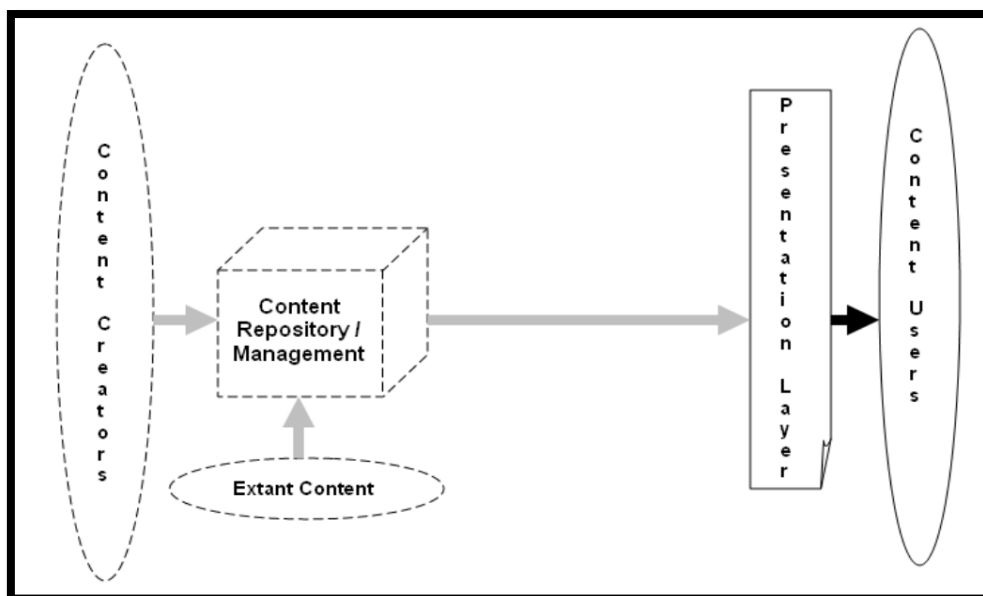
Najpoznatiji LMS sustavi su svakako: Moodle [[www.moodle.org](http://www.moodle.org)], Blackboard [[www.blackboard.com](http://www.blackboard.com)], Sakai [[www.sakaiproject.org](http://www.sakaiproject.org)], te Desire2Learn [[www.desire2learn.com](http://www.desire2learn.com)]. Primjenu su našli u mnogim školama i sveučilištima, te se njihov broj korisnika svake godine dosta povećava. Zanimljivo je napomenuti činjenicu da LMS sustavi otvorenog koda (engl. *Open source*) imaju veći porast broja korisnika i veću on-line zajednicu i razvijeniji „Help“.

U jednom od sljedećih odjeljaka fokus ćemo prebaciti na sustav *Moodle*. Na tom sustavu se i bazira ovaj red te svi nastavni sadržaji su razvijenu upravo na tom sustavu. Opisat ćemo i pobliže način razvoja nastavnog sadržaja na tom sustavu, prednosti sustava u odnosu na tradicionalni sustav poučavanja i arhitekturu sustava *Moodle*.

## 2.2. Sustav za upravljanje sadržajem

Sustav za upravljanje sadržajem (engl. *Content Management System - CMS*) [Nichani, 2001] je sustav za upravljanje web sadržajima odnosno izraz za programirane web stranice što nam omogućava da jednostavno sami unosimo, mijenjamo ili brišemo sadržaj sa svojih stranica. Termin CMS potekao je iz online izdavačke industrije koja omogućava oblikovanje i administriranje različitih sadržaja (članaka, reportaža, slika, transparenata i tome slično).

Slika 2.6 prikazuje komponente koje sadrži tipični sustav za upravljanje sadržajem, te ilustriraju osnovnu prirodu komponenti sadržaja i proces za kombinaciju sadržaja. U suštini, sadržaj je izrađen u obliku kompatibilnog spremnika samog sadržaja sustava. Digitalni oblik prezentacije sadržaja omogućava korisnicima traženje i pregledavanje pojedinih dijelova. Korisnici sadržaj obično ne mijenjaju, ali upozoravaju oblikovatelje nastave ako postoji potreba za promjenom.



2.4 CMS komponente [Irlbeck & Mowat, 2005]

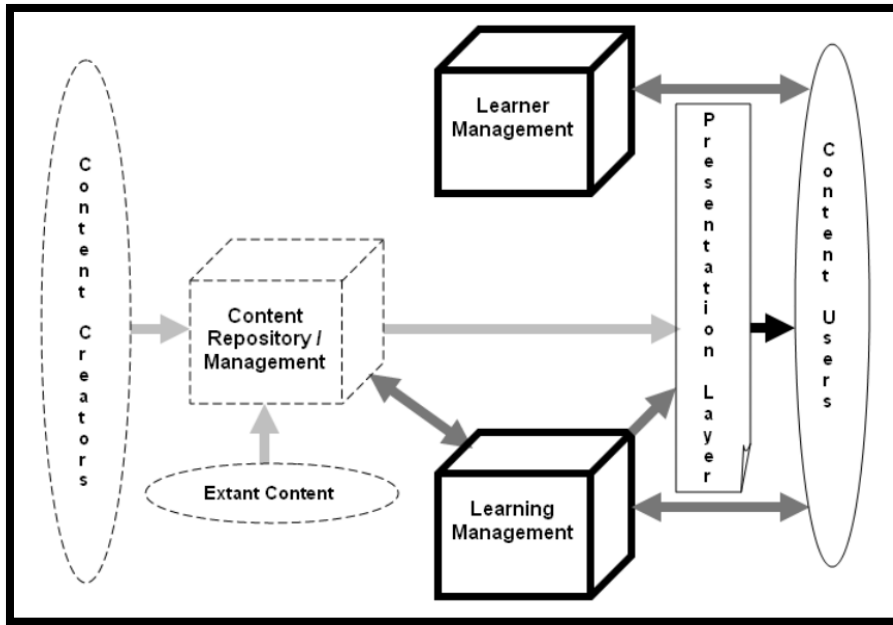
Osnovni dijelovi CMS-a [Maričić & Bandalo] su integracija sadržaja (manipulacije sadržajima od unosa, obrade pa sve do prikaza), kontrola verzija (osigurava da dva autora ne mogu mijenjati isti dokument u isto vrijeme), tok rada (omogućava definiranje procesa koji se tiču kreiranja, pregleda i odobravanja objavljivanja sadržaja) i repozitorijem sadržaja (pomaže u održavanju i upravljanju velike količine različitih sadržaja).

### 2.3. Sustav za upravljanje sadržajem učenja

Sustav za upravljanje sadržajem učenja (engl. *Learning Content Management Systems - LCMS*) [Nichani, 2001] omogućava upravljanje oblikovanjem, pospremanjem, upotrebom i ponovnom upotrebom sadržaja za učenje.

LCMS kombinira administrativne i upravljačke dimenzije tradicionalnog LMS-a uz stvaranje sadržaja. Struktura LCMS sustava se može promatrati i kao nadogradnja strukture LMS sustava kojem se dodaje sustav za upravljanje sadržajem ili ponovno upotrebljivi objekti učenja (engl. *Reusable learning objects - RLO*).

Slika 1.3 prikazuje elemente koje čine tipični sustav za upravljanje sadržajem učenja. Sadržaj je stvoren i pohranjen u spremište kojem se pristupa putem sustava za upravljanje učenjem i dijeli dalje korisnicima (učenicima). Podacima svakog učenika upravlja sustav te su pristupačni za pojedinačnog korisnika.



2.5 LCMS komponente [Irlbeck & Mowat, 2005]

### 3. Oblikovanje nastave u sustavu E-učenja

Tradicionalno poučavanje, kod kojeg u razredu na jednog učitelja dolazi oko 20-30 učenika, u današnje vrijeme postaje veliki problem. Razlog tomu je svakako održavanje motivacije i interesa kod učenika za nastavni sadržaj koji moraju usvojiti tokom obrazovanja.

E-učenje predstavlja brak tehnologije i obrazovanja [Siemens, 2002], a najčešće, oni koji oblikuju nastavu (engl. *Instructional designer*) imaju važnu ulogu u ovom „braku“ koja se očituje u premošćivanju koncepata između ova dva svijeta. Važna uloga osigurava ispravnost razvoja zamisli i koncepata od strane onih koji grafički oblikuju okruženje učenja kao i od strane programera profesionalaca. Nažalost, uloga nastavnog oblikovanja (engl. *Instructional Design - ID*) u E-učenju je često pogrešno shvaćena zbog uočene složenosti procesa i slabog poznavanje pedagoških zahtjeva E-učenja. U velikoj mjeri, ID je proces u kojem se učenje, a ne tehnologija, čuva u fokusu razvoja E-učenja.

Postoje mnoge definicije za oblikovanje nastave, te su sve izraz temeljne filozofije i pogleda na ono što je uključeno u proces učenja. Razlikovanje temeljne filozofije učenja može pomoći instruktorima i onima koji oblikuju nastavu da odaberu model oblikovanja koji je najviše u skladu s njihovim filozofijama učenja. Zbog toga oblikovanje nastave tj. dobro oblikovani nastavnih sadržaj za isporuku u sustavu E-učenja moraju odgovoriti na pitanja:

- Kako se događa učenje?
- Koji čimbenici utječu na učenje?
- Koja je uloga pamćenja?
- Kako se događa transfer?
- Koje vrste učenja su najbolje objašnjene uz pomoć teorije učenja?

#### 3.1. Oblikovanje nastave

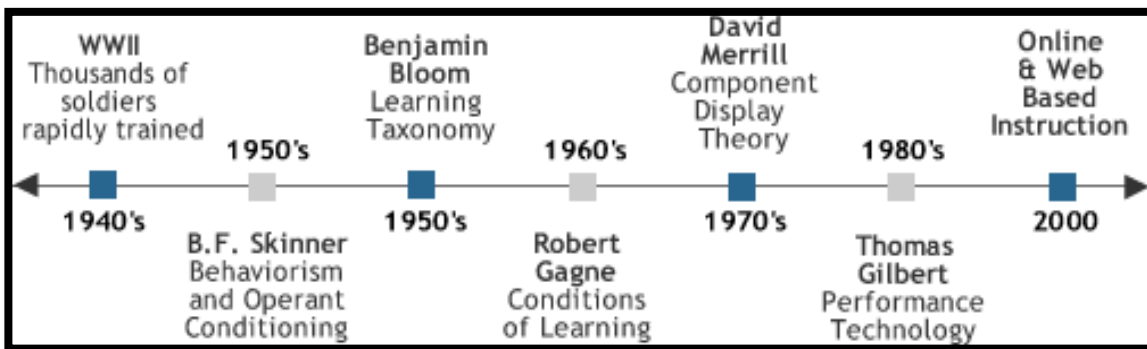
Oblikovanje nastave je umjetnost i znanost o stvaranju nastavnog okruženja i materijala koji će učeniku pomoći ostvariti određene zadatke. Oblikovanje nastave temelji se na teoretskom i praktičnom istraživanju u području spoznaje, obrazovne psihologije i rješavanju problema. [Siemens, 2002]

Sara McNeil [*Edutech Wiki, Instructional design, 2012*] definira oblikovanje nastave kao područje utemeljeno na znanju, te područje uključeno u istraživanje i teoriju o nastavnim strategijama, ali i kao proces za razvoj i provedbu tih strategija, i to iz 4 različite perspektive:



- *Oblikovanje nastave kao proces:* Oblikovanje nastave predstavlja sustavni razvoj nastavnih specifikacija pomoću teorije učenja i teorije nastave kako bi se osigurala kvaliteta nastave. Oblikovanje nastave podrazumijeva proces analize obrazovnih potreba i ciljeva te razvoj sustava za isporuku. To je proces koji uključuje razvoj nastavnih sadržaja i aktivnosti, te vrednovanje svih uputa i aktivnosti učenika.
- *Oblikovanje nastave kao disciplina:* Oblikovanje nastave je područje utemeljeno na znanju, te područje uključeno u istraživanje i teoriju o nastavnim strategijama, ali i kao proces za razvoj i provedbu tih strategija.
- *Oblikovanje nastave kao znanost:* Oblikovanje nastave je znanost o stvaranju detaljne specifikacije za razvoj, implementaciju, vrednovanje i održavanje situacija koje olakšavaju učenje nastavnih sadržaja.
- *Oblikovanje nastave kao stvarnost:* Oblikovanje nastave može započeti u bilo kojoj točki procesa oblikovanja. Uzimajući u obzir vrijeme, cijeli proces je izveden od strane onoga koji oblikuje nastavu gledajući unatrag i provjeravajući jesu li svi dijelovi znanosti uzeti u obzir.

### 3.1.1. Razvoj oblikovanja nastave



Slika 3.1 Razvoj nastavnog oblikovanja

[[http://www.instructionaldesigncentral.com/htm/IDC\\_instructionaltechnologytimeline.htm](http://www.instructionaldesigncentral.com/htm/IDC_instructionaltechnologytimeline.htm)]

*1940-te - Počeci oblikovanja nastave, Drugi svjetski rat* [Wikipedia, Instructional Design]

Pojava oblikovanja nastave povezuje se sa fenomenom iz II. Svjetskog rata kada je američka vojska trebala osposobiti veliki broj ljudi za obavljanje složenih stručnih poslova. Testovi za procjenu sposobnosti korišteni su za testiranje kandidata za programe osposobljavanja. Nakon

uspjeha vojne obuke, psiholozi su počeli promatrati obuku kao sustav, te su razvili razne analize i evaluacijske postupke.

- *1946.godina - Edgar Dale-ov stožac iskustva*

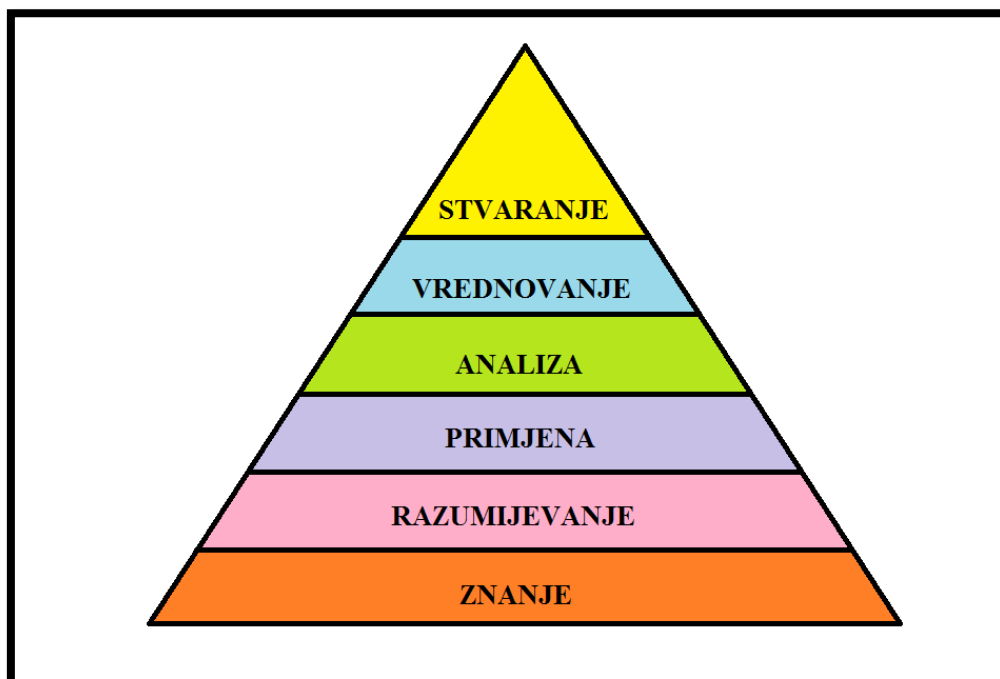
1946. godine Dale navodi hijerarhiju nastavnih metoda i njihovu učinkovitost.

- *Sredina 1950-ih do sredine 1960-ih – Programirana nastava pokreta*

U Skinner-ovom članku „*Znanost o učenju i umjetnosti podučavanja*“ iz 1954. godine, navodi se da se u nastavni materijal, nazvan programirani nastavni materijal, trebaju uključiti mali koraci, česta pitanja, neposredne povratne informacije, te da pojedinac sam sebi određuje tempo.

Robert Mager popularizirao je korištenje ciljeva učenja sa svojim članak „*Priprema ciljeve za programiranu nastavu*“ iz 1962. godine. U članku se navodi kako pisati ciljeve, te kakvo treba biti poželjno ponašanja.

1956. godine odbor na čelu s Benjaminom Bloom-om objavljuje taksonomiju tri područja učenja: kognitivnog područja (što se zna ili misli), psihomotornog područja (što se radi, fizički) i afektivnog područja (što se osjeća). Ove taksonomije i dalje utječu na oblikovanje nastave.



Slika 3.2 Bloom-ova taksonomija

- *Početak 1960-ih – Kriterijima ukazivano testiranje pokreta*

Robert Glaser je prvi upotrijebio izraz „*Mjere ukazane kriterijima*“ u 1962. godini. Za razliku od normativno ukazivanih testova u kojima se mjere izvedbe pojedinca u odnosu na izvedbu grupe, kriterijima ukazivano testiranje je oblikovano za testiranje ponašanja pojedinca u odnosu na objektivni standard. Ona se može koristiti za procjenu ponašanja ulaska učenika na određeni nivo, te da se pokaže u kojoj su mjeri učenici ovladali pojedinim vještinama.

- *1965.godina - Domene učenja, događaji u nastavi, te hijerarhijska analiza*

1965. godine Robert Gagne je opisao pet područja ishoda učenja i devet događaja nastave u „*Uvjeti učenja*“, koji postaju temelji prakse oblikovanja nastave.

Gagne-ov rad u učenju hijerarhije i hijerarhijske analize dovodi do važnog pojma u nastavi.

- *1967.godina – Formativno vrednovanje*

1967.godine nakon neuspjele analize materijala za obuku, Michael Scriven predlaže formativno ocjenjivanje - npr., isprobavanje nastavnih materijala s učenicima (i izmjene u skladu s tim) prije nego što se daju na uvid u razred.

- *1970-te - Rast interesa za sustavni pristup*

Tijekom 1970-ih godina, broj modela oblikovanja nastave uvelike se povećao, te se pokazao dobrim u različitim djelatnostima vojske, znanosti i industrije. Mnogi teoretičari oblikovanja nastave počeli su usvajati pristup oblikovanju nastave temeljen na obradi informacija. Tako je David Merrill razvio *Component Display Theory (CDT)*, koja je usredotočena na način prezentiranja nastavnih materijala (tehnika prezentacije).

- *1980-te - Uvođenje osobnih računala u proces oblikovanja*

Tijekom ovog desetljeća, dok je interes za oblikovanje nastave i dalje jak u poslovanju i vojsci, dogodio se razvoj ID-a u školama i visokom obrazovanju.

To je doba, kada su nastavnici i istraživači počeli razmišljati o tome kako se osobno računalo može koristiti u obrazovnom okruženju i veliki naponi su uloženi za oblikovanje uputa za korištenje novog alata. PLATO (*Programmed Logic for Automatic Teaching Operation*) je jedan od primjera kako su se računala počela koristiti u nastavi. Za prvu upotrebu računala u učionici poslužila je vježba „*vještine bušenja*“. Računalo temeljeno na edukativnim igrama i simulacijama također je postalo jako popularno.

To je vrijeme kada raste interes o tome kako se kognitivna psihologija može primijeniti na oblikovanje nastave. U kasnim osamdesetima i tijekom devedesetih godina teorija kognitivnog opterećenja pronašla je empirijsku potporu raznih tehnika prezentacije.

- *1990-te - Sve veći interes za konstruktivističku teoriju i isticanje važnosti izvedbe*

Kako se konstruktivistička teorija počela „probijati“, tako je i njezin utjecaj na oblikovanje nastave postao izraženiji u tradicionalnoj teoriji kognitivnog učenja. Konstruktivisti vjeruju da iskustva učenja trebaju biti „vjerodostojna“ i da trebaju stvoriti uvjete za učenje u stvarnom svijetu kako bi učeniku pomogli u izgradnji znanja. Ovaj naglasak na učenika predstavlja značajno odstupanje od tradicionalnih oblika oblikovanja nastave.

Još jedan trend koji se pojavljuje u tom razdoblju je prepoznavanje unapređenja sposobnosti kao važnog ishoda učenja koji je potrebno uzeti u obzir tijekom cijelog procesa oblikovanja.

World Wide Web je razvijen i pojavljuje se kao potencijalni online alat za učenje s hipertekstom (engl. *hypertext*) i hipermedijom (engl. *hypermedia*), te je prepoznat kao dobar alat za E- učenje.

- *2000-te - Razvoj interneta i online učenje*

Internet je sa svojim društvenim medijskim alatima i mnoštvom izvora informacija, postao vrlo popularan alat za E-učenje, a nastavni oblikovatelji prepoznali su potrebu za integriranjem E-učenja prilikom stvaranja objekata učenja, te nastavnih planova i programa.

Porast broja online kolegija na visokim učilištima.

Tehnologija napreduje i time učenicima osigurava više autentičnih i realnih iskustava učenja.

- 2010.godina i dalje

Utjecaj E-alata i dalje raste te potiče rast neformalnog učenja kroz život. Izazov za oblikovatelje nastave je kako stvoriti mogućnosti učenja koje se sada može odvijati bilo gdje i bilo kada.

### 3.1.2. Oblikovanje nastavnog sadržaja u sustavu za upravljanje učenjem

Tradicionalno poučavanje, kod kojeg u razredu na jednog učitelja dolazi oko 20-30 učenika, u današnje vrijeme postaje veliki problem. Razlog tomu je svakako održavanje motivacije i interesa kod učenika za nastavni sadržaj koji moraju usvojiti tokom obrazovanja.

Svaki učenik se mora izboriti za svojih „pet minuta“ ukoliko to želi, a često se događa u velikim razredima da učenik pasivno prati nastavu, a učitelj to ne primjećuje (slika 3.2). Nije dovoljno da učenik samo bude prisutan, već mora aktivno sudjelovati kako bi usvojio određena znanja i vještine. Kod takvog načina poučavanja važna je komunikacija kao i povratna informacija učeniku.



Slika 3.3 Tradicionalno poučavanje

[[www.srednja.hr](http://www.srednja.hr)]

Postoje brojni modeli oblikovanja nastave od kojih je ADDIE najpoznatiji model, a korijeni oblikovanja nastave potječu iz kognitivnih i biheviorističkih teorija učenja. U novije vrijeme i

konstruktivizam kao teorija učenja utječe na proces oblikovanja nastave. U e-učenju oblikovanja nastave strukturira tok obrazovnog procesa, tehnologija nije u prvom planu, već rezultati učenja i poučavanja. Možemo slobodno reći da je upotreba tehnologije kako bismo zadovoljili potrebe onoga koji se uči i poučava. Oblikovanje nastave možemo promatrati kao primjenu didaktičkih i metodoloških principa kako bi se ostvarilo učenje i poučavanje u ovom kontekstu putem računala.

Oblikovanje nastavnog sadržaja kod tradicionalnog poučavanja traži dosta vremena za pripremu nastavnog gradiva, kao i veliki napor prilikom tumačenja. Usporedimo li tradicionalnu nastavu s nastavom potpomognutu e-učenjem, uvidjet ćemo razne prednosti i nedostatke koje nosi novi način prenošenja znanja. Istaknimo da učenike sve više privlači multimedijalna prezentacija gradiva, ali takav način učenja zahtjeva više samostalnog rada što utječe na kvalitetu takvih nastavnih sadržaja.

Oblikovanje nastavnog sadržaja u sustavu za e-učenje nudi novi i poboljšani pristup. Takav pristup prikazat ćemo kroz sustav *Moodle*, čija riječ je akronim od izraza Modularno objektno-orijentirano dinamičko obrazovno okruženje (engl. *Modular Object-Oriented Dynamic Learning Environment*) koje pruža punu računalnu podršku pri organizaciji i izvođenju *online* kolegija [Bosnić, 2006.].

The screenshot shows a Moodle course interface. At the top, there is a header with the logo of SVEUČILIŠTE U SPLITU PRIRODOSLOVNO-MATEMATIČKI FAKULTET and contact information. Below the header is a breadcrumb trail: Moja naslovnica > Moji > Odj > Odj > Odj > TSEU. The main content area is titled 'Tehnologije sustava e-učenja' and contains several sections: 'Obavijesti', 'Uvod' (with a sub-section 'e učenje'), 'Web 2.0' (with a sub-section 'Web 2 0'), 'Semantički Web' (with a sub-section 'semanticki web'), and 'Primjena agenata u sustavima e-učenja' (with a sub-section 'Agenti u sustavima e-učenja'). On the right side, there are several widgets: 'NAVIGACIJA' (Navigation) with links to 'Moja naslovnica', 'Naslovnica sustava', 'Stranice na razini sustava', 'Moj profil', 'Trenutni kolegij', and 'TSEU' (with sub-links for 'Sudionici', 'Značke', 'Opći dio', 'Uvod', 'Web 2.0', 'Semantički Web', 'Primjena agenata u sustavima e-učenja', and 'Lokacijski zasnovane usluge i sustavi proširene st...'); 'POSTAVKE' (Settings) with links to 'Administracija kolegija', 'Ispiši me iz kolegija TSEU', 'Ocjene', and 'Postavke profila'; 'PRETRAŽI FORUME' (Search Forums) with a search box and 'Kreni' button; 'NOVE OBAVIJESTI' (New News) with a list of announcements including 'Sljedeće predavanje TSEU', 'Odgoda predavanja TSEU', and 'Predavanje profesora Ben Arija'; 'BUDUĆA DOGAĐANJA' (Future Events) with a calendar icon and text 'Nema predstojećih događaja'; and 'NEDAVNA AKTIVNOST' (Recent Activity) with a clock icon and text 'Aktivnost od Utorak, 24. Veljača 2015., 21:48'.

Slika 3.4 Izgled oblikovanog nastavnog sadržaja u sustavu *Moodle*

Moodle, sustav za upravljanje učenjem, omogućuje:

- pristup resursima
- rad s resursima na bilo kojem računalu koje ima pristup internetu
- učenje na daljinu
- oblikovanje nastavnih sadržaja
- interakcija korisnika sa sadržajima
- komunikaciju korisnika s drugim korisnicima
- izrada velikog broja tečaja na jednom sustavu
- planiranje tečaja – raspored aktivnosti, kalendar
- upravljanje korisnicima, korisničkim ulogama i grupama korisnika na tečaju
- rad s već postojećim datotekama i obrazovnim sadržajima
- provjeru znanja i ocjenjivanje korisnika
- praćenje aktivnosti korisnika
- mnogobrojne alate za komunikaciju i kolaboraciju među korisnicima
- upravljanje sustavom – sigurnosne kopije, statistike, logovi
- sustav pomoći

Razne funkcionalnosti, koje ovaj sustav nudi, imali smo priliku i sami koristiti u našem radu prilikom oblikovanja i vrednovanja nastavnog sadržaja. Nastavni sadržaj obogatili smo raznim multimedijalnim sredstvima, te na taj način poboljšali motivaciju učenika.

Na *slici 3.4.* možemo vidjeti strukturu jedne nastavne jedinice. Ona može sadržavati dodatne sadržaje za napredne učenike, domaću zadaću, radionicu, razne igre, kvizove, testove sa kojima se lakše prati usvajanje naučenog gradiva i sl. , što znači da se sustav može prilagoditi svakoj osobi. Učitelju je omogućeno praćenje napretka učenika, uvid u vrijeme provedeno na nekoj nastavnoj jedinici, kao i mogućnost komunikacije i suradnje u bilo kojem trenutku.

Koliko god su sustavi za e-učenje pogodni za individualizaciju učenja, smatra se da to ne bi smjelo u potpunosti zamijeniti učitelja, niti isključivati grupno poučavanje jer bi učenici bili uskraćeni za međusobnu komunikaciju, zajedničko rješavanje problema i suradničko učenje. Svakako, sustavi za upravljanje učenjem mogu biti dobra pomoć u nastavi ako se koriste na pravi način, a pristupanjem bazi podataka sustava može se upravljati cjelokupnim procesom nastave.

Sljedeći odjeljak posvetit ćemo arhitekturi sustava Moodle, te opisati na koji način svaka osoba može koristiti sustav na svom računalu te detaljnije opisati njegove dijelove, posebice bazu podataka.

### 3.1.3. Sustav Moodle

Sustav Moodle namijenjen je oblikovanju nastavnih sadržaja, te održavanju nastave na daljinu, a razvija se od 2002. godine. Njegov začetnik je Martin Dougiamas, administrator WebCT-a na tehnološkom sveučilištu u Curtinu. Danas ovaj sustav razvija mnoštvo stručnjaka, te brojni korisnici iz cijelog svijeta. Izdan je pod licencom *GNU Public License*, iako je zaštićen, korisnicima je dozvoljeno koristiti, kopirati i mijenjati kôd, ukoliko:

- omogućuje drugima korištenje kôda po jednakim uvjetima
- ne mijenjaju originalnu licencu ni zaštitu
- primijene istu licencu na bilo koji drugi rad koji iz Moodle-a proizlazi

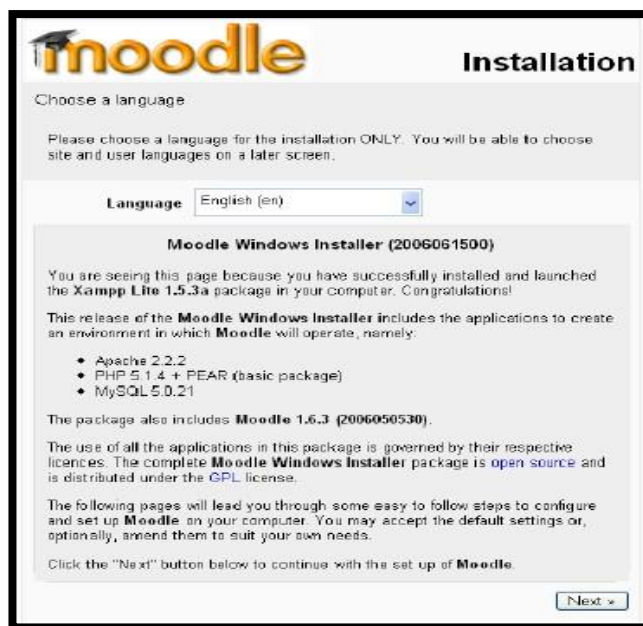
Aplikaciju se može besplatno preuzeti sa službene Moodle Web stranice. Dakle, sustav Moodle je računalni program otvorenog koda čija licenca omogućuje uvid u izvorni programski kod i njegovo modificiranje prema vlastitim potrebama. Ovaj sustav je doživio mnogo verzija, a posljednja verzija je 2.8.3+.

Napisan je u PHP-u, za rad koristi *Apache*, besplatni web poslužitelj otvorenog koda, a podržava više vrsta baza podataka (posebno MySQL i PostgreSQL). Sučelje je prevedeno na 65 jezika, koristi se u 239 država, ima 73,718,700 registriranih korisnika [www.moodle.org].

Zbog svoje fleksibilnosti i jednostavnosti, ovaj sustav trenutno je najpopularniji u Svijetu. Da bi ga korisnici mogli koristiti na svojim računalima potrebno je izvršiti njegovu instalaciju odabirom instalacijskog paketa, te *XAMPP* ili *WAMP* servera, bez kojih instalacija ne bi bila moguća.

Instalacija zahtjeva i dodatne postavke u *php.ini* datoteci, te kreiranje nove Moodle baze podataka. Za njezino uspješno kreiranje, potrebno je u Internet pregledniku (engl. *browser*) unijeti sljedeće: *localhost/moodle*, te na stranici odabrati opciju: *phpmyadmin*. PhpMyAdmin je besplatan alat napisan u PHP-u, a služi za upravljanje i administraciju MySQL-a preko World Wide Web-a, odnosno korisnik upravlja bazama podataka, tablicama, poljima, indeksima, korisnicima, dozvolama pristupa samim bazama, te izvršavanje vlastitih upita nad bazom. Na slici 3.5 možemo vidjeti kako izgleda početna stranica instalacije sustava Moodle.





Slika 3.5 Početna stranica instalacije sustava *Moodle*

Instalirani sustav Moodle na početku je „prazan“, odnosno nema nikakvih instaliranih modula, kao i prijavljenih korisnika, pa je sve to potrebno naknadno dodati ovisno o potrebama korisnika.

Baza podataka sustava Moodle sadrži preko 300 tablica, ovisno o instaliranim modulima, ali rad s njegovim tablicama je jednostavan korištenjem raznih računalnih programa. Za potrebe ovog rada korištena su dva računalna programa: *MySQL Workbench* koji omogućuje modeliranje podataka, SQL razvoj i sveobuhvatne alate za administraciju, te *dbForge Studio Express for MySQL* za upravljanje, razvoj i administraciju. Potrebno je povezati odabrani računalni program sa bazom podataka, a nakon toga korisnik se može upoznati sa organizacijom i strukturom same baze. Na slici 2.10 prikazan je programski alat *dbForge Studio Express for MySQL* povezan na bazu podataka sustava Moodle. Sa lijeve strane nalazi se popis svih tablica iz baze podataka, a sa desne strane je moguće unositi SQL upite kako bi upravljali podacima iz tih tablica. Ovaj programski alat nudi mogućnost automatskog generiranja SQL upita, povezivanjem tablica putem primarnog ključa.

Nakon uspješnog povezivanja programskih alata sa bazom podataka *Moodle* potrebno je upoznati se sa organizacijom i strukturom same baze. Zbog brojnosti tablica unutar baze podataka, baza je organizirana u grupe tablica nazvane klasterima (engl. *Clusters*) koji su organizirani prema svojim funkcionalnostima: Forum, Korisnici, Kviz, Uloge, itd. Postoji 37 grupa tablica, a naziv

svake tablice sastoji se od *prefiks\_nazivObjekta* gdje prefiks označuje grupu, a nazivObjekta atribute. Svaka od prethodno navedenih funkcionalnosti (Forum, Korisnici, Uloge, Kviz...) sadržavaju svoju skupinu tablica koje su međusobno povezane.

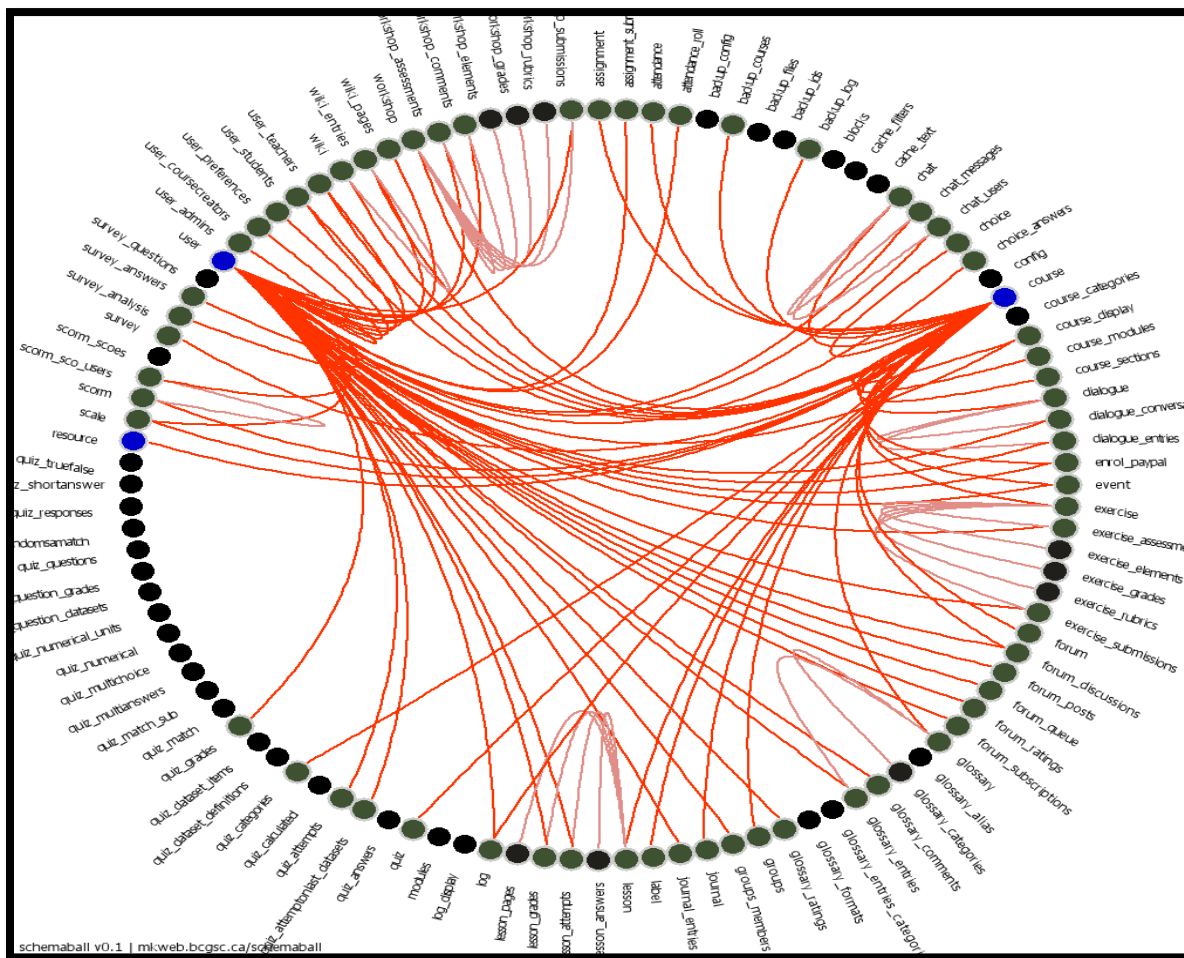
Sve grupe tablica prikazat ćemo u tablici koja slijedi, kao i njihov opis što predstavljaju i broj tablica koji se nalazi u pojedinoj grupi.

Naziv grupe	Namjena	Broj tablica
<b>Backups</b>	<i>Sigurnosna kopija podataka</i>	<b>8</b>
<b>Blocks</b>	<i>Upravljanje komandnim blokovima</i>	<b>8</b>
<b>Courses</b>	<i>Organizacija kolegija</i>	<b>15</b>
<b>Chat System</b>	<i>Omogućava protok informacija za chat</i>	<b>4</b>
<b>Roles</b>	<i>Omogućava dodjeljivanje uloga korisnicima</i>	<b>7</b>
<b>Messages</b>	<i>Sustav za komunikaciju između korisnika</i>	<b>9</b>
<b>Moodle Network (Mnet)</b>	<i>Dio arhitekture zadužen za mrežnu komunikaciju</i>	<b>13</b>
<b>Logging</b>	<i>Detaljna pohrana korisnikovih aktivnosti na sustavu</i>	<b>3</b>
<b>Repository</b>	<i>Predstavlja repozitorij – spremište resursa</i>	<b>3</b>
<b>Statistics</b>	<i>Tablice za vođenje detaljne statistike</i>	<b>6</b>
<b>Survey</b>	<i>Podsustav za rad s anketama</i>	<b>4</b>
<b>Tags</b>	<i>Grupa tablica za opis i korištenje opisnih oznaka</i>	<b>3</b>
<b>Users</b>	<i>Velika grupa tablica koja opisuje korisnike sustava</i>	<b>9</b>
<b>Forums</b>	<i>Grupa tablica koja omogućava rad foruma</i>	<b>7</b>
<b>Enrollments</b>	<i>Tablice koje opisuju uplate stvarnog novca kojima Moodle barata</i>	<b>5</b>
<b>External Services</b>	<i>Tablice za uspješan rad sa vanjskim servisima</i>	<b>5</b>
<b>Grades</b>	<i>Grupa namjenjena ocjenama</i>	<b>12</b>

<b>Portfolio</b>	<i>Opis portfolija korisnika</i>	<b>8</b>
<b>Context</b>	<i>Grupa tablica koje sadržavaju metapodatke o ostalim tablicama</i>	<b>7</b>
<b>Assignments</b>	<i>Zadace koje se mogu definirati</i>	<b>2</b>
<b>Configuration</b>	<i>Skup tablica za konfiguriranje</i>	<b>5</b>
<b>Blogs</b>	<i>Mala grupa tablica koja omogućuje definiranje blogova</i>	<b>3</b>
<b>Choices</b>	<i>Podgrupa tablica koja opisuje koncept izbora odgovora u pitanjima</i>	<b>3</b>
<b>Workshop System</b>	<i>Kompleksan sustav koji se bavi svim detaljima radionica</i>	<b>13</b>
<b>Data</b>	<i>Skup tablica koji opisuje tipove podataka</i>	<b>4</b>
<b>Miscellaneous</b>	<i>Razne tablice koje ne pripadaju niti jednoj specifičnoj grupi, već individualno predstavljaju neki objekt</i>	<b>17</b>
<b>Events</b>	<i>Tablice koje opisuju događaje koji mogu nastupiti</i>	<b>4</b>
<b>Lessons</b>	<i>Važna grupa tablica koja opisuje lekcije i resurse vezane za njih</i>	<b>8</b>
<b>Grouping</b>	<i>Tablice koje opisuju grupe i njihove članove</i>	<b>4</b>
<b>Feedback System</b>	<i>Sustav koji definira i obrađuje povratne informacije</i>	<b>9</b>
<b>Quiz System</b>	<i>Dio arhitekture koji omogućuje rad sa kvizovima</i>	<b>11</b>
<b>Questions</b>	<i>Dio posvećen definiranju i obradi pitanja</i>	<b>13</b>
<b>Question Types</b>	<i>Grupa tablica posvećena definiranju tipova pitanja</i>	<b>10</b>
<b>Glossary</b>	<i>Skup tablica posvećen unosima u razne opise</i>	<b>7</b>
<b>Wiki</b>	<i>Podskup tablica koji definira Wiki stranice unutar</i>	<b>8</b>
<b>Scorm integration</b>	<i>Sustav koji omogućava prilagodbu sadržaja SCORM normama</i>	<b>9</b>

Tablica 3.1 Popis svih grupa tablica iz sustava Moodle

Organizacija tablica unutar ovakve baze podataka je poprilično složena jer je potrebno poznavati strukturu svake tablice kako bismo uspješno mogli doći do potrebnih podataka temeljem SQL upita. Primijetimo na sljedećoj slici 3.6 oko kojih grupa je najveća povezanost tj. koje grupe su najvažnije u ovakvoj strukturi i na koje se veže većina drugih grupa i tablica.



Slika 3.6 Povezanost tablica unutar *Moodle* baze podataka

[[www.moodle.org](http://www.moodle.org)]

Slika 3.5 prikazuje nam strukturu baze podataka, te na njoj možemo vidjeti koje grupe tablica su najvažnije u bazi. Većina tablica spojena je na grupu *Korisnici* i na *Kolegiji*. Crvene linije predstavljaju veze između tablica, a najveća „gustoća“ tih linija je oko grupe *Korisnici*.

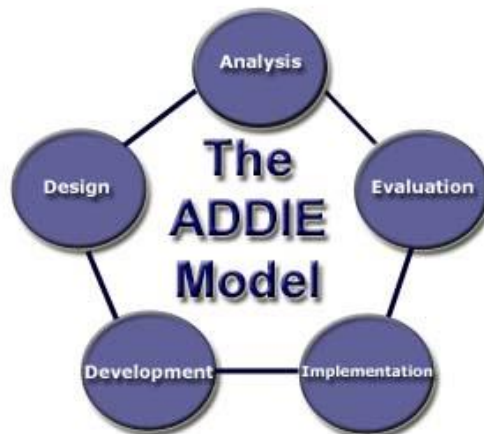
Kako bi analizirali podatke iz tablica baze podataka koristeći neki od programskih alata potrebno je imati definirano što nas točno zanima, kakve informacije želimo dobiti. Npr. Zanima nas koje uloge su dodijeljene pojedinim korisnicima, koliko poruka je razmijenjeno na forumu ili chatu, statistički podaci vezani za aktivnosti pojedinog korisnika, vrijeme provedeno učeći neku lekciju i sl. Na smislen i logičan način moramo „pročitati“ podatke iz povezanih tablica.

U sljedećem poglavlju ćemo detaljnije objasniti jedan od najčešće korištenih modela za oblikovanje nastavnih sadržaja. Također ćemo objasniti njegove faze te zašto se baš taj model često koristi.

## 3.2. ADDIE model za oblikovanje nastavnih sadržaja

ADDIE model je najpoznatiji model za oblikovanje nastavnih sadržaja. Sastoji od pet faza, a to su „Analiza“ (engl. *Analyze*), „Oblikovanje“ (engl. *Design*), „Razvoj“ (engl. *Development*), „Implementacija“ (engl. *Implementation*) i „Vrednovanje“ (engl. *Evaluation*). Vrednovanje je u centru modelu tako da se nakon završetka pojedine faze odmah pristupa vrednovanju, a tek onda se pristupa sljedećoj fazi modela.

Ovaj model ima široku primjenu i kroz sam razvoj modela postojalo je više varijanti. ADDIE model zapravo daje smjernice razvoja nastavnog materijala, plana i programa a jedna od najvećih prednosti u odnosu na ostale modele je to što se ovim modelom brzo dobije prototip i samim time se često dobivaju povratne informacije. ADDIE model u obrazovanju se koristi za oblikovanje nastave i uvelike pomaže u razvoju nastave, odnosno ADDIE pomaže u razvoju plana i programa. Svaka faza „postavlja“ nekoliko pitanja te mi sami odabiremo na koja pitanja hoćemo odgovor i u skladu s time razvijamo nastavni sadržaj. Dakle mi sami određujemo što nam je bitno prateći smjernice ADDIE modela. Svaka faza ADDIE modela je temelj prethodne faze, pa izbacivanje jedne od fazi može stvoriti negativne učinke tijekom cijelog procesa. ADDIE modelom se ostvaruje racionalni pristup oblikovanja nastave a cilj modela je ostvarivanje znanja i vještina znanja.



Slika 3.7 Grafički prikaz ADDIE modela [Chico State IDTS, 2009]



Tablica 3.2 Prikaz razvoja ADDIE modela [Clark, 1995]

Povijest razvoja ADDIE modela:

- 1975. godina: povijest ADDIE modela seže sve do 1950. godine ali godina koja zapravo predstavlja prekretnicu je 1975. Odsjek Centra obrazovnih tehnologija (The Centre for Educational Technology) Državnog sveučilišta u Floridi je razvio model za Američku vojsku [Branson, Rayner, Cox, Furman, kralj, Hannum, 1975]
- 1980-ih godina: Model dobiva novu verziju gdje se faze mijenjaju da bi bolje odgovarale potrebama organizacije.
- Sredina 1980-ih godina: Vojska je razvila novu verziju modela u kojem se model razvio iz vodopadnog u dinamički model. Posljednja faza je promijenila naziv iz „Vrednovanje i kontrola“ u „Vrednovanje“. [US Army 1984]
- 1995. godine: ADDIE je prvi put korišten kao akronim (Schlegel, 1995)
- 1997. godine: ADDIE model postaje široko prihvaćen te postaje baza svim ostalim modelima. Merrieboer ga je nazvao „plug and play“ modelom [Merrieboer, 1997]
- 2002. godina: model se razvio i počeo koristiti za dobivanje uvida u problem te njegovo rješavanje [DeSimone, Werner, Harris, 2002].

U daljnjem tekstu ukratko je opisano što svaka od navedenih faza uključuje.

### 3.2.1. Analiza

Analiza predstavlja prvi i najvažniji korak u oblikovanju nastave. U fazi analize trebamo odlučiti što želimo napraviti i dobro razmisliti koji nam je cilj jer je ova faza temelj za sve ostale faze.

Fokus oblikovatelja nastave u ovoj fazi je na ciljanu publiku, tj. za koga se radi nastavni sadržaj. Također nastavni program mora odgovarati nivou znanja i vještina onima za koje se razvija nastavni sadržaj.

U ovoj fazi je potrebno razlučiti što učenici već znaju i što moraju znati nakon završetka procesa učenja i poučavanja. U ovoj fazi je potrebno odgovoriti na sljedeća pitanja [Forest, 2014]  
:

- Za koga radimo nastavni sadržaj
  - Predznanje učenika (potrebno je utvrditi njihovo predznanje i odlučiti što moraju znati nakon završetka nastavnog procesa) ?
- Koji je cilj procesa učenja i poučavanja (što učenici moraju naučiti na kraju nastavnog procesa)?
  - Koje vještine su potrebne za usvajanje nastavnih materijala?
- Odrediti nastavni sadržaj (nastavni sadržaj treba biti on-line)
  - Prema kojem sadržaju ćemo oblikovati nastavne sadržaje
- Na koji način ćemo prikazati nastavni sadržaj (prezentacije, video materijali, multimedija, kvizovi...)
  - Postojanje vježbi, domaćih radova...
  - Zadaci za vježbanje
- Odrediti načine poučavanja (više stilova učenja)
  - Odrediti nastavne strategije (na koji način će se organizirati učenje)
- Odrediti razine i načine interakcije
  - Načini komunikacije (chat, forumi...)
- Korištenje slika iz stvarnog života (razbijaju monotonost predavanja i proču na razmišljanje)
- Odrediti načine ocjenjivanja (testovi, kvizovi, križaljke, rasprave na forumu)
  - Testovi (na kraju lekcije, unutar lekcije ili kombinirano)
  - Broj testova

- Koji su ograničavajući faktori krajnjem cilju? (tehnička podrška, vrijeme, ljudski resursi, tehničke vještine, financije...)

### 3.2.2. Oblikovanje

Ova faza određuje sve ciljeve i alate koji će se koristiti za procjenu izvedbe, raznih testova, tijelo nastavnog sadržaja, planiranje i resurse. Ovdje se pridržavamo strogih uputa i tim sistematskim pristupom osiguravamo da sve spada unutar racionalnih i planiranih strategija kojima je za cilj doći do ciljane publike. Faza zahtijeva izrazitu upornost i temeljitost da bi se postigao konačni cilj. U ovoj fazi osnovno je bazirati se na ciljeve učenja, sadržaj, analizu predmeta, vježbe, planiranje nastave, instrumente za procjenu i odabir medija. U ovoj fazi se pristupa izradi nacрта te se razvija struktura nastavnog sadržaja. Važnost ove faze je u tome što u njoj primjenjujemo sve ono što smo u fazi analize postavili kao kriterije.

Kriteriji faze oblikovanja su [Forest, 2014]:

- Postavljanje vremenskog ograničenja potrebnog za razvoj nastavnog sadržaja
  - Rok za svaku aktivnost. Koliko će se vremena zahtijevati za obavljanje različitih zadataka?
- Navesti cilj nastave
- Određivanje dostupnih materijala potrebnih za završetak projekta
- Razine i vrste aktivnosti koje će biti uključene tijekom procesa (interaktivna, suradnička ili po pojedincu)
- Načini poučavanja (bihevioristički, konstruktivistički...)
  - Da li su lekcije poredane linearno u smislu težine (od jednostavnijih do složenijih)?
  - Nastavni sadržaj „razbiti“ na manje, logične dijelove
- Navesti i opisati način ocjenjivanja
- Koji će se sve alati koristiti?



- Različiti mentalni procesi potrebni sudionicima kako bi se zadovoljile ciljevi projekta. Koje su propisane kognitivne sposobnosti za studente za postizanje ciljeva učenja nastavnog sadržaja?
- Znanje i vještine razvijene nakon svakog zadatka. Da li imate način određivanja tih vrijednosti kod učenika?
- Plan (nacrt) nastavnog sadržaja opisati i nacrtati na papiru.
- Ako je nastavni sadržaj baziran na web-sučelju, koje će se korisničko sučelje koristiti?
- Koji mehanizam povratnih informacija od strane učenika će se koristiti?

### 3.2.3. Razvoj

Faza razvoja je dio gdje oblikovatelji nastave izrađuju i sastavljaju sadržaj prema nacrtu koji je razvijen u fazi oblikovanja. U ovoj fazi, nastavni sadržaj je napisan i izrađena grafika.

U ovoj fazi potrebno se držati sljedećih smjernica [Forest, 2014]:

- Izrada početne stranice
  - Napisati kratki opis nastavnog sadržaja
  - Napisati načine i pravila komuniciranja
  - Napisati nazive programa koji će biti potrebni za rad
- Izraditi nastavno okruženje
  - Definirati prostor za chat, prostori za vođenje rasprava između učenika i između učenika i nastavnika (forumi)
  - Izrada kvizova, križaljki, dodatnih zadataka, wiki, zadaća i testova
  - U svim sadržajima korištenje multimedije
  - Sav nastavni sadržaj treba biti konzistentan
- Pregled
  - Sav nastavni sadržaj treba biti u skladu s nastavnim planom i programom
  - Provjera da poveznice (engl. *link*) ispravno vode na vanjske stranice

### 3.2.4. Implementacija

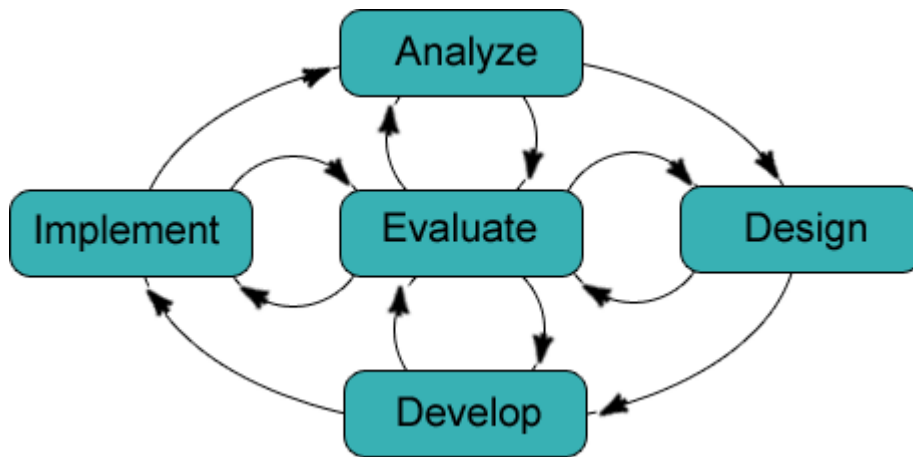
U ovoj fazi počinje planirani nastavni proces. Ova faza nije samo prolazak kroz nastavni sadržaj da se utvrdi da li je sve ispravno, već u ovoj fazi moramo odlučiti da li je to dovoljno, da li je negdje nedostaje objašnjenja važnih pojmova. Na taj način ćemo možda utvrditi da neke dijelove moramo preoblikovati, nadopuniti ili pojednostavniti, a sve u cilju boljeg usvajanja gradiva. U ovoj fazi je također moguće napraviti i pilot istraživanje.

Kriteriji ove faze su [Forest, 2014]:

- Provjeravanje oblikovanog nastavnog sadržaja
  - Ukoliko su postavljeni vremenski uvjeti, provjeriti njihovu ispravnost
  - Provjeriti postavke zadaća i kvizova (ukoliko su postavljeni određeni uvjeti)
- Upoznavanje učenika sa sustavom
  - Učenici čitaju uvod o nastavnom sadržaju i pregledavaju početnu stranicu
  - Nastavnik upućuje učenike što se od njih očekuje
  - Od učenika se traži povratna informacija

### 3.2.5. Vrednovanje

Nakon faze implementacije dolazi faza vrednovanja. U toj fazi se zapravo određuje razlika ostvarenih rezultata i postavljenih ciljeva u fazi analize. Dakle vrednujemo, rezultate učenja ali i sami sustav. Tijekom prethodnih faza se uvijek radi vrednovanje prethodne odrađene faze tako da će sada centar vrednovanja biti rezultat učenja. Iako ne smijemo zanemariti i identifikaciju mogućih grešaka u sustavu.



Slika 3.8 Vrednovanje se obavlja u svim fazama modela [Clark, 1995]

## 4. Oblikovanje prilagodljivog modela courseware-a

U današnje vrijeme se sve više primjenjuje računalna i informacijska tehnologija u procesu učenja i poučavanja. Time je ostvarena težnja za izradu posebnih programskih podrški koji su namijenjeni za učenje i poučavanje potpomognuto računalom a upravo takva programska podrška se zove „courseware“.

„Courseware je međunarodno prihvaćeni termin za nastavni sadržaj oblikovan za izvođenje na računalu.“ [Stankov, 2007]

Svaki nastavni sadržaj se oblikuje prema didaktičkim načelima a kod oblikovanja nastavnog sadržaja na courseware-u moramo paziti na aspekt da se nastavni sadržaj ne nalazi samo na računalu u posebnu oblikovanoj programskoj podrški, courseware, nego da se tu odvija proces učenja i poučavanja.

Courseware se sastoji od više razina, a te razine se: nastavna cjelina, nastavna tema, nastavna jedinica i elementi testiranja i vrednovanja znanja učenika. Upravo ta višerazinska podjela nastavnog sadržaja unutar courseware-a je u skladu s didaktičkim načelima. Nastavna cjelina je zapravo najveća razina, te ona se sastoji od više nastavnih tema, dok se nastavna tema sastoji od više nastavnih jedinica. Odabrani elementi vrednovanja usvojenog znanja učenika se može nalazi u bilo kojoj razini courseware-a. Karakteristike koje svaki courseware ima su: interoperabilnost (engl. *interoperability*), ponovna upotrebljivost (engl. *reusability*), trajnost (engl. *durability*) i dostupnost (engl. *accessibility*).

### 4.1. Područno znanje „Programiranje 1“

Područno znanje je specifično znanje koje se odnosi na određeno usko područje, domenu, za razliku od općeg znanja rješavanja problema. [P. Jackson, 1999]

Područno znanje je često vrlo kompleksno područje, te je potrebno definirati podskup područnog znanja koji će reprezentirati odabrano područno znanje. Tim podskupom obuhvaćamo sve koncepte i relacije koji su relevantni za odabrano područno znanje [Grubišić, 2012].

Za ovaj rad je odabrano područno znanje koje se odnosi na kolegij „Programiranje 1“. Na samom početku bilo je potrebno prikupiti svu dostupnu literaturu koja se nalazila na stranicama

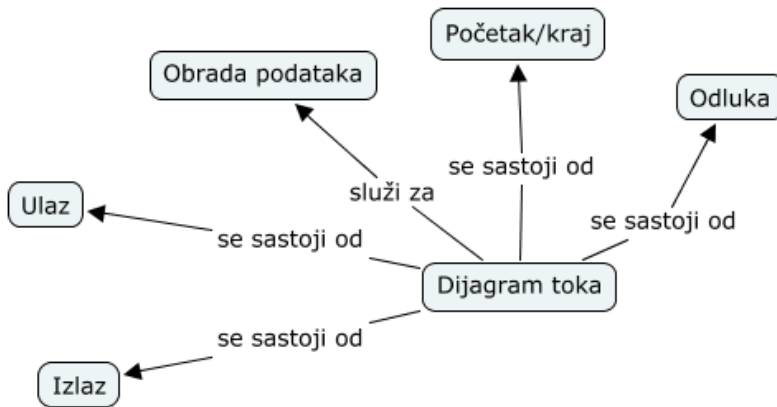
kolegija. Na taj način se osiguralo da se nastavni sadržaj u ovom radu u potpunosti poklapa s nastavnim sadržajem koji su studenti obradili kroz navedeni kolegij. Naime, to je bilo od izuzetne važnosti jer upravo ta grupa studenata će biti ispitna skupina, skupina koja će vrednovati programsku podršku ovoga rada, te će se na taj način dobiti ispravna povratna informacija metodom vrednovanja koja će se opisati u nekom od sljedećih poglavlja.

Cilj kolegija „Programiranje 1“ je identičan cilju izrade programske podrške ovoga rada, tj. oblikovanog nastavnog sadržaja u sustavu Moodle, a to je steći temeljna znanja iz područja programiranja. Zadani cilj se postiže učenjem i poučavanjem temeljnih pojmova o programiranju, osnova razvoja programske podrške, temeljnih algoritamskih struktura i metoda programiranja.

#### **4.1.1. Mapa konceptata „Programiranje 1“**

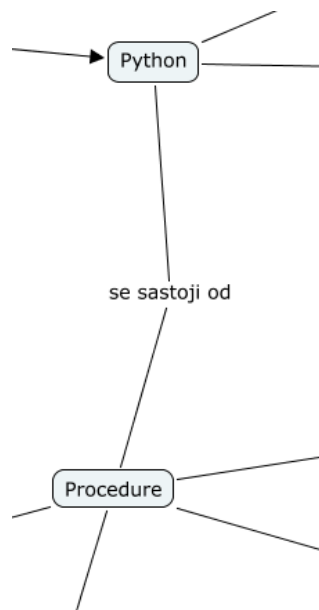
Mapa konceptata (engl. *concept map*) služi za grafičko prezentiranje odabranog područnog znanja na temelju klastera računalom oblikovanog nastavnog sadržaja i matrice korelacija konceptata koji omogućuju sistematično učenje [Chang, Chang, Heh, & Liu, 2008].

Za odabrano područno znanje smo prvo izdvojili sve važne pojmove koji će u mapi konceptata zapravo biti koncepti. Za izradu mape konceptata koristili smo sustav Cmap Tools COE. Program se može besplatno preuzeti sa sljedeće stranice: <http://cmap.ihmc.us/>. U sustavu CMAP Tools COE koncepti su povezani određenim relacijama koji predstavljaju kako su dva koncepta zavisna jedna o drugom. U mapi konceptata potrebno je naznačiti smjer povezanosti konceptata i tada govorimo o pojmovima: *podkoncept* i *nadkoncept*. Bilo koja dva koncepta su povezana relacijom koja sadrži tekst koji opisuje vezu kojom su ta dva koncepta međusobno povezana i usmjerenje koja prikazuje odnos među konceptima. Na slici ispod se prikazuje dio mape konceptata izrađenu u svrhu ovog rada na kojoj je prikazan koncept „Dijagram toka“ s pripadajućim konceptima povezanim određenim relacijama. Na slici je vidljivo da je koncept „Dijagram toka“ nadkoncept konceptima s kojima je povezan. Odnos nadkoncept - podkoncept je vidljiv usmjerenjem relacija između njih.



Slika 4.1 Nadkoncept i i njegovi podkoncepti

Svaka mapa koncepata treba biti pregledna i to je vrlo jednostavno kod jednostavnijih mapa koncepata, tj. s manjim brojem koncepata, dok u slučaju velikog broja koncepata vrlo je složeno organizirati koncepte na način da je lako shvatljiva veza i povezanost manjih dijelova mape koncepata. Problem kod CMAP Tools-a je to da usmjerenje relacija nije uvijek vidljivo.



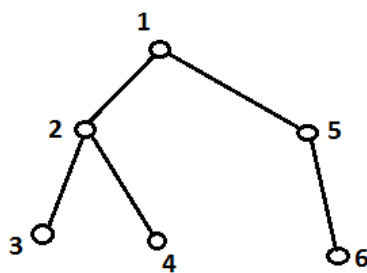
Slika 4.2 Relacija na kojoj nije vidljivo usmjerenje

Za područno znanje odabrano je 112 koncepata koji predstavljaju reprezentativno znanje odabranog područnog znanja. Odabrane koncepte je trebalo sistematično ubaciti u Cmap Tools COE te ih povezati određenim relacijama. U Cmap Tools COE je moguće odabirom koncepta

dodavati slike, poveznice u svrhu dodatnog objašnjenja, te također postoji opcija da se za odabrani koncept prikažu samo njegova, djeca, roditelji, korijen, podstablo i nadstablo.

Stablo T je konačni skup vrhova tako da vrijedi [Veljan, 2009]:

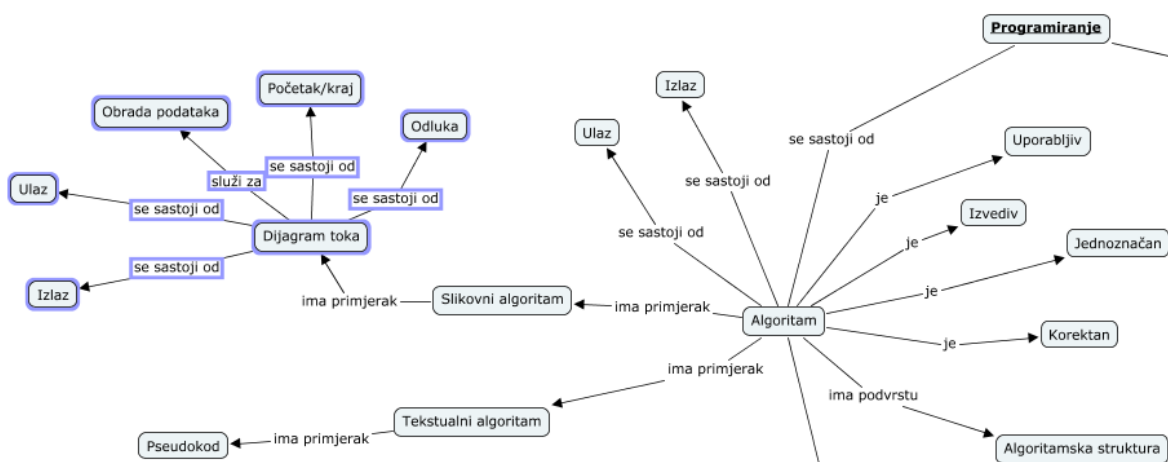
- Posebno odabrani vrh zove se **korijen** od T
- Ostali vrhovi (osim korijena) su participirani u  $m \geq 0$  disjuntnih nepraznih skupova  $T_1, \dots, T_m$  koji su stabla i zovu se **podstabla** korijena



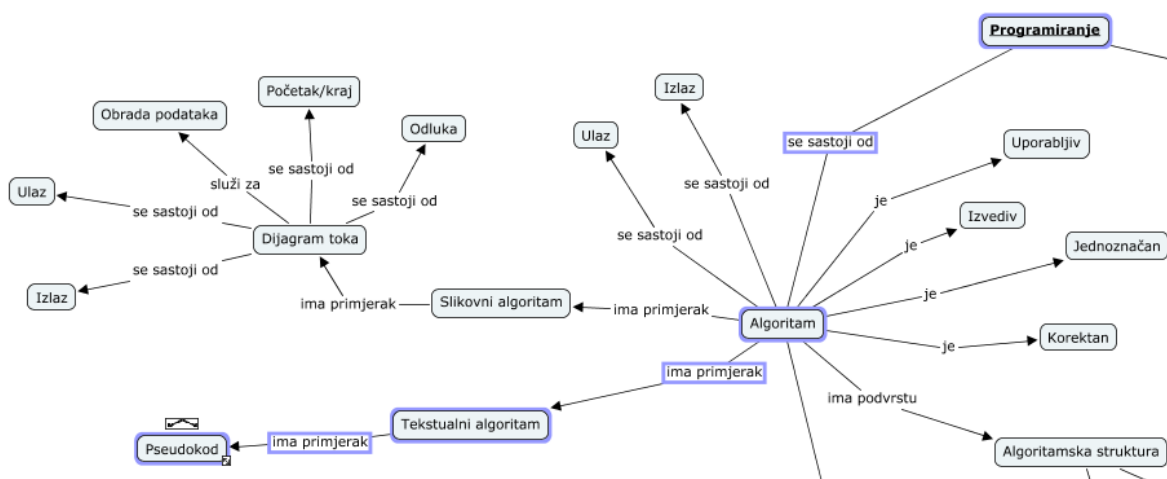
Slika 4.3 Stablo s vrhovima

Na slici iznad, vrh 1 je **korijen**, vrhovi 2 i 5 su **djeca** (nasljednik vrha stabla) od 1, vrhovi 3 i 4 su djeca od vrha 2, a vrh 6 od vrha 5. Svaki vrh bez djeca zove se **list**. Vrhovi 2, 3 i 4 čine jedno **podstablo** od stabla T.

Na slici ispod se vidi dio od mape koncepata „Programiranje 1“ izrađenoj u svrhu ovog rada. Na slikama će biti prikazani neki od iznad pojašnjanih pojmova koristeći funkcionalnosti Cmap Tools-a COE.



Slika 4.4 Djeca koncepta "Dijagram toka"



Slika 4.5 Put od lista do korijena

Slikama su prikazane neke od mogućnosti sustava, te kako na jednostavan način dobiti grafički prikaz nekih od složenih matematičkih funkcija koje su se koristile u ovom radu.

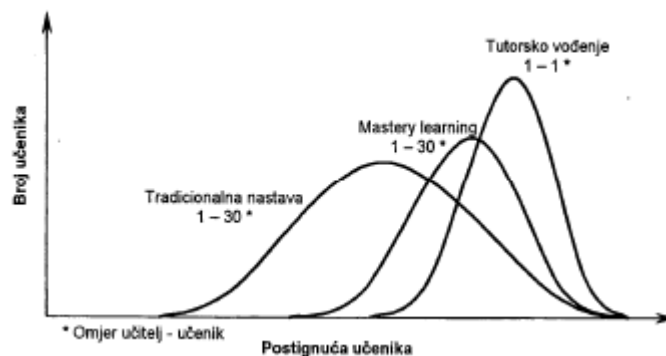
Čitav popis koncepata nalazi se u dodatku.

## 4.2. Prilagodljivi courseware u inteligentnim tutorskim sustavima

U procesu učenja i poučavanja je došlo do saznanja da postoje tri načina učenja novih nastavnih sadržaja, a to su: tradicionalno učenje (engl. *Conventional learning*), učenje s provjeravanjem (engl. *Mastery learning*) i tutorsko učenje (engl. *Tutoring learning*). Osnovna razlika tutorskog poučavanja s preostala dva je: na svakog učenika ide jedan nastavnik (ili jedan tutor ide na dva ili tri učenika), dok kod preostala dva ide jedan nastavnik do 30 učenika [Bloom, 1984].

Benjamin Bloom je napravio istraživanje u kojem je rezultat sljedeći: ustanovljeno je da je prosječan učenik u tutorskoj grupi za oko dvije standardne devijacije bolji od prosječnog učenika u kontrolnom razredu (prosječni učenik u tutorskom sustavu je bolji od 98% učenika iz kontrolnog razreda) [Bloom, 1984]. Problem kod tutorskog način poučavanja je što nije ekonomski održiv, takav način poučavanja u obrazovnim ustanovama je vrlo skup u usporedbi s preostalim načinima poučavanja.





Slika 4.6 2-sigma problem

Razvojem tehnologije došlo je do razvoja sustava e-učenja u smjeru da sustavi e-učenja aktivno sudjeluju u učenju u poučavanju, tj. da sustavi e-učenja budu u mogućnosti poučavati učenike. Upravo takvi sustavi se nazivaju inteligentni tutorski sustavi (ITS) i takvi sustavi su se temeljili na sljedećim definicijama:

- Poučavanje po modelu jedan-na-jedan (engl. *one-to-one tutoring*) je specijalizirana vrsta nastave koja se odvija u intenzivnoj interakciji učenika ili male grupe učenika (onaj/oni koga/kojih se poučava) i učitelja - tutora (onaj tko poučava). [Stankov, 2010]
- Poučavanje po modelu jedan-na-jedan koje se odvija pod okriljem ljudskog tutora je dokazano uspješno i smatra se da je to najučinkovitiji način odvijanja nastavnog procesa [Cohen i drugi, 1982; Bloom, 1984].

Dakle inteligentni tutorski sustavi su područje umjetne inteligencije koje je povezano sa aktivnostima za „oponašanje“ tutorskih vrednota „živog“ tutora. Inteligentan tutorski sustav mora raspolagati sa znanjem, te osim toga biti sposoban precizno dijagnosticirati znanje učenika. Svaki inteligentni tutorski sustav se sastoji od:

- Modul stručnjaka (područno znanje)
- Modul učenika (znanje učenika)
- Modul učitelja (tutorsko znanje)
- Modul komunikacije (okruženje nastavnog procesa i sučelja učenika)

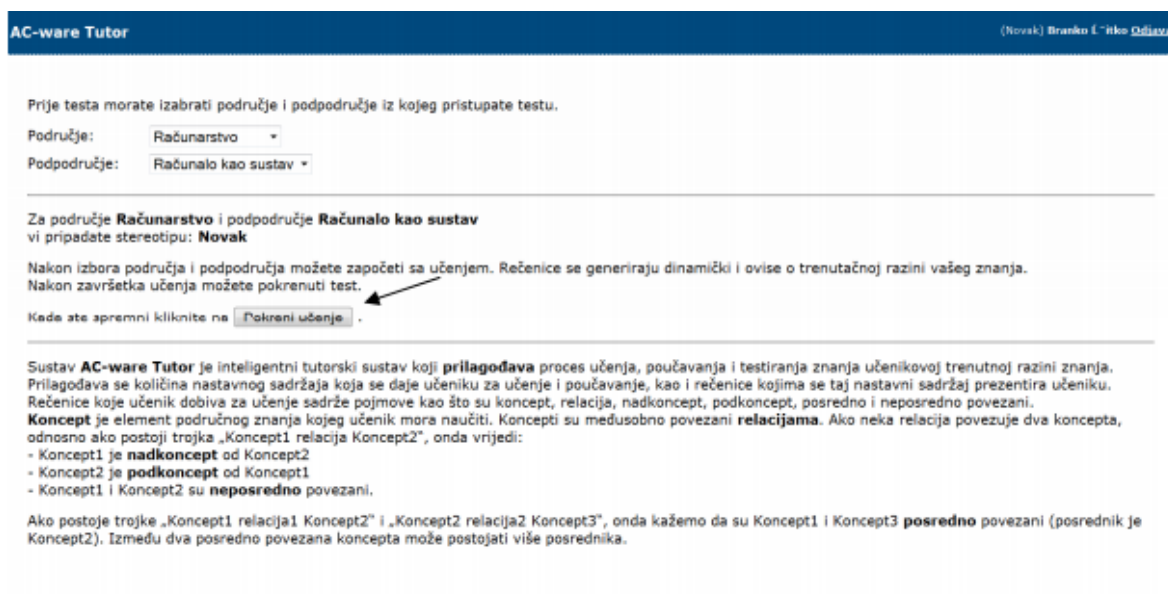
Inteligentnim tutorskim sustavima se zapravo nastojao riješiti 2-sigma problem. Cilj je vrlo zahtjevan i ambiciozan i sama pomisao da računalo zamijeni živog tutora zvuči nestvarno, no krajnji cilj nije da računalo zamijeni živog tutora (što možda neće nikad biti u potpunosti moguće),

već je krajnji cilj da se tutorski način poučavanja omogući svim učenicima (bez obzira na financijski status). Zbog toga tom cilju treba težiti, jer iako je možda neostvariv, vrlo je plemenit.

#### 4.2.1. Prilagodljivi courseware AC – ware Tutoru

AC – ware Tutor [Grubišić, 2012] je sustav namijenjen za učenje, poučavanje i testiranje učenika. Naziv sustava je skraćenica od engleskog izraza: *Adaptive Courseware Tutor model*. To je zapravo model prilagodljivog i dinamičkog generiranja nastavnog sadržaja koji je računalom oblikovan u inteligentnim tutorskim sustavima. Glavna karakteristika ovog sustava je što se prilagođava učeniku prema njegovu trenutnom znanju.

Da bi se sustav mogao prilagođavati njegovu trenutnom znanju, potrebno je da takav sustav sadrži model učenika, model koji predstavlja učenikovo znanje. Nakon što se korisnik prijavi u sustav, odabire područje i podpodručje koje želi učiti.



Slika 4.7 Početna stranica sustava AC – ware Tutor

Unutar sustava područno znanje je prezentirano u obliku koncepata i njihovih relacija a ocjenjivanje učenika je u skladu s Bloom-ovom taksonomijom znanja:

- Ocjena nedovoljan (1) predstavlja nedostatak znanja
- Ocjena dovoljan (2) predstavlja znanje na razini reprodukcije
- Ocjena dobar (3) predstavlja znanje na razini razumijevanja
- Ocjena vrlo dobar (4) predstavlja znanje na razini primjene

- Ocjena odličan (5) predstavlja znanje na razni analize, sinteze i procjene

U sustavu se prema znanju učenici grupiraju u 5 stereotipova, a stereotipovi su sljedeći:

- Novak
- Početnik
- Osrednji
- Napredni
- Stručnjak

Sljedeća tablica prikazuje na koji način će se učeniku prezentirati znanje obzirom na stereotip kojem pripada.

Stereotip prema znanju	Razina znanja
<b>Novak</b>	Reprodukcija
<b>Početnik</b>	Razumijevanje
<b>Osrednji</b>	Razumijevanje
<b>Napredni</b>	Primjena
<b>Stručnjak</b>	Analiza, sinteza i razumijevanje

Tablica 4.1 Prezentiranje znanja prema stereotipovima [Grubišić, 2012]

### 4.3. Oblikovanje courseware-a

Programska podrška ovog rada temelji se na principu rada AC-ware Tutora. U skladu s tim, u radu se učenikovo znanje također dijeli na stereotipove spomenute u prethodnom poglavlju. Dakle, postoji 5 stereotipova: *Novak*, *Početnik*, *Osrednji*, *Napredni* i *Stručnjak*. U ovom poglavlju ćemo objasniti postupak podjele na stereotipove koji je razvijen u AC-Ware Tutoru i korišten u ovom radu. Dakle, u našem slučaju koristimo stereotipove prema znanju. Ako znamo kojem stereotipu učenik pripada, tada znamo i njegovu trenutnu razinu znanja. Ova podjela na stereotipove je zapravo podjela prema Bloom-ovoj taksonomiji znanja. Vidimo da na ovakav način

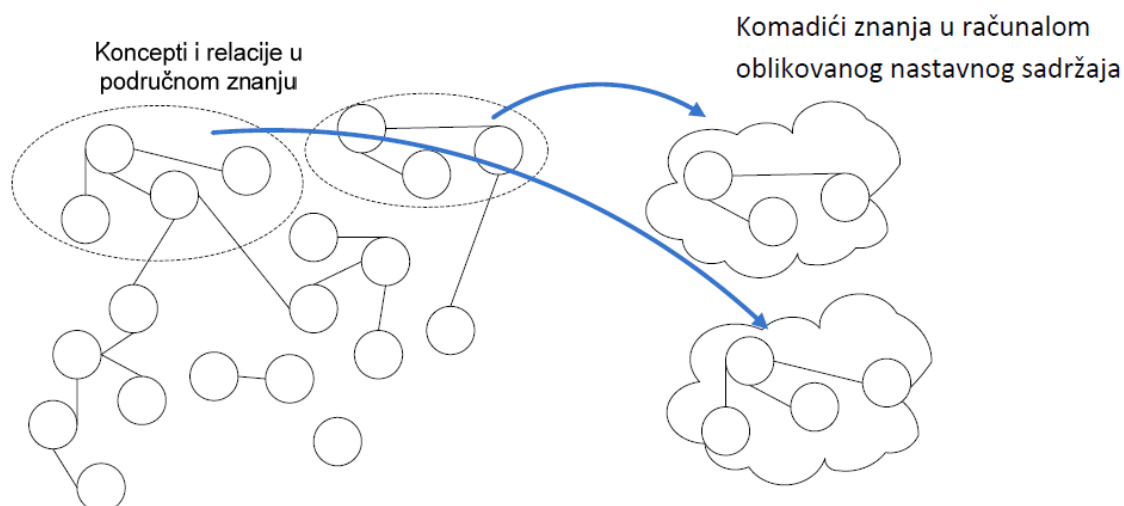
sve učenike koji ne posjeduju nikakvo znanje (osim možda nešto znanja na razini reprodukcije) smatramo *novacima*, one koji iskazuju znanje na razini reprodukcije (i možda malo znanja na razini razumijevanja) smatramo *početnicima*, one koji pokazuju reprodukciju i razumijevanje (i možda malo znanja na razini primjene) smatramo *osrednjima*, one koji reproduciraju, razumiju i primjenjuju znanje (i možda analiziraju, sintetiziraju i procjenjuju) smatramo *naprednima*, dok one koji to znanje mogu reproducirati, razumjeti, primijeniti, te analizirati, sintetizirati i procijeniti smatramo *stručnjacima*.

Ocjena	Stereotip prema znanju	Reprodukcija	Razumijevanje	Primjena	Analiza, sinteza i vrednovanje
1	Novak	Možda	Nema	Nema	Nema
2	Početak	Većina	Možda	Nema	Nema
3	Osrednji	Većina	Većina	Možda	Nema
4	Napredni	Većina	Većina	Većina	Možda
5	Stručnjak	Većina	Većina	većina	većina

Tablica 4.2 Odnos stereotipova učenika i razina znanja [Grubišić, 2012]

### 4.3.1. Elementi za učenje

Općenito, kada učitelj prvi put pristupa nastavnom sadržaju, najčešće ga podijeli na manje dijelove. Na taj način se lakše ostvaruju nastavni ciljevi i pravovremeno dobivanje povratne informacije trenutne razine učenikova znanja.



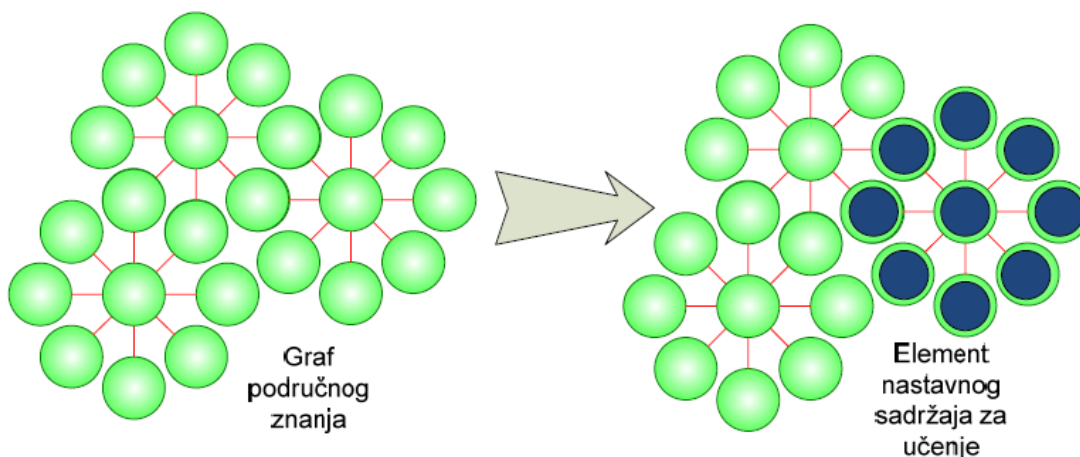
Slika 4.8 Odnos područnog znanja i računalom oblikovanog nastavnog sadržaja

Elementi računalom oblikovanog nastavnog sadržaja su u skladu s hrvatskom pedagoškom praksom, a ti elementi su: *nastavne cjeline* (NC), *nastavne teme* (NT) i *nastavne jedinice* (NJ). Navedeni elementi su elementi za učenje, a testovi su elementi za provjeravanje znanja.

„Objekt znanja pruža okvir za organiziranje područnog znanja na takav način da se može koristiti u različitim kontekstima i da se može prikazati na različite načine“ [Merril, 1998].

U našem slučaju, objekti znanja su elementi računalom oblikovanog nastavnog sadržaja za učenje. Uvjeti koji predstavljaju minimum za generiranje računalom oblikovanog nastavnog sadržaja za učenje su [Grubišić, 2012]:

- Nastavna cjelina odgovara cjelini područnog znanja » postojanje barem jednog puta duljine 4 i više od početka elementa
- Nastavna tema » postojanje barem jednog puta duljine 2 od početka elementa i barem dva podkoncepta ili postojanje barem jednog puta duljine 3 od početka elementa
- Nastavna jedinica » postojanje barem jednog puta duljine 1 od početka elementa i barem dva podkoncepta ili postojanje barem jednog puta duljine 2 od početka elementa.



Slika 4.9 element računalom oblikovanog nastavnog sadržaja za učenje

#### 4.3.1.1 Generiranje elemenata računalom oblikovanog nastavnog sadržaja

Elementi računalom oblikovanog nastavnog sadržaja su generirani algoritmima korištenim u sustavu AC-Ware Tutor [Grubišić, 2012]. U ovom potpoglavlju ćemo detaljnije objasniti njihov postupak. Područno znanje „Programiranje 1“ je veliko, a predstavljeno je mapom koncepata koja zbog veličine nije prikladna za prikaz objašnjenje navedenih algoritama, koristit ćemo se njenim dijelovima, dok je cijela mapa koncepata prikazana u dodatku.

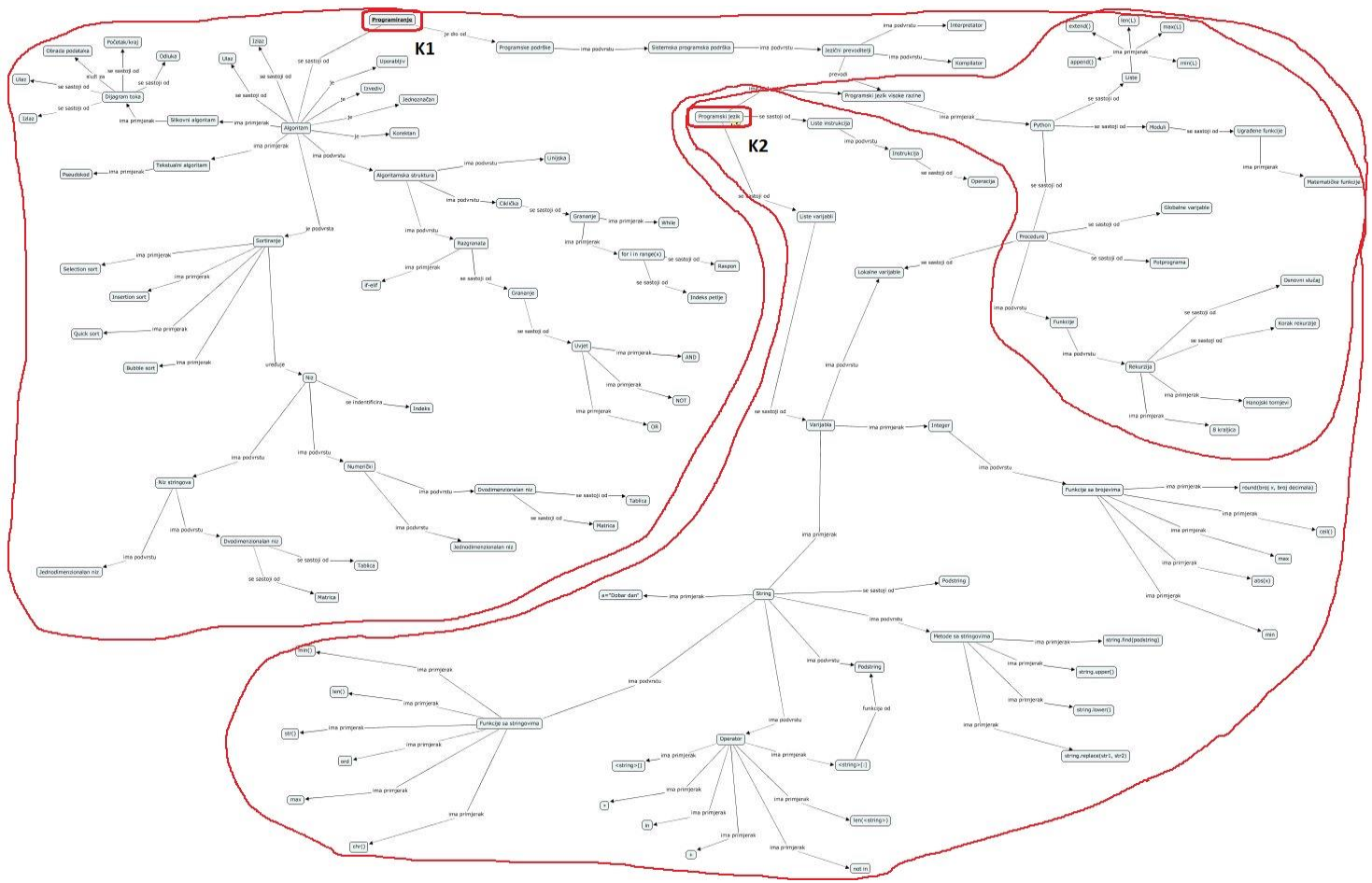
##### *Faza 1 – generiranje nastavnih cjelina*

Za svaki korijen  $K_x$  u grafu područnog znanja koji je centralni vrh neke cjeline  $C_i$  ( $i$  je sortiran prema rangu cjeline) ponavljamo sljedeći algoritam za određivanje elemenata učenja [Grubišić, 2012]:

Algoritam 5. Generiranje nastavnih cjelina	
1.	Ako u toj cjelini postoji put duljine veće od 3, onda definiramo $NC_{K_x}$ , tj. nastavnu cjelinu s početkom u $K_x$ . Rang $R_{NC_{K_x}} = (i, 0, 1)$ .
2.	Inače, ako u toj cjelini postoji put duljine veće od 1, onda definiramo $NT_{K_x}$ , tj. nastavnu temu s početkom u $K_x$ . $R_{NT_{K_x}} = (i, 0, 2)$ .
3.	Inače definiramo $NJ_{K_x}$ , tj. nastavnu jedinicu s početkom u $K_x$ . $R_{NJ_{K_x}} = (i, 0, 3)$ .
4.	Na kraju se u elemente računalom oblikovanog nastavnog sadržaja dodaju atributi i njihove vrijednosti (jer smo ih izbacili prilikom definiranja cjelina područnog znanja).

Tablica 4.3 Algoritam za generiranje nastavnih cjelina [Grubišić, 2012]

Na slici ispod je prikazana čitava mapa koncepata, te njena podjela na 2 nastavne cjeline.



Slika 4.10 Nastavne cjeline u mapi koncepata „Programiranje 1“

Razlika Cmap Tools-a COE u odnosu na Cmap Tools je u tome što korišteni sustav ima mogućnost lakog otkrivanja korijena. Naime, to je važno, jer prvi korak za određivanje NC je upravo detekcija korijena, a naučili smo da je korijen zapravo koncept koji ima karakteristike da nema nadkoncepta, tj. sve relacije iz njega „izlaze“. Nakon što smo to otkrili pristupili smo određivanju NC prema algoritmu 5. Oznake na slici  $K_1$  i  $K_2$  predstavljaju dva korijena. U našem slučaju oba korijena ispunjavaju uvjete za NC.

- Cjelina  $C_1$  s centralnim vrhom „**Programiranje**“ sadrži put duljine veće od 3, pa je ona nastavna cjelina, tj.  $NC_{Programiranje} = C_1, R_{NC_{Programiranje}} = (1,0,1)$ .
- Cjelina  $C_2$  s centralnim vrhom „**Programski jezik**“ sadrži put duljine veće od 3, pa je ona nastavna cjelina, tj.  $NC_{Programski jezik} = C_1, R_{NC_{Programski jezik}} = (2,0,1)$ .

## Faza 2 – generiranje nastavnih tema

Za svaku nastavnu cjelinu  $NT_{K_x}$  ponavljamo sljedeći algoritam za određivanje njenih nastavnih tema. Nastavnu temu čine podstabla, tj. koncept i njegovi podkoncepti i njihovi podkoncepti.

Kod određivanja nastavnih tema treba imati u obzir kada koncept ima samo jedan podkoncept, tada se uzimaju i podkoncepti od tog podkoncepta [Grubišić, 2012].

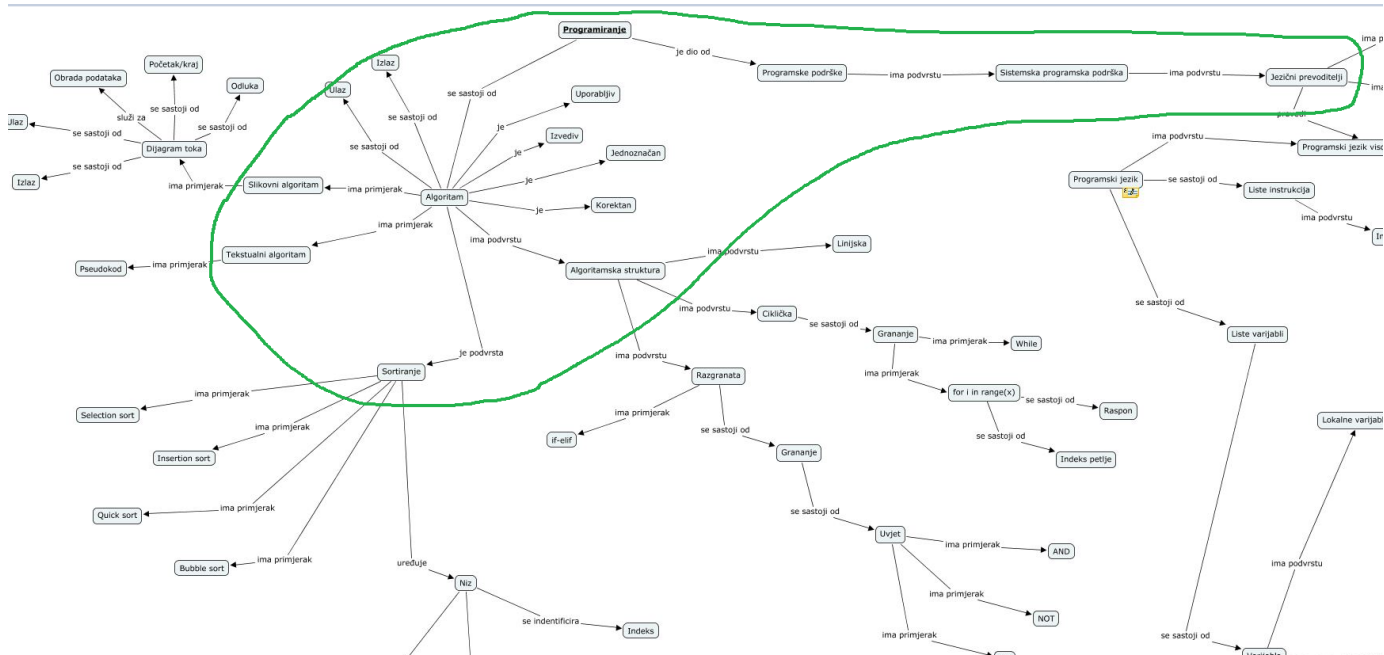
Algoritam 6. Generiranje nastavnih tema
<ol style="list-style-type: none"><li>1. Odaberemo početak <math>K_x</math> nastavne cjeline <math>NC_{K_x}</math>.</li><li>2. Skup <math>V'</math> vrhova nastavne teme <math>NT_{K_x}</math> čini početak <math>K_x</math> i njegovi podkoncepti (ali samo ako pripadaju odabranoj nastavnoj cjelini).</li><li>3. Ako podkoncept od <math>K_x</math> ima samo jedan podkoncept, onda u <math>V'</math> ulazi i taj podkoncept i njegovi podkoncepti (dakle koncepti koji su za 3 udaljeni od <math>K_x</math>, ali samo ako pripadaju odabranoj nastavnoj cjelini ili ako su listovi).</li><li>4. Ako podkoncept od <math>K_x</math> ima više podkonceptata, onda u <math>V'</math> ulaze samo ti podkoncepti (dakle koncepti koji su za 2 udaljeni od <math>K_x</math>, ali samo ako pripadaju odabranoj nastavnoj cjelini).</li><li>5. Ako nema nijedan koncept u točkama 3 i 4, onda ne možemo konstruirati nastavnu temu. Ako <math>K_x</math> ima podkoncepte, onda konstruiramo nastavnu jedinicu od tog koncepta i njegovih podkonceptata te od podpodkonceptata (ako postoje).</li><li>6. Za svaki list nastavne teme <math>NT_{K_x}</math> provjeravamo da li možemo konstruirati nastavnu temu koja počinje u njemu (mora postojati put u cjelini <math>NC_{K_x}</math> koji počinje u tom listu, a ima duljinu barem 2, a taj list mora imati više od jednog podkoncepta) na način opisan u koracima 2, 3 i 4.</li><li>7. Korak 6 ponavljamo za svaku konstruiranu nastavnu temu.</li><li>8. Na kraju se u elemente računalom oblikovanog nastavnog sadržaja dodaju atributi i njihove vrijednosti (jer smo ih izbacili prilikom definiranja cjelina područnog znanja).</li></ol>

Tablica 4.4 Algoritam generiranja nastavnih tema [Grubišić, 2012]

Sada ćemo objasniti postupak Algoritam 6 na primjeru mape konceptata „Programiranje 1“:

- Definirajmo nastavne teme unutar nastavne cjeline  $NC_{Programiranje}$
- Krećemo od centralnog vrha „**Programiranje**“





- Promotrimo sada listove nastavne teme  $NT_{Programiranje}$  – *ulaz, izlaz, uporabljiv, izvediv, jednoznačan, korektan, algoritamska struktura, sortiranje, tekstualni algoritam, slikovni algoritam, programska podrška, sistemska programska podrška, jezični prevoditelji*.
- Za koncepte: *ulaz, izlaz, uporabljiv, izvediv, jednoznačan, korektan* se ne može konstruirati niti nastavna tema niti nastavna jedinica.
- Za koncepte: *slikovni algoritam, tekstualni algoritam* možemo konstruirati nastavne jedinice, ali ne možemo konstruirati nastavne teme:

$$- V_{N|tekstualni\ algoritam} = \{tekstualni\ algoritam, pseudokod\}$$

$$- V_{N|slikovni\ algoritam} = \{slikovni\ algoritam, dijagram\ toka, izlaz, ulaz, obrada\ podataka, po\etak/kraj, odluka\}$$

- Za koncepte: *sortiranje, algoritamska struktura, jezični prevoditelji* možemo konstruirati nastavne teme jer postoji put u cjelini  $NC_{Programiranje}$  koji počinje u tom listu, a ima duljinu barem 2.

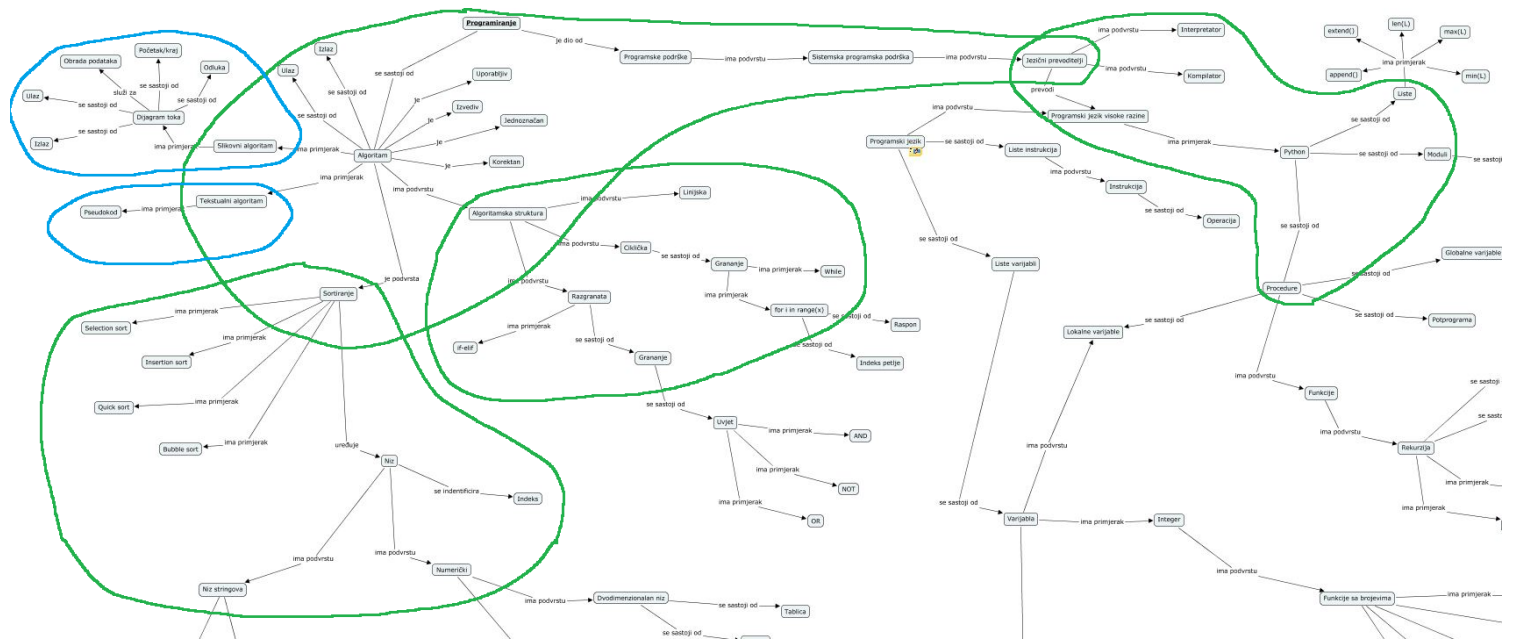
$$- V_{NT_{sortiranje}} = \{sortiranje, selection\ sort, insertion\ sort, quick\ sort, bubble\ sort, niz, indeks, numerički, niz\ stringova\}$$

$$- V_{NT_{Algoritamska\ struktura}} = \{algoritamska\ struktura, linijska, ciklička, razgranata, if-elif, grananje, while, for\ i\ in\ range\}$$

$$- V_{NT_{jezični\ prevoditelji}} = \{jezični\ prevoditelji, interpretator, kompilator, programski\ jezik\ visoke\ razine, python, liste, moduli, procedure\}$$

- za koncept „*Slikovni algoritam*“ se ne može generirati nastavna tema jer koncept sadrži jedan podkoncept, a u tom slučaju ukupna duljina bi trebala biti 3 dok je ovdje slučaj da je ukupna duljina 2. Stoga podstablo s korijenom „*Slikovni algoritam*“ je nastavna jedinica.

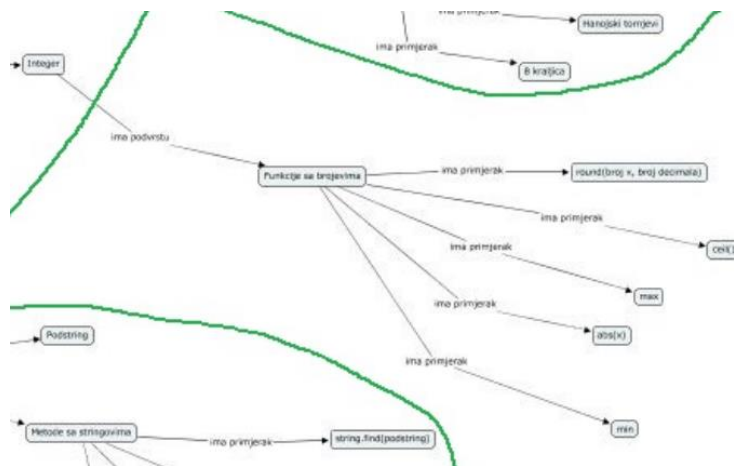
- za koncept „*Tekstualni algoritam*“ se ne može generirati nastavna tema jer je ukupni put duljine 1, te stoga podstablo s korijenom „*Tekstualni algoritam*“ je nastavna jedinica.



- Na isti način se generiraju preostale nastavne teme

Na sljedećoj slici su prikazane nastavne teme u mapi konceptata „Programiranje 1“. Nakon određivanja nastavnih tema, neke nastavne teme su pripadale objema nastavnim cjelinama. Tom problemu smo pristupili na sljedeći način. Ukoliko je određena nastavna tema imala više koncepata u određenoj nastavnoj cjelini, tada nastavna tema pripada onoj cjelini u kojoj sadrži više koncepata.





Slika 4.12 podstablo koje ne odgovara uvjetima algoritma za nastavnu temu

Prema algoritmu 6. promatramo koncept „Integer“ jer je on posljednji koncept promatranog podstabla koji pripada određenoj nastavnoj temi. Udaljenost od koncepta „Integer“ do najudaljenijeg lista promatranog podstabla je 2: udaljenost od koncepta „Integer“ do koncepta „Funkcija s brojevima“ je 1, dok je udaljenost od koncepta „Funkcija s brojevima“ do bilo kojeg lista promatranog podstabla također 1. Dakle, ukupna udaljenost je 2, a jedan od uvjeta da podstablo bude nastavna tema je da udaljenost bude barem 3 ukoliko korijen sadrži samo jedan podkoncept, te zbog toga promatrano podstablo ne može biti nastavna tema, već nastavna jedinica.

### Faza 3 – generiranje nastavnih jedinica

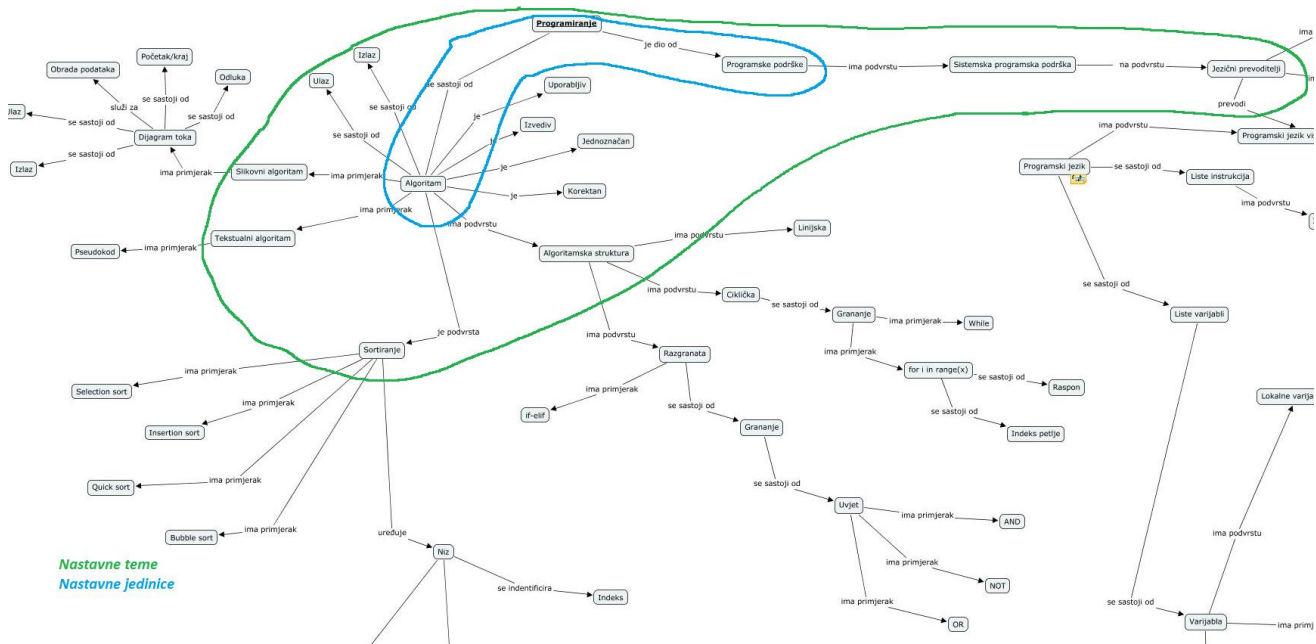
Nakon generiranja nastavnih tema slijedi generiranje nastavnih jedinica. Nastavne jedinice predstavljaju najmanji skup koncepata a njihovo generiranje se vrši na sljedeći način [Grubišić, 2012]:

- Za svaku nastavnu temu  $NT_{K_x}$  ponavljamo sljedeći algoritam za određivanje njenih nastavnih jedinica (nastavnu jedinicu čine koncept i njegovi podkoncepti; ako koncept ima samo jedan podkoncept, onda se uzimaju i podkoncepti od podkoncepta):

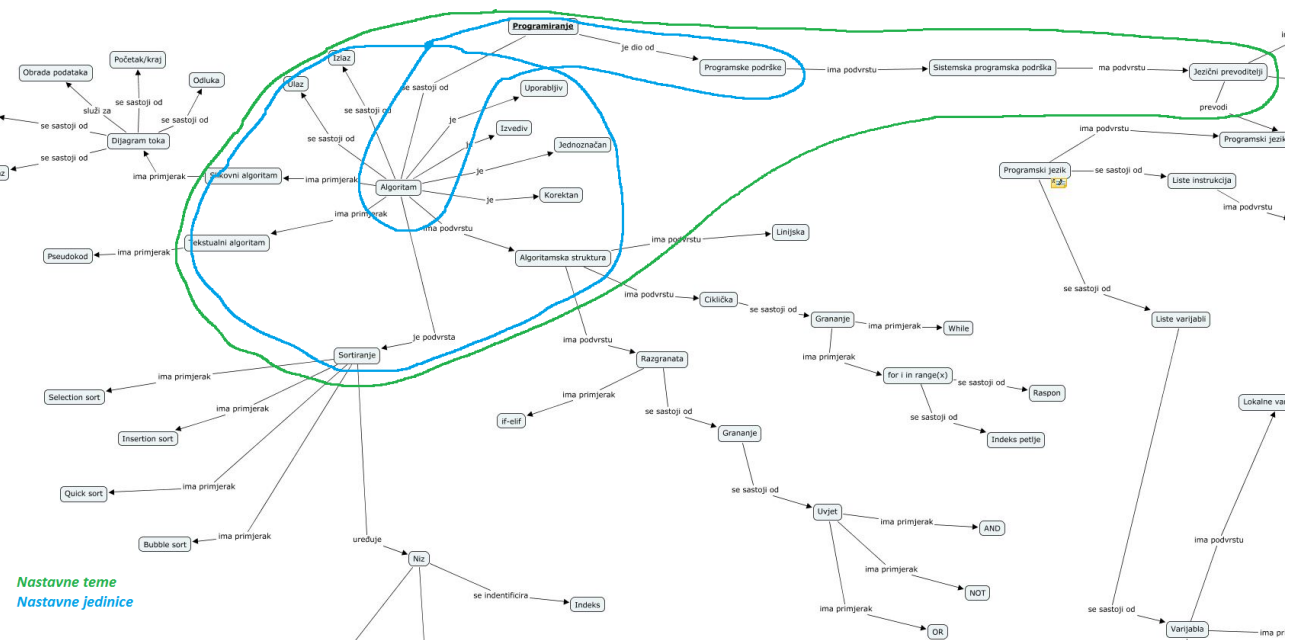
Algoritam 7. Generiranje nastavnih jedinica
1. Odaberemo početak $K_x$ nastavne cjeline $NT_{K_x}$ .
2. Skup $V'$ vrhova nastavne teme $NJ_{K_x}$ čini početak $K_x$ i njegovi podkoncepti (ali samo ako pripadaju odabranoj nastavnoj cjelini).
3. Ako $K_x$ ima samo jedan podkoncept, onda u $V'$ ulazi i njegovi podkoncepti (dakle koncepti koji su za 2 udaljeni od $K_x$ , ali samo ako pripadaju odabranoj nastavnoj temi ili ako su listovi).
4. Za svaki list nastavne teme $NJ_{K_x}$ provjeravamo da li možemo konstruirati nastavnu jedinicu koja počinje u njemu (mora postojati put u odabranoj temi koji počinje u tom listu, a ima duljinu barem 1) na način opisan u koracima 2 i 3.
5. Korak 4 ponavljamo za svaku konstruiranu nastavnu jedinicu.
6. Na kraju se u elemente računalom oblikovanog nastavnog sadržaja dodaju atributi i njihove vrijednosti (jer smo ih izbacili prilikom definiranja cjelina područnog znanja).

Sada ćemo ukratko objasniti postupak generiranja nastavnih jedinica na primjeru mape konceptata „Programiranje 1“:

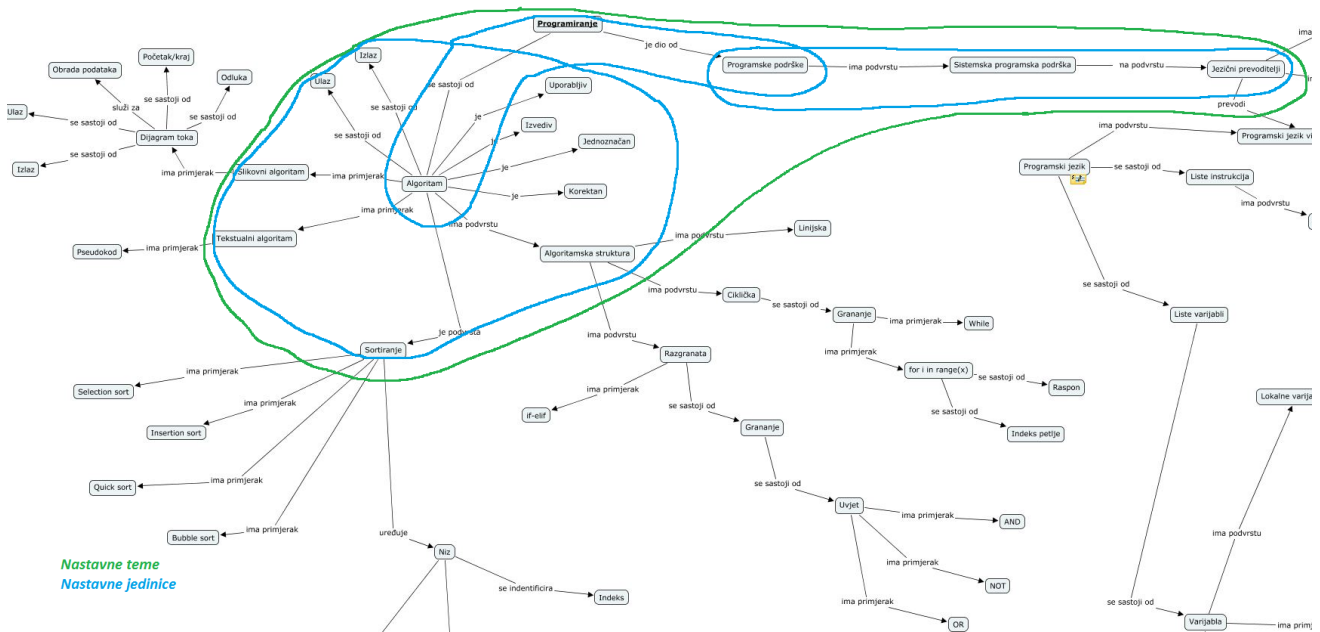
- Krećemo od centralnog vrha, korijena 1. nastavne cjeline, „Programiranje“
- Prolaskom kroz korake Algoritam 7. odabiremo odgovarajuće koncepte nastavne jedinice



- Nakon toga promatramo listove nastavne jedinice  $N_{J_{Programiranje}}$  – **Programirane podrška**, **Algoritam**. Za svih se može konstruirati nastavna jedinica jer postoji put **unutar** teme  $N_{T_{Programiranje}}$  koji počinje u tom listu, a ima **duljinu barem 1**.



- $V_{NJ_{Algoritam}} = \{Izlaz, ulaz, uporabljiv, izvediv, jednoznačan, korektan, algoritamska struktura, sortiranje, tekstualni algoritam, slikovni algoritam\}$
- Unutar nastavne teme  $NT_{Programiranje}$  je moguće generirati još jednu nastavnu jedinicu, i to  $NJ_{Programska podrška}$ .



- $V_{NJ_{Programska podrška}} = \{Sistemska programska podrška, jezični prevoditelji\}$
- Promotrimo sada nastavne listove  $NJ_{Programska podrška}$ . Za nijednog se ne može konstruirati nastavna jedinica jer ne postoji put u temi  $NT_{Programiranje}$  koji počinje u tom listu, a ima duljinu barem 1.
- Isti proces se ponavlja za ostale nastavne jedinice

Algoritmi za generiranje nastavnih cjelina, tema i jedinica (*Algoritam 5, Algoritam 6 i Algoritam 7*) osiguravaju sljedeće [Grubišić, 2012]:

- Presjek svake dvije nastavne teme je najviše jedan koncept (list jedne nastavne teme je korijen druge)
- Presjek svake dvije nastavne jedinice je najviše jedan koncept (list jedne nastavne jedinice je korijen druge)
- Svaka nastavna cjelina se sastoji od barem jedne nastavne teme i jedne nastavne jedinice
- Svaka nastavna tema se sastoji od barem dvije nastavne jedinice
- Jedan koncept može biti korijen različitih vrsta elemenata računalom oblikovanog nastavnog sadržaja u istoj cjelini, ali i u različitim cjelinama
- Jedan koncept može biti korijen iste vrste elemenata računalom oblikovanog nastavnog sadržaja u različitim cjelinama

Algoritmi za generiranje elemenata računalom oblikovanog nastavnog sadržaja se temelje na grani matematike „Teorija grafova“. Prilikom školovanja je upravo taj dio matematike dio obveznog kolegija „Kombinatorna i diskretna matematika“, te je znanje usvojeno na tom kolegiju bilo od velikog značaja za ovaj rad.

#### 4.3.2. Strukture statičkog računalom oblikovanog nastavnog sadržaja za stereotipove

Nakon definiranja nastavnog sadržaja (u nastavku „NS“), te njegove podjele na stereotipove prema znanju, pristupili smo izradi struktura statičkog računalom oblikovanog sadržaja za sve stereotipove. Struktura sadrži točno definiran raspored prikazivanja nastavnog sadržaja i raspored testova zbog vrednovanja znanja. Odabir elemenata računalom oblikovanog nastavnog sadržaja se vrši prema razini elemenata, a razina elemenata nam određuje količinu nastavnog sadržaj kojeg će učenik učiti u sljedećem ciklusu učenja i poučavanja.

Dakle, nastavni sadržaj smo podijelili na nastavne cjeline, nastavne teme i nastavne jedinice. S obzirom da smo definirali pet stereotipova učenika (novak, početnik, osrednji, napredni, stručnjak), a imamo samo tri razine elemenata računalom oblikovanog nastavnog sadržaja (nastavna cjelina, nastavna tema i nastavna jedinica), te će se zbog toga neki stereotipovi učiti i poučavati na istoj razini (Tablica 4.5).

Stereotip prema znanju	Razina elemenata računalom oblikovanog nastavnog sadržaja
<b>Novak</b>	Nastavna jedinica
<b>Početaknik</b>	Nastavna jedinica
<b>Osrednji</b>	Nastavna tema
<b>Napredni</b>	Nastavna tema
<b>Stručnjak</b>	Nastavna cjelina

Tablica 4.5 Stereotip i razine elemenata računalom oblikovanog nastavnog sadržaja

Sada pristupamo odabiru nastavnog sadržaja i njegovog redoslijeda za svaki stereotip. To se obavlja na sljedeći način [Grubišić, 2012]:

1. Prema oznaci cjeline
2. Rastući prema udaljenosti početka elementa od korijena cjeline
3. Prema razini elementa (nastavna cjelina, nastavna tema, nastavna jedinica)
4. Abecedno prema nazivu korijena

Na ovaj način smo definirali uređaj na skupu NS za svaki stereotip, a nakon toga smo pristupili definiranju kada će učenik provjeravati svoje znanje u ovisnosti trenutnoj razini znanja, tj. kojem stereotipu pripada. Za sve stereotipove je određeno da zadnji element računalom oblikovanog nastavnog sadržaja bude test.

Elementi za provjeravanje znanja se dodaju ovisno kojem stereotipu učenik pripada. Dakle, nakon definiranja kojem stereotipu učenik pripada, potrebno je definirati i način prezentiranja računalom oblikovanog nastavnog sadržaja koje je definirano obzirom na stereotip učenika. Sljedeća tablica se razvijena prema Bloom-ovoj taksonomiji te odgovara tablici 4.1 (Poglavlje 4.2.1)

Stereotip prema znanju	Pravila za dodavanje elemenata za provjeravanje znanja
<b>Novak</b>	<ul style="list-style-type: none"> <li>- Kod promjene oznake cjeline</li> <li>- Kod promjena udaljenosti od korijena (osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica)</li> <li>- nakon tri nastavne jedinice u nizu koje nisu test (osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica)</li> <li>- zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>
<b>Početak</b>	<ul style="list-style-type: none"> <li>- Kod promjene oznake cjelina (jedino ako se ispred zadnje nastavne jedinice u cjelini nalaze još barem dvije nastavne jedinice u nizu),</li> <li>- Kod promjena udaljenosti od korijena (osim ako bi testu prethodile samo dvije nastavne jedinice ili ako bi iza testa ostale samo dvije nastavne jedinice)</li> <li>- Zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>
<b>Osrednji</b>	<ul style="list-style-type: none"> <li>- Kod promjene oznake cjelina (jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test)</li> <li>- Kod promjene udaljenosti od korijena (osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostale samo dva elementa koja nisu test),</li> <li>- Nakon tri elementa koja u nizu koje nisu test (osim ako bi testu prethodila samo jedan element koji nije test ili ako bi iza testa ostao samo jedan element koji nije test)</li> <li>- Zadnji element računalom oblikovanog nastavnog sadržaja mora biti test )</li> </ul>
<b>Napredni</b>	<ul style="list-style-type: none"> <li>- Kod promjene oznake cjelina (jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test),</li> <li>- Kod promjena udaljenosti od korijena (osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostale samo dva elementa koja nisu test)</li> <li>- Zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>
<b>Stručnjak</b>	<ul style="list-style-type: none"> <li>- Zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>

Tablica 4.6 Pravila za dodavanje elemenata za provjeravanje znanja [Grubišić, 2012]

Prema prethodnoj tablici smo napravili tablicu u kojoj se nalaze svi elementi računalom oblikovanog nastavnog sadržaja, njihov redni broj, oznaka cjeline, vrsta elementa (NC, NT i NJ), udaljenost od korijena određene cjeline i broj koncepata. Tablica je izrađena s namjerom da olakša kada se pristupi izradi strukture računalom oblikovanog nastavnog sadržaj prema pravilima za



dodavanje elemenata za provjeravanje znanja (Tablica 4.6). Tablica u kojoj se nalaze svi elementi i njihovi atributi se nalazi u prilogu A.

Primijetimo da što je niža razina znanja učenika, to će se njegovo znanje češće provjeravati. Važno je napomenuti da se kod nižih razina znanja, znanje provjerava nad manjim podskupom područnog znanja, dok će se učenicima s većom razinom znanja, provjeravanje vršiti rjeđe nad većim podskupom područnog znanja.

Upravo prema tome su se razvili algoritmi za odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja za svaki stereotip [Grubišić, 2012]:

- Ako učenik pripada stereotipu „**Novak**“, onda će mu se generirati nastavni sadržaj prema sljedećem algoritmu:

Novak – algoritam generiranja nastavnog sadržaja
1. Svaku nastavnu jedinicu dodaj u NS, osim ako je nastavna jedinica s istim korijenom već usvojena na razini 1, 2, 3 ili 4.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka nastavnih jedinica od korijena cjeline, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip novak.
3. Između svake dvije nastavne jedinice u nizu koje imaju različite oznake cjelina stavi test.
4. Između svake dvije nastavne jedinice u nizu koje imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica.
5. Nakon tri nastavne jedinice u nizu koja nisu test dodaj test, osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica.
6. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

Tablica 4.7 Algoritam za generiranje nastavnog sadržaja za stereotip „Novak“ [Grubišić, 2012]

- Ako učenik pripada stereotipu „**Početak**“, onda će mu se generirati nastavni sadržaj prema sljedećem algoritmu:

Početak – algoritam generiranja nastavnog sadržaja
1. Svaku nastavnu jedinicu dodaj u NS, osim ako je nastavna jedinica s istim korijenom već usvojena na razini 1, 2, 3 ili 4.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka nastavnih jedinica od korijena cjeline, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip početnik.
3. Između svake dvije nastavne jedinice u nizu koje imaju različite oznake cjelina stavi test, ali jedino ako se ispred zadnje nastavne jedinice u cjelini nalaze još barem dvije nastavne jedinice u nizu.
4. Između svake dvije nastavne jedinice u nizu koje imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodile samo dvije nastavne jedinice ili ako bi iza testa ostale samo dvije nastavne jedinice.
5. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

Tablica 4.8 Algoritam za generiranje nastavnog sadržaja za stereotip „Početnik“ [Grubišić, 2012]

- Ako učenik pripada stereotipu „**Osrednji**“, onda će mu se generirati nastavni sadržaj prema sljedećem algoritmu:

Osrednji – algoritam generiranja nastavnog sadržaja
1. <i>Svaku nastavnu temu dodaj u NS, osim ako je nastavna tema s istim korijenom već usvojena na razini 1, 2, 3 ili 4. Ako nema više tema, a postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu jedinicu koja ih sadrži</i>
2. <i>Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema vrsti elementa, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip osrednji.</i>
3. <i>Između svaka dva elementa u nizu koje imaju različite oznake cjelina stavi test, ali jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test.</i>
4. <i>Između svake dva elementa u nizu koja imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostala samo dva elementa koja nisu test.</i>
5. <i>Nakon tri elementa u nizu koja nisu test dodaj test, osim ako bi testu prethodila samo jedna nastavna tema ili jedinica ili ako bi iza testa ostala samo jedna nastavna tema ili jedinica.</i>
6. <i>Na kraj računalom oblikovanog nastavnog sadržaja stavi test.</i>

Tablica 4.9 Algoritam za generiranje nastavnog sadržaja za stereotip „Osrednji“ [Grubišić, 2012]

- Ako učenik pripada stereotipu „**Napredni**“, onda će mu se generirati nastavni sadržaj prema sljedećem algoritmu:

Napredni – algoritam generiranja nastavnog sadržaja
1. <i>Svaku nastavnu temu dodaj u NS, osim ako je nastavna tema s istim korijenom već usvojena na razini 1, 2, 3 ili 4. Ako nema više tema, a postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu jedinicu koja ih sadrži.</i>
2. <i>Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema vrsti elementa, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip napredni.</i>
3. <i>Između svaka dva elementa u nizu koje imaju različite oznake cjelina stavi test, ali jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test.</i>
4. <i>Između svake dva elementa u nizu koja imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostala samo dva elementa koja nisu test.</i>
5. <i>Na kraj računalom oblikovanog nastavnog sadržaja stavi test.</i>

Tablica 4.10 Algoritam za generiranje nastavnog sadržaja za stereotip „Napredni“ [Grubišić, 2012]

- Ako učenik pripada stereotipu „**Stručnjak**“, onda će mu se generirati nastavni sadržaj prema sljedećem algoritmu:

<b>Stručnjak – algoritam generiranja nastavnog sadržaja</b>
1. Svaku nastavnu cjelinu dodaj u NS, osim ako je nastavna cjelina s istim korijenom već usvojena na razini 4. Ako u cjelini nema više nastavnih cjelina, a u cjelini još postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu temu koja ih sadrži. Ako u cjelini nema više tema, a u cjelini još postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu jedinicu koja ih sadrži.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema vrsti elementa, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip stručnjak.
3. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

Tablica 4.11 Algoritam za generiranje nastavnog sadržaja za stereotip „Stručnjak“ [Grubišić, 2012]

Prema prethodnim tablicama koje opisuju algoritme za generiranje nastavnog sadržaja napravljene su sljedeće tablice za područno znanje „Programiranje 1“ koje se odabralo za ovaj rad.

### Stereotip „Novak“

Stereotip *novak* na područnom znanju „Programiranje 1“ sadrži 31 nastavnu jedinicu (5 se izbacuju jer se ponavljaju):

- 22 nastavne jedinice iz cjeline „Programiranje“ (oznaka cjeline = 1)
- 9 nastavnih jedinica iz cjeline „Programski jezik“ (oznaka cjeline = 2)

Zbog navedenih elemenata računalom oblikovanog nastavnog sadržaja dodaje se 10 testova prema algoritmu za generiranja nastavnog sadržaja za stereotip „Novak“ (tablica 4.7).

### Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip „Novak“

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Programiranje	1	0	Nastavna jedinica	3
Programska podrška	1	1	Nastavna jedinica	3
Algoritam	1	1	Nastavna jedinica	10
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Slikovni algoritam	1	2	Nastavna jedinica	7
Tekstualni algoritam	1	2	Nastavna jedinica	2
Sortiranje	1	2	Nastavna jedinica	6

Algoritamska struktura	1	2	Nastavna jedinica	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Jezični prevoditelji	1	3	Nastavna jedinica	4
Niz	1	3	Nastavna jedinica	4
Razgranata	1	3	Nastavna jedinica	3
Ciklička	1	3	Nastavna jedinica	2
Dijagram toka	1	3	Nastavna jedinica	6
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Programski jezik visoke razine	1	4	Nastavna jedinica	5
Grananje	1	4	Nastavna jedinica	5
Niz stringova	1	4	Nastavna jedinica	3
Numerički	1	4	Nastavna jedinica	3
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Dvodimenzionalan niz	1	5	Nastavna jedinica	3
For i in range(x)	1	5	Nastavna jedinica	3
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Lista	1	6	Nastavna jedinica	6
Moduli	1	6	Nastavna jedinica	3
Procedura	1	6	Nastavna jedinica	5
Funkcije	1	7	Nastavna jedinica	6
<b>TEST – zbog završetka cjeline</b>				
Programski jezik	2	0	Nastavna jedinica	4
Programski jezik visoke razine	2	1	Nastavna jedinica	5
Lista instrukcije	2	1	Nastavna jedinica	3
Lista varijabli	2	1	Nastavna jedinica	5

**TEST – zbog promjene udaljenosti od korijena**

Integer	2	3	Nastavna jedinica	7
---------	---	---	-------------------	---

**TEST – zbog promjene udaljenosti od korijena**

String	2	4	Nastavna jedinica	7
--------	---	---	-------------------	---

**TEST – zbog promjene udaljenosti od korijena**

Funkcije sa stringovima	2	5	Nastavna jedinica	7
-------------------------	---	---	-------------------	---

Operator	2	5	Nastavna jedinica	8
----------	---	---	-------------------	---

Metode sa stringovima	2	5	Nastavna jedinica	5
-----------------------	---	---	-------------------	---

**TEST – zbog završetka računalom oblikovanog nastavnog sadržaja****Stereotip „Početnik“**

Stereotip *početnik* na područnom znanju “Programiranje 1” sadrži 31 nastavnu jedinicu (5 se izbacuju jer se ponavljaju):

- 22 nastavne jedinice iz cjeline „Programiranje“ (oznaka cjeline = 1)
- 9 nastavnih jedinica iz cjeline „Programski jezik“ (oznaka cjeline = 2)

Zbog navedenih elemenata računalom oblikovanog nastavnog sadržaja dodaje se 9 testova prema algoritmu za generiranja nastavnog sadržaja za stereotip „Početnik“ (*tablica 4.8*).

**Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip *Početnik***

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Programiranje	1	0	Nastavna jedinica	3
Programska podrška	1	1	Nastavna jedinica	3
Algoritam	1	1	Nastavna jedinica	10

**TEST – zbog promjene udaljenosti od korijena**

Slikovni algoritam	1	2	Nastavna jedinica	7
--------------------	---	---	-------------------	---

Tekstualni algoritam	1	2	Nastavna jedinica	2
----------------------	---	---	-------------------	---

Sortiranje	1	2	Nastavna jedinica	6
------------	---	---	-------------------	---

Algoritamska struktura	1	2	Nastavna jedinica	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Jezični prevoditelji	1	3	Nastavna jedinica	4
Niz	1	3	Nastavna jedinica	4
Razgranata	1	3	Nastavna jedinica	3
Ciklička	1	3	Nastavna jedinica	4
Dijagram toka	1	3	Nastavna jedinica	6
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Programski jezik visoke razine	1	4	Nastavna jedinica	5
Grananje	1	4	Nastavna jedinica	5
Niz stringova	1	4	Nastavna jedinica	3
Numerički	1	4	Nastavna jedinica	3
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Dvodimenzionalan niz	1	5	Nastavna jedinica	3
For i in range(x)	1	5	Nastavna jedinica	3
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Lista	1	6	Nastavna jedinica	6
Modul	1	6	Nastavna jedinica	3
Procedura	1	6	Nastavna jedinica	5
Funkcije	1	7	Nastavna jedinica	6
<b>TEST – zbog završetka cjeline</b>				
Programski jezik	2	0	Nastavna jedinica	4
Programski jezik visoke razine	2	1	Nastavna jedinica	5
Lista instrukcije	2	1	Nastavna jedinica	3
Lista varijabli	2	1	Nastavna jedinica	5
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Integer	2	3	Nastavna jedinica	7
String	2	4	Nastavna jedinica	7
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Funkcije sa stringovima	2	5	Nastavna jedinica	7
Operator	2	5	Nastavna jedinica	8

Metode sa stringovima	2	5	Nastavna jedinica	5
<b>TEST – zbog završetka računalom oblikovanog nastavnog sadržaja</b>				

## Stereotip „Osrednji“

Stereotip *osrednji* na područnom znanju “Programiranje 1” sadrži 10 nastavnih tema i 8 nastavnih jedinica:

- 7 nastavnih tema iz cjeline „Programiranje“ (oznaka cjeline = 1)
- 4 nastavne jedinice iz cjeline „Programiranje“ (oznaka cjeline = 1)
- 3 nastavne teme iz cjeline „Programski jezik“ (oznaka cjeline = 2)
- 4 nastavne jedinice iz cjeline „Programski jezik“ (oznaka cjeline = 2)

Zbog navedenih elemenata računalom oblikovanog nastavnog sadržaja dodaje se 6 testova prema algoritmu za generiranja nastavnog sadržaja za stereotip „Osrednji“ (*tablica 4.9*).

### Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip „Osrednji“

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Programiranje	1	0	Nastavna tema	15
Sortiranje	1	2	Nastavna tema	9
Algoritamska struktura	1	2	Nastavna tema	9
<b>TEST – zbog niza od tri nastavna elementa</b>				
Slikovni algoritam	1	2	Nastavna jedinica	7
Tekstualni algoritam	1	2	Nastavna jedinica	2
Jezični prevoditelji	1	3	Nastavna tema	8
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Niz stringova	1	4	Nastavna tema	5
Numerički	1	4	Nastavna tema	5
Grananje	1	5	Nastavna jedinica	5
<b>TEST – zbog niza od tri nastavna elementa</b>				
For i in range(x)	1	5	Nastavna jedinica	3
Procedure	1	6	Nastavna tema	10
<b>TEST – zbog završetka cjeline</b>				

Programski jezik	2	0	Nastavna tema	14
String	2	3	Nastavna tema	24
Liste	2	3	Nastavna jedinica	6
<b>TEST – zbog niza od tri nastavna elementa</b>				
Moduli	2	3	Nastavna jedinica	3
Integer	2	3	Nastavna jedinica	7
<b>TEST – zbog završetka računalom oblikovanog nastavnog sadržaja</b>				

## Stereotip „Napredni“

Stereotip *napredni* na područnom znanju “Programiranje 1” sadrži 9 nastavnih tema i 7 nastavnih jedinica:

- 7 nastavnih tema iz cjeline „Programiranje“ (oznaka cjeline = 1)
- 4 nastavne jedinice iz cjeline „Programiranje“ (oznaka cjeline = 1)
- 2 nastavne teme iz cjeline „Programski jezik“ (oznaka cjeline = 2)
- 3 nastavne jedinice iz cjeline „Programski jezik“ (oznaka cjeline = 2)

Zbog navedenih elemenata računalom oblikovanog nastavnog sadržaja dodaje se 6 testova prema algoritmu za generiranja nastavnog sadržaja za stereotip „napredni“ (*tablica 4.10*).

## Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip „Napredni“

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Programiranje	1	0	Nastavna tema	15
Sortiranje	1	2	Nastavna tema	9
Algoritamska struktura	1	2	Nastavna tema	9
Slikovni algoritam	1	2	Nastavna jedinica	7
Tekstualni algoritam	1	2	Nastavna jedinica	2
Jezični prevoditelji	1	3	Nastavna tema	8
Niz stringova	1	4	Nastavna tema	5
Numerički	1	4	Nastavna tema	5
Grananje	1	5	Nastavna jedinica	5
For i in range(x)	1	5	Nastavna jedinica	3
Procedure	1	6	Nastavna tema	10



**TEST – zbog završetka cjeline**

Programski jezik	2	0	Nastavna tema	14
String	2	3	Nastavna tema	24
Liste	2	3	Nastavna jedinica	6
Moduli	2	3	Nastavna jedinica	3
Integer	2	3	Nastavna jedinica	7

**TEST – zbog završetka računalom oblikovanog nastavnog sadržaja****Stereotip „Stručnjak“**

Stereotip *stručnjak* na područnom znanju “Programiranje 1” sadrži 2 nastavne cjeline:

- 1 nastavnu cjelinu „Programiranje“ (oznaka cjeline = 1)
- 1 nastavnu cjelinu „Programski jezik“ (oznaka cjeline = 2)

Zbog navedenih elemenata računalom oblikovanog nastavnog sadržaja dodaje se 1 test prema algoritmu za generiranja nastavnog sadržaj za stereotip „stručnjak“ (*tablica 4.11*).

**Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip „Stručnjak“**

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Programiranje	1	0	Nastavna cjelina	75
Programski jezik	2	0	Nastavna cjelina	31

**TEST – zbog završetka računalom oblikovanog nastavnog sadržaja****4.3.3. Klasteri**

Klasteri (engl. *Clusters*) su posebna grupa stranica u Lesson Moodle-u. Klaster se tvori tako da smjestimo posebnu navigacijsku „cluster“ stranicu na početku grupe i „end cluster“ stranicu nakon zadnje stranice u toj grupi pitanja.

Grupa klastera (niz stranica, grupa pitanja) sadrži stranice sa pitanjima na koje trebamo usmjeriti našu pažnju. Najbolje je prvo dodati „branch table“ u lekciju, zatim stranice sa pitanjima i na kraju razmisliti gdje bi najbolje bilo smjestiti navigacijske stranice definirane klasterima.

*Primjer:*

Q1, **C1**, Q2, Q3, Q4, Q5, Q6, Q7, **EC1**, Q8

Početak klastera

Kraj klastera

Iz primjera se vidi da svaki klaster mora završiti sa „End of cluster“ stranicom. Klaster omogućava da slučajnim odabirom, učenik nikad ne dobije isto pitanje na stranici s klasterima. Klasteri mogu sadržavati podklaster koristeći „Branch Table“ i „End of Branch“.

*Primjer:*

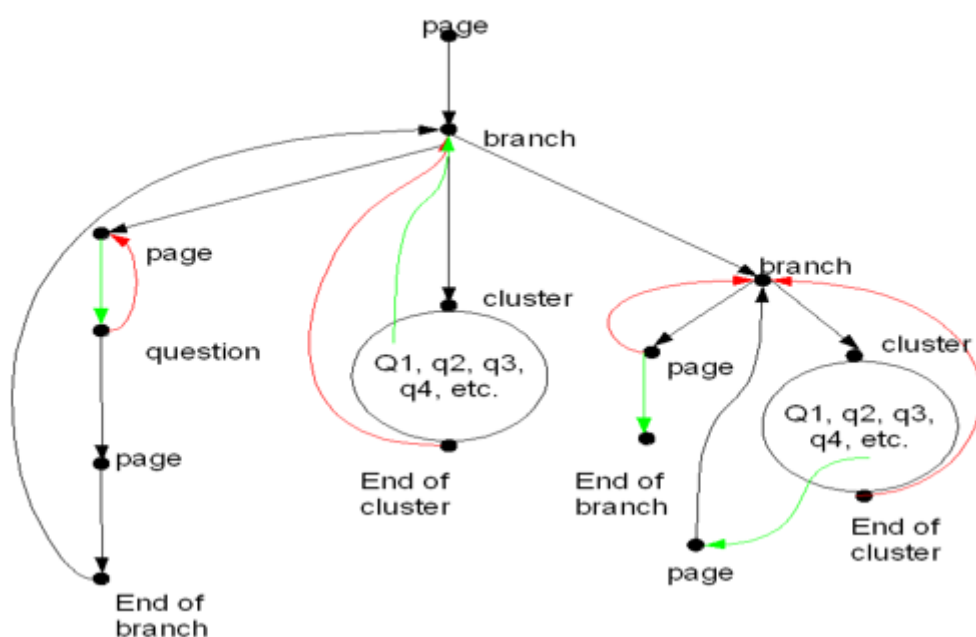
Q1, **C1**, Q2, Q3, **B1**, Q4, Q5, Q6, **EB1**, Q7, **EC1**, Q8

Kada učenik pristupi lekciji, samo jedna slučajno odabrana stranica s pitanjem unutar podklastera će biti prikazana učeniku. Klaster koristimo kada želimo da svaki od učenika, ma koliko ih bilo, pristupi istoj lekciji dobije različit redoslijed pitanja.

*Primjer:*

Q1, Q2, **C1**, Q3, Q4, Q5, **EC1**, Q6, Q7, Q8

Kada učenik dođe do klastera *C1*, nastavnik na tom klasteru može postaviti skok na neko pitanje unutar klastera koje učenik još nije vidio. Nakon što pređe pitanja *Q3*, *Q4* i *Q5*, može ga se vratiti ponovo na klaster *C1*, ali tamo će ga dočekati opet novo pitanje. Kada je učenik vidio sva pitanja u klasteru, oni će biti poslani *EC1*. Učenik neće nikada vidjeti *EC1*, samo će biti preusmjeren skokom iz *EC1*.



Slika 4.13 Grafički prikaz klastera

Upravo zbog mogućnosti da se nakon pogrešnog odgovora učenika vrati na stranicu s točnim odgovorom odabrali smo da se stereotip „*Novak*“ izradi upravo u obliku klastera. Zapravo smo testirali klastere zbog određene vrste dinamičnosti i prilagođavanja nastavnog sadržaja učeniku, jer je upravo to jedina funkcionalnost *Moodle*-a u smjeru prilagođavanja nastavnog sadržaja učeniku. Na početku pristupili smo izradi svake nastavne jedinice kao zasebne lekcije. To smo odlučili jer ćemo za ostale stereotipove te iste lekcije moći iskoristiti prema već izrađenoj strukturi statičkog računalom oblikovanog nastavnog sadržaja (struktura statičkog računalom oblikovanog nastavnog sadržaja je objašnjena u prethodnom potpoglavlju). Preostale stereotipove smo razvili i implementirali na principu da svaki element računalom oblikovanog nastavnog sadržaja bude predstavljen lekcijom izrađenoj u *Moodle-u*. Te lekcije će sadržavati određene kriterije za napredovanje koje ćemo objasniti u jednom od sljedećih poglavlja.

Sljedeći korak nam je bio izrada klastera u *Moodle-u* prateći strukturu statičkog računalom oblikovanog nastavnog sadržaja za stereotip „*Novak*“. Zamišljeno je da svaki pojedini klaster predstavlja test koji se javlja ili zbog promjene udaljenosti od korijena ili zbog niza od tri nastavne jedinice. To je značilo da moramo napraviti 10 klastera jer upravo toliko stereotip „*Novak*“ sadrži testova.

Za svaku nastavnu jedinicu potrebno je izraditi lekciju, dok bi test izradili pomoću klastera. Razgovorom smo došli do zaključka da bi za svaki koncept u mapi koncepata trebalo izraditi minimalno 3 pitanja. Nakon netočnog odgovora učenika se treba vrati na stranicu s točnim odgovorom u cilju da bolje prouči nastavni sadržaj potrebne nastavne jedinice koje prethode testu izraditi u jednu lekciju u Moodle-u. U ovom slučaju, nastavne jedinice: *Programiranje, programska podrška i algoritam* predstavljaju jednu lekciju. Nakon toga slijedi test koji smo izradili pomoću klastera. Taj postupak smo ponavljali za sve nastavne elemente stereotipa „Novak“ prateći tablicu statičkog računalom oblikovanog nastavnog sadržaja za stereotip „Novak“.

Kako je svaki test ograničen na određen nastavni sadržaj koji se njime provjerava, tako su i klasteri bile ograničeni određenim brojem koncepata. U slučaju netočnog odgovora, učenik nema pravo prelaziti na sljedeći nastavni sadržaj, tj. na podkoncepte danih koncepata čiji se sadržaj provjerava testom. U postavkama klastera, potrebno je izbaciti izbornik sa sadržajem koji se nalazi na lijevoj strani jer njegovim izbacivanjem, učenik je „bespomoćan“ dok prelazi lekciju i jedino što može je prelaziti stranice onako kako smo mi zamislili. To je jako važno jer je cilj da učenik pređe sve stranice unutar lekcije, a ne da pojedine preskače, kako to učenici često znaju raditi.

### **Dodavanje klastera**

Pokazat ćemo kako smo dodali u lekciju „*Programiranje, programska podrška i algoritam*“ klaster. Općenito, ako unutar lekcije želimo dodati klaster idemo na *Uredi* → *Dodaj grupu pitanja*

ili ako želimo dodati kraj grupe pitanja: *Uredi* → *Dodaj kraj grupe pitanja*.

Algoritam - nastavak	Tablica grananja	Nasumičnim redom odabrano pitanje unutar grane	🔍 ⚙️ 🔍 ✖️	Pitanje
<b>Grupa pitanja</b>	Grupa pitanja	Pitanje 1	🔍 ⚙️ 🔍 ✖️	<b>Dodaj novu stranicu...</b>
Pitanje 1	Točno/Netočno	Neprikazano pitanje iz grupe pitanja Programiranje	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 2	Višestruki odabir	Neprikazano pitanje iz grupe pitanja Neprikazano pitanje iz grupe pitanja Programiranje Neprikazano pitanje iz grupe pitanja	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 3	Višestruki odabir	Neprikazano pitanje iz grupe pitanja Faze programiranja Neprikazano pitanje iz grupe pitanja Neprikazano pitanje iz grupe pitanja	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 4	Točno/Netočno	Neprikazano pitanje iz grupe pitanja Programska podrška	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 5	Višestruki odabir	Neprikazano pitanje iz grupe pitanja Programska podrška Neprikazano pitanje iz grupe pitanja	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 6	Točno/Netočno	Neprikazano pitanje iz grupe pitanja Algoritam	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 7	Višestruki odabir	Neprikazano pitanje iz grupe pitanja Neprikazano pitanje iz grupe pitanja Neprikazano pitanje iz grupe pitanja Algoritam	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 8	Odgovarajući	Neprikazano pitanje iz grupe pitanja Algoritam	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
Pitanje 9	Točno/Netočno	Kraj grupe pitanja Algoritam - nastavak	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...
<b>Kraj grupe pitanja</b>	Kraj grupe pitanja	Kraj lekcije	🔍 ⚙️ 🔍 ✖️	Dodaj novu stranicu...

Slika 4.14 Klaster (grupa pitanja) unutar lekcije

Na prethodnoj slici je vidljivo da smo unutar 3 nastavne jedinice napravili 9 pitanja koja su reprezentativna za koncepte unutar tih nastavnih jedinica.

Unutar svakog klastera moguće je dodati neograničen broj pitanja. To se napravi na jednostavan način: *Uredi* → *Dodaj ovdje novu stranicu s pitanjem*. Sada ćemo objasniti kako smo dodali pitanja u klaster unutar lekcije „*Programiranje, programska podrška, algoritam*“. Odlučili smo unutar ove grupe, kao i svih ostalih koje ćemo napraviti, ubaciti tri ili četiri pitanja, ovisno o sadržaju koji možemo njima provjeravati.

Klikom na *Uredi* → *Dodaj ovdje novu stranicu s pitanjem* dobijemo prozor:

## Programiranje, programska podrška, algoritam <sup>?</sup>

▼ Create a question page

Odaberite vrstu pitanja

Višestruki odabir ▼

Slika 4.15 Dodavanje pitanja unutar klastera

Možemo odabrati koju vrstu pitanja želimo dodati unutar grupe pitanja: **Brojčani**, **Esej**, **Kratki odgovor**, **Odgovarajući**, **Točno/Netočno**, **Višestruki odabir**:

- **Brojčani** → Daje mogućnost da kod postavljenog pitanja učenik može pored odgovora koji smatra točnim staviti npr. broj 1, a pored netočnog 0. Unutar lekcija koristit ćemo ovakav način postavljanja pitanja, što će kasnije biti prikazano.
- **Esej** → Učeniku se postavi pitanje na koje on mora dati cjelovit odgovor koji se sastoji od niza smislenih rečenica. Ovakva pitanja su često nepregledna jer njihovu točnost ne može provjeriti sam sustav, već ih mora analizirati sam nastavnik. Zato je često stavljeno ograničenje da broj riječi pri pisanju esejskog odgovora ne prelazi broj 100 ili 150. Mi nećemo koristiti ovakva pitanja unutar nijedne grupe pitanja.
- **Kratki odgovor** → U ovakvim pitanjima se od učenika na postavljeno pitanje zahtjeva točan kratak odgovor. Velika mana ovog načina postavljanja pitanja je što sustav ne razlikuje različit raspored riječi unutar niza od dvije ili tri riječi, ne razumije deklinaciju riječi. Odgovor će biti točan samo ako je unesena riječ točno onako kako ju je napisao nastavnik u postavkama ovog pitanja. Zato je poželjno izbjegavati ovakva pitanja ili ih koristiti samo onda kada je odgovor neka kratka riječ ili recimo, kratica ili oznaka za neki pojam. Osim što je ovo jedina vrsta pitanja koja se ne ocjenjuje automatski, ova pitanja nije moguće dodavati slučajnim odabirom u test.
- **Odgovarajući** → Ovdje se također unosi točan i netočan odgovor, kao i povratna informacija o ostvarenom broju bodova. Dodatno se mora unijeti barem dva povezana para, te navesti koji se odgovor preklapa s kojim. To je zapravo povezivanje pojmova s lijeve strane onima s desne. Ovu vrstu pitanja nismo koristili u projektu.
- **Točno/Netočno** → Predstavljaju najjednostavnija pitanja. To su pitanja na koja se može odgovoriti samo sa točno ili netočno. Ova pitanja smo izbjegavali gotovo u svim stereotipovima. Koristili smo ih kod stereotipa „Novak“ i „Početnik“, dok kod ostalih stereotipova se od učenika zahtijeva veća razina znanja te ih nismo koristili.

**Višestruki odabir** → Od više ponuđenih odgovora učenik bira ili samo jedan koji je točan ili više njih ukoliko ih takvih ima. Ova vrsta pitanja korištena je unutar svake grupe pitanja. Ukoliko, recimo, pitanje tipa višestrukih odgovora sadrži dva točna odgovora i student odabere samo jedno kao točno od navedenih, onda u ovom slučaju dobiva 50% od broja ukupno mogućih bodova, tj. 0,5 bodova. Ako su tri točna odgovora, dobiva 0,33% od ukupnog broja bodova i sl. Negativnih bodova za netočne odgovore nema.

**Programiranje, programska podrška, algoritam** 🔗

▼ Create a question page ▼ Sažmi sve

**Naslov stranice\***

**Sadržaj stranice\***

Označite pojmove koji pripadaju fazama programiranja:

Višestruki odgovor

---

**Odgovor\***

Moodle auto-oblik ▼

**Povratna informacija**

Moodle auto-oblik ▼

**Predi na** ?

**Rezultat** ?

▼ **Odgovor 2**

**Odgovor\***

Moodle auto-oblik ▼

**Povratna informacija**

Moodle auto-oblik ▼

**Predi na** ?

**Rezultat** ?

Slika 4.16 Grananje unutar klastera ovisno o odgovoru

Vidimo da su naslov i sadržaj stranice obavezna polja koja se moraju ispuniti. *Naslov stranice* označava naziv pitanja, a *sadržaj stranice* odnosi se na tekst pitanja koji će biti prikazan učeniku i na osnovu kojeg će dati odgovor na pitanje. Opcija *Višestruki odgovor* je uključena što znači da je u ovom pitanju više točnih odgovora. U bloku *Odgovor 1*, zvjezdica iznad *Odgovora* označava da se polje mora popuniti i to je tekst koji će učeniku biti prikazan kao potencijalni odgovor na postavljeno pitanje. Ispod u padajućem izborniku možemo odabrati kakav će oblik imati priloženi odgovor: *HTML oblik*, *Moodle auto-oblik*, *Tekstualni oblik* ili *Markdown oblik*. Mi smo koristili *Moodle auto-oblik* koji se i preporučuje. U sljedećem polju *Odgovor* nastavnik unosi tekst koji će biti povratna informacija je li odabrani odgovor prikazan u polju iznad točan ili netočan, a *Rezultat* prikazuje dodijeljeni broj bodova. Ukupan broj bodova koje učenik može dobiti za svako točno odgovoreno pitanje je 1. Odlučili smo da u slučaju netočnog odgovora nećemo dodjeljivati negativne bodove iako je i to često pri raznim provjerama učenika. Pristup sljedećoj lekciji je onemogućen sve dok učenik ne odgovori točno barem na 50% pitanja iz prethodne lekcije, pa je i to razlog nedavanja negativnih bodova za netočne odgovore. Dakle, u ovom slučaju ukoliko učenik odgovori točno dobiva 1 bod, a netočno 0 bodova. Ukoliko je odgovor točan, učenika se šalje na sljedeće pitanje, a kraj klastera učenika šalje na kraj lekcije. Na prethodnoj slici je vidljivo da smo u postavama odabrali da nakon točnog odgovora, učeniku se prikaže „*Neprikazano pitanje unutar grupe pitanja*“. To je napravljeno iz razloga da se učeniku pitanja prikažu nasumičnim redoslijedom. Učeniku se neće prikazivati više pitanja kada na sva pitanja odgovori s točnim odgovorom.

Na prethodno opisan način smo izradili nastavne jedinice za stereotip „*Novak*“.

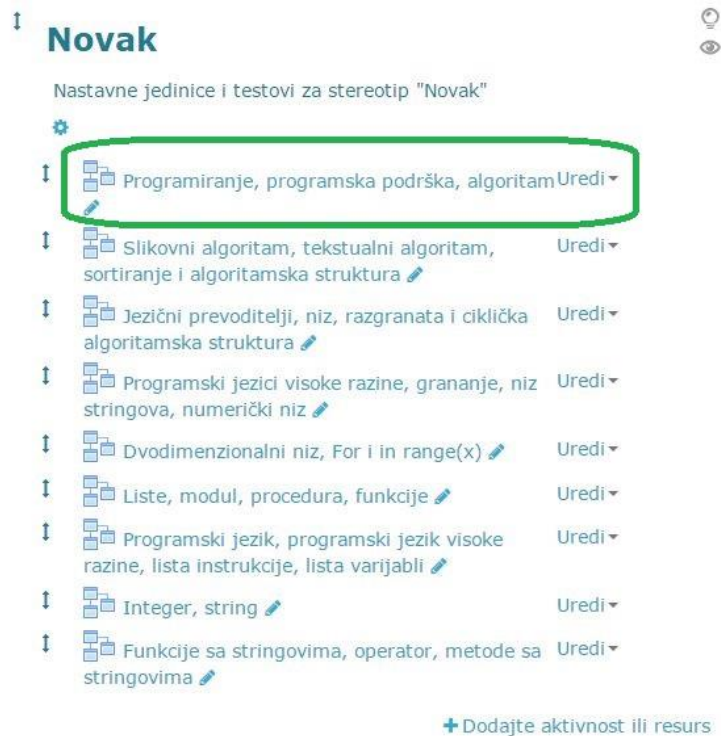
Tablica ispod je dio strukture statičkog računalom oblikovanog nastavnog sadržaja za stereotip „*Novak*“.

#### Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip „*Novak*“

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Programiranje	1	0	Nastavna jedinica	3
Programska podrška	1	1	Nastavna jedinica	3
Algoritam	1	1	Nastavna jedinica	10

TEST – zbog promjene udaljenosti od korijena





Slika 4.17 Stereotip „Novak“

Prethodna tablica i slika su umetnute da se vidi povezanost prvog klastera unutar stereotipa „Novak“ i dijela strukture statičkog računalom oblikovanog nastavnog sadržaja za stereotip „Novak“. Test zbog promjene udaljenosti od korijena je izrađen pomoću klastera, kao i svi ostali testovi unutar stereotipa „Novak“.

U sljedećem potpoglavlju ćemo objasniti postupak izrade preostalih stereotipova i kriterije prema znanju koje smo odredili za učenikovo napredovanje na višim razinama znanja.

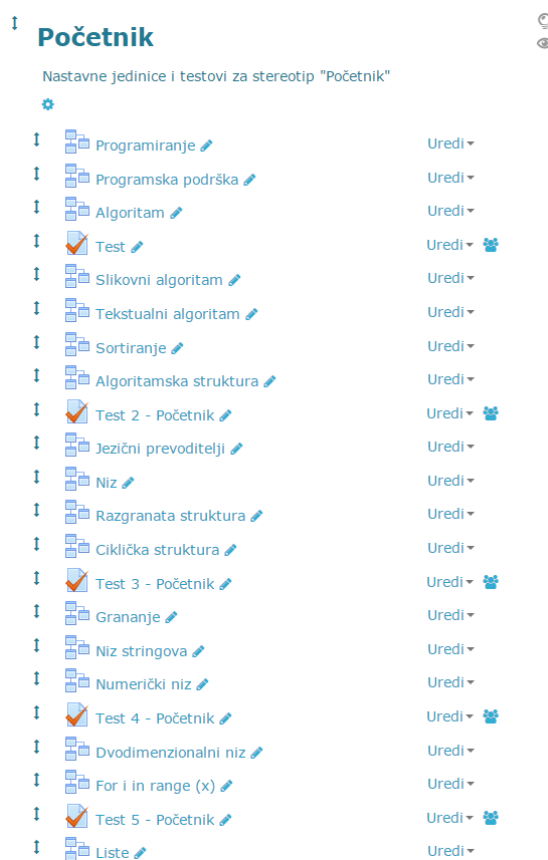
#### 4.3.4. Kriteriji za napredovanje prema znanju

Nakon što smo stereotip „Novak“ implementirali u nizovima nastavnih jedinica koje su predstavljene lekcijama i testovima koji su predstavljeni klasterima, sljedeće stereotipove ćemo implementirati na malo drugačiji način.

Ideja je bila da preostale stereotipove dizajniramo na način da svaki nastavni element (nastavna cjelina, nastavna tema i nastavna jedinica), ovisno o kojem se stereotipu radi, dizajniramo na sljedeći način:

- Svaki nastavni element će biti prezentiran u obliku lekcije izrađenoj u sustavu Moodle, a test koji se javlja ili zbog promjene udaljenosti od korijena, niza od tri nastavna elementa ili kraja cjeline će biti prezentiran u obliku testa izrađenom u sustavu Moodle.

Razlika u odnosu na test u stereotipu „Novak“ koji je prezentiran klasterima, test izrađen u ovakvom obliku nema mogućnost da se nakon pogrešnog odgovora učenika vrati na stranicu s točnim odgovorom. Zbog toga smo iskoristili neke od funkcionalnosti Moodle-a koje nam omogućavaju željeni slijed napredovanja kroz nastavne elemente, a sve u cilju što učinkovitijeg procesa učenja i poučavanja. Prvo ćemo objasniti proces izrade lekcije i testova u tom procesu objasniti kriterije koje smo koristili za što bolje učenikovo napredovanje prema znanju.



Slika 4.18 Dio nastavnog sadržaja za stereotip „Početnik“ izrađenom na sustavu Moodle

Lekciju na Moodle je jednostavno dodati: kliknemo na *Dodajte aktivnost ili resurs* → *Lekcija*.

Sada ćemo objasniti postavke naših lekcija. To ćemo objasniti na primjeru 2. lekcije: „Slikovni algoritam, tekstualni algoritam i algoritamska struktura“ Ovako ona izgleda na Moodle-u:

Slika 4.19 Postavke lekcije

U bloku **Općenito** moramo u prvo polje unijeti *Ime* lekcije i ono je obavezno (što nam pokazuje crvena zvjezdica iznad). Svi ostali podaci u ovom bloku mogu se izostaviti.

Blok **Lekcija koja je preduvjet** nudi četiri mogućnosti. U našim postavkama lekcija „*Programiranje*“ ne zavisi ni o jednoj drugoj lekciji budući da se od nje počinje. Međutim, lekcija „*Programska podrška*“ ovisi o prethodnoj lekciji i njoj nije moguće pristupiti ukoliko nije pregledana prethodna lekcija. „*Utrošeno vrijeme (u minutama)*“ daje mogućnost ograničavanja da se određenoj lekciji ne može pristupiti ukoliko se nije proveo određeni broj minuta na prethodnoj lekciji, čiji je pregled nužan kako bi se pristupilo sljedećoj lekciji. Uvjet „*Ocjena viša od (%)*“ predstavlja postotak da nije moguće pristupiti sljedećoj lekciji ukoliko na prethodnoj nije ostvaren određeni prolaz u postotcima. Može se koristiti bilo koja kombinacija utrošenog vremena, završenosti ili ocjene više od.

U postavkama smo postavili uvjete za napredovanje:

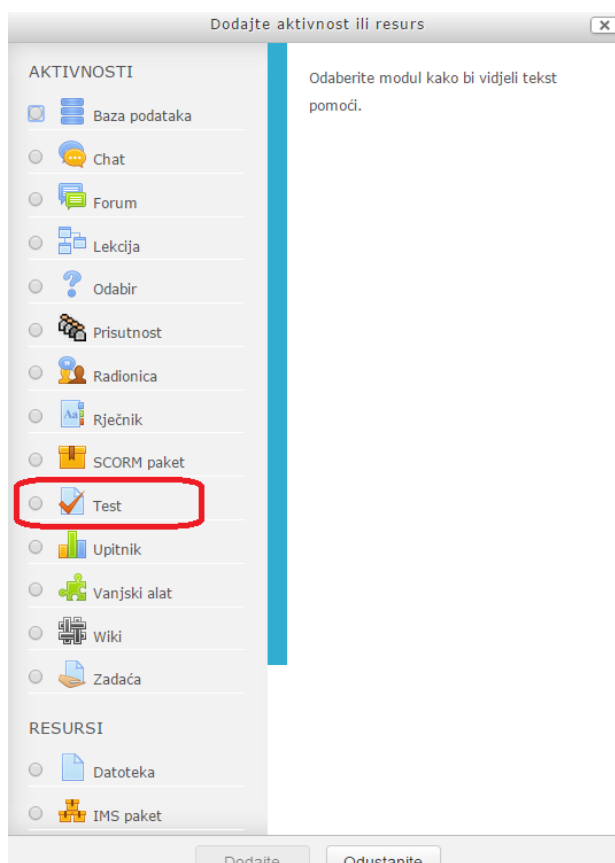
- „**Zavisnost**“: svaka lekcije zavisi o prethodnoj. Na taj način smo prisilili učenike da pravilno „prolaze“ lekcije, po već unaprijed određenom slijedu.
- „**Utrošeno vrijeme (u minutama)**“: to nismo smatrali da je ključno iako se može postaviti vremensko ograničenje za koje se smatra da je potrebno učeniku za proučiti zadani nastavni sadržaj.
- „**Završeno**“: završeno predstavlja uvjet da li je učenik došao do kraja lekcije. To sustav prepoznaje jer se na kraju lekcije nalazi gumb kojim se potvrđuje kraj lekcije

- „*Ocjena viša od (%)*“: uvjet koji predstavlja rezultat koji je učenik ostvario u testu izrađenom pomoću klastera.

#### 4.3.5. Kriteriji za napredovanje prema znanju

Test se dodaje kao i preostale aktivnosti na *Moodle-u*, odabirom „*Dodajte aktivnost ili resurs*“.

Unutar testa pristupamo izradom pitanja za koja smatramo da su reprezentativna za određene koncepte koje učenici moraju usvojiti te relacije kojim su ti koncepti povezani. U ovom stereotipu smo izradili dosta pitanja, te smo ta pitanja iskoristili i za sljedeće stereotipove, iako smo kod preostalih stereotipova izbjegavali pitanja tipa „*Točan/netočan*“ jer ta vrsta pitanja su „odgovarajuća“ za ovaj stereotip, dok za ostale ta vrsta pitanja nije prikladna jer se u sljedećim stereotipovima zahtijeva veća razina znanja prema Bloom-ovoj taksonomiji znanja koju smo obradili u prethodnim poglavljima.



Nakon dodavanja testa, obvezno je upisati naziv testa. Test sadrži razne postavke, kao što su: *vrijeme* (određivanje vremena u kojem je test dostupan), *ocjena* (postavke o načinu ocjenjivanja i o dopuštenom broju pokušaja), *karakteristike pitanja* (ovdje smo iskoristili mogućnost da se izmjeni redoslijed odgovora) i dr.

Nakon dodavanja testa pristupa se procesu odabira odgovarajućih pitanja. Pitanja smo izradili nakon izrade svih nastavnih jedinica. Za svaki koncept unutar mape koncepata je izrađeno 2-3 pitanja, a sva pitanja se nalaze u bazi pitanja na sustavu *Moodle*. Dakle ukupno smo izradili 104 pitanja za koja smatramo da su reprezentativna za usvajanje odabranog nastavnog sadržaja. Na slici 4.18 se može primijetiti da test nastupa nakon niza od 3 nastavne cjeline. Dakle, tu smo obradili 3 koncepta, dok u testu izrađenog za to područje se nalazi 9 pitanja.

Primjer odabranih pitanja iz baze pitanja za test zbog promjene udaljenosti od korijena za stereotip „Početnik“ se nalazi na sljedećoj slici.

Ukloni označeno

Umetni nove stranice poslije odabranih pitanja

Odaberi sve / Odznači sve

Premjesti odabrana pitanja na stranicu:  Premjesti

Promijeni redoslijed pitanja

Stranica 1

1	<input type="checkbox"/>	•• Programiranje 1 Programiranje je postupak koji...	Q *	10
2	<input type="checkbox"/>	☰ Programiranje 2 Programiranje na najvišoj razini...	Q *	20
3	<input type="checkbox"/>	☰ Programiranje 3 Označite pojmove koji pripadaju...	Q *	30
4	<input type="checkbox"/>	•• Programska podrška Programska podrška se sastoji o...	Q *	40
5	<input type="checkbox"/>	☰ Sistemska prog. podrška Sistemska programska podrška ...	Q *	50
6	<input type="checkbox"/>	•• Algoritam Algoritam je djelomično opisan ...	Q *	60
7	<input type="checkbox"/>	☰ Algoritam 2 Označite karakteristike algoritma:	Q *	70
8	<input type="checkbox"/>	☰ Algoritam 3 Pridružite odgovarajuće tvrdnje ...	Q *	80
9	<input type="checkbox"/>	•• Algoritam 4 Algoritam se ne može prikazati s...	Q *	90

Ukloni označeno

Promijeni redoslijed pitanja

Odaberi sve / Odznači sve

Premjesti odabrana pitanja na stranicu:  Premjesti

Slika 4.20 Primjer testa za stereotip „Početnik“ nakon niza od 3 nastavne jedinice

Na isti način se pristupilo izradi preostalih testova za sve stereotipove. Proces u kojem se postavljaju prethodno objašnjeni kriteriji treba pristupiti sistematično. Izuzetno je važno da se kriteriji postave pravilno. Prilikom rada na postavljaju kriterija je bio problem kod kriterija „zavisi o“ jer *Moodle* u padajućem izborniku nudi sve nastavne sadržaje koji se nalaze u kolegiju. Kod nas je slučaj da se ista nastavna jedinica nalazi u 4 stereotipa (kod stereotipa „Novak“ nastavne jedinice imaju različite nazive jer su oblikovani pomoću klastera), te zbog toga dolazi do zbrke i poteškoće kod odabire pravilne nastavne jedinice. U tom slučaju treba odabrati određenu nastavnu jedinicu koja pripada željenom stereotipu, a to je otežano jer se nastavne jedinice nalaze kod 4 stereotipa.

## Provjera predznanja

Nakon što smo implementirali svih 5 stereotipova, pristupili smo izradi testa koja će nam služiti za provjeru predznanja učenika. Na isti način funkcionira i *AC-Ware Tutor*. Dakle, ideja je da prije samog pristupanja procesu učenja i poučavanja provjeri učenikovo predznanje iz područnog znanja koje će se poučavati pomoću sustava. To je zapravo prvi korak kod svih sustava koji žele prilagoditi proces učenja i poučavanja učeniku. Ulazni test osim što služi provjeri predznanja učenika, također služi da se lakše i točnije prati učenikov proces u savladavanju nastavnog sadržaja u određenim stereotipovima. Pitanja koja se nalaze u ulaznom testu su podskup pitanja iz baze pitanja koju smo napravili za stereotipove. Odabrali smo manji uzorak pitanja za koji smatramo da može utvrditi učenikovo predznanje iz odabranog područnog znanja. U brojevima, to bi značilo da smo od ukupno 104 pitanja, u ulazni test odabrali 24 reprezentativna pitanja za provjeru predznanja.



Slika 4.21 Ulazni test na sustavu Moodle

Stranica 1

1	•• Algoritam <b>Algoritam je djelomično opisan ...</b> Q *	10	↓ ⊗
2	•• Dijagram toka <b>Algoritam se grafički predočava ...</b> Q *	20	↑ ↓ ⊗
3	•• For i in range (x) - Range <b>Funkcija RANGE() u petlji "for i ...</b> Q *	30	↑ ↓ ⊗
4	⇒ For i in range (x) 10 <b>Što će ispisati sljedeći dio progr...</b> Q *	40	↑ ↓ ⊗
5	≡ Funkcija 2 <b>Označite točne tvrdnje.</b> Q *	50	↑ ↓ ⊗

Stranica 2

6	•• Grananje - Operator <b>Operator - operator je «funkcija...</b> Q *	70	↑ ↓ ⊗
7	⇒ Integer <b>Upišite naredbu za izračunavanj...</b> Q *	80	↑ ↓ ⊗
8	•• Jezicni prevoditelji - Objektni k... <b>Kompilator je program koji prev...</b> Q *	90	↑ ↓ ⊗
9	≡ Lista instrukcija, program <b>Povežite pojmove s odgovarajuć...</b> Q *	100	↑ ↓ ⊗
10	≡ Liste <b>Koji dio koda neostaje da bi se d...</b> Q *	110	↑ ↓ ⊗

Slika 4.22 Dio odabranih pitanja za provjeru predznanja učenika

U sustavu *AC-Ware Tutor*, ulazni test služi da se dinamički odredi stereotip koji će se odrediti učeniku, ovisno o rezultatu ulaznog testa. Pošto *Moodle* nema tu razinu prilagođavanja mi smo

odredili bodovni rang te napisali upute i usmeno objasnili studentima na testiranju kojem će rangu pristupiti ovisno o rezultatu na ulaznom testu.

Rezultati učenika kod ulaznog testa, te njihovi rezultati na sustavu razvijenog u svrhu ovog diplomskog rada na sustavu *Moodle*, služe za točniju i detaljnu analizu. Kod analize će se utvrditi prednosti i mane oblikovanog nastavnog sadržaja, a analiza je detaljno objašnjena u sljedećem poglavlju.

## 5. Vrednovanje modela

Vrednovanje modela se provelo na Prirodoslovno-matematičkom fakultetu u Splitu. Model su vrednovali studenti 1. godine diplomskog studija „Informatika“. Vrednovanje se provelo na sljedeći način: studentima se objasnio rad oblikovanog modela, te su nakon toga pristupili rješavanju ulaznog testa. Rezultati ulaznog testa su pokazatelji njihova znanja iz odabranog područnog znanja te im se usmenim putem objasnio kojem će stereotipu pristupiti u proces učenja i poučavanja. Ukoliko su na svakom testu unutar stereotipa imali preko 50% bodova mogli su pristupiti sljedećem stereotipu. Svaki rezultat testa unutar stereotipa se zapisivao kao i promjena stereotipa.

Svi ispitanici su napredovali prema znanju koristeći model. Da bi omogućili usporedbu oblikovanog nastavnog sadržaja pomoću klastera i pomoću lekcija i testova opisanog u prethodnim poglavljima, svi ispitanici su pristupili stereotipu „Novak“ koji je oblikovan pomoću klastera. Na kraju vrednovanja ispitanicima je dan izlazni upitnik koji se sastoji od 7 tvrdnji u obliku Likertove skale s 5 tvrdnji, te jednim esejskim pitanjem.

Izlaznim upitnikom smo dobili podatke da se ispitanicima sviđa oblikovani nastavni sadržaj i da je ostvareno napredovanje prema znanju. Također, izlaznim upitnikom smo saznali da se ispitanicima sviđa ovakav oblik učenja i poučavanja, iako navode nedostatak multimedije. Iako je vrednovanju pristupio mali broj ispitanika (4), dobili smo nove uvide i saznanja o oblikovanom modelu. Izlazni upitnik se nalazi u **prilogu – c** diplomskog rada.



## 6. Zaključak

Cilj ovoga rada je oblikovati nastavni sadržaj s karakteristikama prilagodljivosti, uz naravno i cilj svakog nastavnog sadržaja, to je napredovanje prema znanju. Sustav Moodle je zbog svojih karakteristika našao široku primjenu u obrazovnim ustanovama, ali je bio izazov na tom sustavu oblikovati nastavni sadržaj s karakteristikama prilagodljivosti. Stereotipni model smo smatrali prikladnim upravo zbog načina napredovanja prema znanju, te zbog sukladnosti s Bloom-ovom taksonomijom znanja. Klasteri su novost u sustavu Moodle 2.x, te smo htjeli saznati njihove mogućnosti. Rezultati vrednovanja su nam dali informacije da je napredovanje prema znanju ostvareno, kao i prilagodljivost, iako smatramo da je sustav Moodle dosta „krut“ u pogledu prilagodljivog nastavnog sadržaja.

Stereotipni model se pokazao primjerenim upravo zbog toga jer se prilikom razvoja nastavnog sadržaja pristupa prvo najmanjim oblikom nastavnog sadržaja (nastavne jedinice). Nakon razvoja svih nastavnih jedinica, sustav Moodle omogućuje jednostavno oblikovanje većim oblicima nastavnog sadržaja (nastavne teme i nastavne cjeline), što u konačnici omogućuje efikasni razvoj preostalih stereotipova.

Klasteri su novost u sustavu Moodle i njihova prednost u odnose na „uobičajeni“ nastavni sadržaj na sustavu Moodle je upravo mogućnost povratka učenika na prethodno gradivo ukoliko netočno odgovori na pitanje. Time se prekida linijsko izvođenje nastavnog sadržaja, povratna informacija je trenutna a učenika se odmah „ispravlja“ i vraća na dio nastavnog sadržaja kojeg treba bolje usvojiti.

Kod vrednovanja je bio nedostatak mali broj ispitanika, ali ispitanici su omogućili povratnu informaciju koja ukazuje na moguća poboljšanja samoga modela. Naime, ispitanici ističu upravo gore navedene karakteristike klastera kao karakteristiku prilagodljivosti i pozitivnu karakteristiku tutora koji trenutno ispravlja učenike u slučaju netočnog odgovara. U vrednovanju modela, učenike se usmjerilo na stereotip u ovisnosti o njihovim rezultatima na ulaznom testu, dok su na kraju pristupili stereotipu „Novak“ koji je prema znanju, najniži rang. Na taj način rezultati i nisu u potpunosti vjerodostojni. Bolji način bi bio koristeći stereotipni model dok bi jedna ispitna skupina pristupila procesu učenja i poučavanja s nastavnim sadržajem oblikovanom pomoću klastera, a druga pomoću niza lekcija i testova. Oblikovani model je ostvario cilj: „*oblikovati nastavni sadržaj s karakteristikama prilagodljivosti, uz naravno i cilj svakog nastavnog sadržaja, to je napredovanje prema znanju*“. Ostavljena su i otvorena pitanja:

- da li je u sustavu Moodle moguće ostvariti veću prilagodljivost?
- da li su klasteri korak naprijed prema prilagodljivosti u sustavu Moodle?

## Literatura

- [1] GRUBIŠIĆ, A.: „Vrednovanje učinka inteligentnih tutorskih sustava e – učenja“, Doktorski rad, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva, 2012.
- [2] STANKOV, S.: „Inteligentni tutorski sustavi“, 2010.
- [3] KHAN, B.: „A Framework for E-Learning“, 2012
- [4] IRLBECK, S. & MOWAT, J.: „Learning Content Management System (LCMS)“, 2005
- [5] NICHANI, M.: *Articles about LCMS, LMS, CMS and RLOS*, 2001
- [6] CLARK, D.: „ADDIE timeline“, 1995
- [7] CLARK, D.: „Instructional System Design Concept Map“, 2004
- [8] CHICO STATE IDTS: „ADDIE model“, 2009
- [9] WAY, R.: „Using ADDIE Model“, 2006
- [10] EDUTECH WIKI, INSTRUCTIONAL DESIGN  
[[HTTP://EDUTECHWIKI.UNIGE.CH/EN/INSTRUCTIONAL\\_DESIGN](http://edutechwiki.unige.ch/en/Instructional_Design) , 02.12.2013
- [11] HILL, P.: „Online Educational Delivery Models: A Descriptive View“, 2012
- [12] SIEMENS, G.: „Instructional Design in Elearning“, 2002
- [13] WIKIPEDIA, E-UČENJE
- [14] ALEKSIĆ-MASLAĆ, K.: „Online obrazovanje u Hrvatskoj – bliža ili dalja budućnost“, 2013
- [15] MALAMED, C.: „A Framework for Developing Online Learning“, 2013.
- [16] Službena stranica sustava Moodle, [www.moodle.org](http://www.moodle.org)
- [17] BOŽIĆ, H.N., „Korištenje sustava za učenje (LMS) u e-obrazovanju“, 2013
- [18] MARIČIĆ, S. & BANDALO, M.: „Alati za upravljanje sadržajem“
- [19] D. VELJAN, *Kombinatorna i diskretna matematika, Algoritam*, 2001.
- [20] EDUTECH WIKI, Instructional design,  
[http://edutechwiki.unige.ch/en/Instructional\\_design](http://edutechwiki.unige.ch/en/Instructional_design) , 02.12.2013.
- [21] WIKIPEDIA, *Instructional Design*  
[http://en.wikipedia.org/wiki/Instructional\\_design#History\\_of\\_the\\_System\\_Approach\\_to\\_Instructional\\_Design](http://en.wikipedia.org/wiki/Instructional_design#History_of_the_System_Approach_to_Instructional_Design), 02.12.2013.

## Sažetak

### **Pristup ostvarivanju prilagodljivosti prema znanju u sustavima za upravljanje učenjem**

Rad pokriva područje istraživanja u sustavima e-učenja, te nastoji oblikovati nastavni sadržaj na način koji ima karakteristike prilagodljivosti. Razvojem sustava e-učenja, realizacija prilagodljivosti je jedna od najvećih težnji sustava e-učenja današnjice, a u ovom radu prilagodljivost smo bazirali prema sustavu AC-ware Tutor. Za oblikovani nastavni sadržaj odabrano je područno znanje koje odgovara kolegiju „Programiranje 1“. Nastavni sadržaj je oblikovan na sustavu Moodle. Moodle ima veliku primjenu u obrazovanju, dok je u ovom radu naglasak na prilagodljivosti koja u Moodle nije „ugrađena“ već se nastoji postići korištenjem raznih funkcionalnosti sustava Moodle. Oblikovani nastavni sadržaj je rangiran prema znanju u stereotipove koji predstavljaju trenutnu razinu usvojenog znanja, te je temeljen na Bloom-ovoj taksonomiji znanja. Dio oblikovanog nastavnog sadržaja je razvijen i implementiran pomoću klastera, a dio pomoću lekcija i testova na Moodle-u prema algoritmima za generiranje nastavnih sadržaja u svrhu usporedbe i vrednovanja nastavnog sadržaja.

Ključne riječi — sustavi e-učenja, Moodle, oblikovanje nastavnog sadržaja, , prilagodljivost, stereotipovi, courseware, klasteri.

# Summary

## **An approach to providing knowledge-based adaptivity in learning management systems**

This document covers research in the field of e-learning and also tries to design course content that has properties of adaptivity. Developing e-learning systems, reaching adaptivity of learning content became the prime goal of every e-learning systems. We based our work on e-learning system AC-Ware Tutor and we will explain how adaptivity is developed in that system. We selected domain knowledge of college course „Programming I“ and design a map of concept that contains representative concepts for selected domain knowledge. Domain knowledge was designed on e-learning system, Moodle. Moodle is a free and open-source software learning management system. Developed on pedagogical principles, Moodle is used for blended learning, distance education, flipped classroom and other e-learning projects in schools, universities, workplaces and other sectors. Moodle doesn't have properties of adaptivity, but we tried to design course content using all functionality of system Moodle. For reaching knowledge-based adaptivity, course content is divided in five stereotypes, which represent the current level of student knowledge of course content and all course content is developed based on Bloom's taxonomy of knowledge. We wanted to use clusters so we could test their advantages and properties of adaptivity and because of that one part of course content is design using clusters and another part is design using sequence of lessons and tests, also in Moodle, following algorithms for generating course content.

Keywords – e-learning, Moodle, instructional design, adaptivity, stereotypes, courseware, clusters.

## Skraćenice

ICT	<i>Information and Communication Technology</i>	informacijske i komunikacijske tehnologije
LMS	<i>Learning Management Systems</i>	sustav za upravljanje učenjem
HR	<i>Human Resource</i>	ljudski resursi
CMS	<i>Content Managment System</i>	sustav za upravljanje sadržajem
LCMS	<i>Learning Content Management Systems</i>	sustav za upravljanje sadržajem učenja
RLO	<i>Reusable learning objects</i>	ponovno upotrebljivi objekti učenja
ID	<i>Instructional Design</i>	oblikovanje nastavnog sadržaja
Moodle	<i>Modular Object-Oriented Dynamic Learning Environment</i>	
AC-Ware Tutor	<i>Adaptive Courseware Tutor model</i>	

# Prilozi

## Prilog A - Elementi računalom oblikovanog nastavnog sadržaja „Programiranje“

Rbr	Korijen elementa	Oznaka cjeline	Vrsta elementa	Udaljenost od korijena određene cjeline	Broj koncepata	Popis koncepata koji se nalaze u elementu
1.	Programiranje	Programiranje	Nastavna cjelina	0	75	8 kraljica Algoritam Algoritamska struktura AND append() Bubble sort Ciklička Dijagram toka Dvodimenzionalan niz Dvodimenzionalan niz extend() for i in range(x) Funkcije Globalne varijable Grananje Grananje Hanojski tornjevi if-elif Indeks Indeks petlje Insertion sort Interpretator Izlaz Izlaz Izvediv Jednodimenzionalan niz Jednodimenzionalan niz Jednoznačan Jezični prevoditelji Kompilator Korak rekurzije Korektan len(L) Linijska Liste Lokalne varijable Matematičke funkcije

						Matrica Matrica max(L) min(L) Moduli Niz Niz stringova NOT Numerički Obrada podataka Odluka OR Osnovni slučaj Potprograma Početak/kraj Procedure Programiranje Programske podrške Programski jezik visoke razine Pseudokod Python Quick sort Raspon Razgranata Rekurzija Selection sort Sistemska programska podrška Slikovni algoritam Sortiranje Tablica Tablica Tekstualni algoritam Ugrađene funkcije Ulaz Ulaz Uporabljiv Uvjet While
2.	Algoritam	Programiranje	Nastavna jedinica	1	10	Ulaz Sortiranje Uporabljiv Korektan Izlaz Izvediv Jednoznačan Slikovni algoritam Algoritamska struktura Tekstualni algoritam
3.	Algoritamska struktura	Programiranje	Nastavna tema	2	9	Algoritamska struktura Linijska



						Ciklička Razgranata If-elif Grananje Grananje While For i in range(x)
4.	Algoritamska struktura	Programiranje	Nastavna jedinica	2	4	Algoritamska struktura Razgranata Ciklička Linijaska
5.	Razgranata	Programiranje	Nastavna jedinica	3	3	Razgranata If-elif grananje
6.	Ciklička	Programiranje	Nastavna jedinica	3	4	Ciklička Grananje While For i in range (x)
7.	For i in range (x)	Programiranje	Nastavna jedinica	5	3	For i in range (x) Raspon Indeks polja
8.	Grananje	Programiranje	Nastavna jedinica	4	5	Grananje Uvjet AND NOT OR
9.	Tekstualni algoritam	Programiranje	Nastavna jedinica	2	2	Tekstualni algoritam Pseudokod
10.	Slikovni algoritam	Programiranje	Nastavna jedinica	2	7	Slikovni algoritam Dijagram toka Izlaz Ulaz Obrada podataka Početak/kraj Odluka
11.	Sortiranje	Programiranje	Nastavna tema	2	9	Sortiranje Selection sort Insertion sort Quick sort Bubble sort Niz stringova Numerički Indeks Niz

12.	Sortiranje	Programiranje	Nastavna jedinica	2	6	Sortiranje Selection sort Insertion sort Quick sort Bubble sort Niz
13.	Niz	Programiranje	Nastavna jedinica	3	4	Niz stringova Numerički Indeks Niz
14.	Niz stringova	Programiranje	Nastavna tema	4	5	Niz stringova Jednodimenzionalni niz Dvodimenzionalni niz Tablica Matrica
15.	Niz stringova	Programiranje	Nastavna jedinica	4	3	Niz stringova Jednodimenzionalni niz Dvodimenzionalni niz
16.	Dvodimenzionalni niz	Programiranje	Nastavna jedinica	5	3	Dvodimenzionalni niz Tablica Matrica
17.	Numerički	Programiranje	Nastavna tema	4	5	Numerički Jednodimenzionalni niz Dvodimenzionalni niz Tablica Matrica
18.	Numerički	Programiranje	Nastavna jedinica	4	3	Numerički Jednodimenzionalni niz Dvodimenzionalni niz
19.	Dvodimenzionalni niz	Programiranje	Nastavna jedinica			Dvodimenzionalni niz Tablica Matrica
20.	Programiranje	Programiranje	Nastavna tema	0	15	Programiranje Algoritam Ulaz Sortiranje Uporabljiv Korektan Izlaz Izvediv Jednoznačan Slikovni algoritam Algoritamska struktura Tekstualni algoritam Programska podrška Sistemska programska podrška Jezični prevoditelj
21.	Programiranje	Programiranje	Nastavna jedinica	0	3	Programiranje

						Algoritam Programska podrška
22.	Programska podrška	Programiranje	Nastavna jedinica	1	3	Programska podrška Sistemska programska podrška Jezični prevoditelj
23.	Jezični prevoditelj	Programiranje	Nastavna tema	3	8	Jezični prevoditelj Interpreter Kompilator Programski jezik visoke razine Python Procedure Moduli Liste
24.	Jezični prevoditelj	Programiranje	Nastavna jedinica	3	4	Jezični prevoditelj Interpreter Kompilator Programski jezik visoke razine
25.	Programski jezik visoke razine	Programiranje	Nastavna jedinica	4	5	Programski jezik visoke razine Python Procedure Moduli Liste
26.	Liste	Programiranje	Nastavna jedinica	6	6	Liste Append() Extend() Len() Max() Min()
27.	Moduli	Programiranje	Nastavna jedinica	6	3	Moduli Ugrađene funkcije Matematičke funkcije
28.	Procedure	Programiranje	Nastavna tema	6	10	Procedure Lokalne varijable Globalne varijable Potprogrami Funkcije Rekurzija Osnovni slučaj Korak rekurzije Hanojski tornjevi 8 kraljica
29.	Procedure	Programiranje	Nastavna jedinica	6	5	Procedure Lokalne varijable Globalne varijable Potprogrami Funkcije
30.	Funkcije	Programiranje	Nastavna jedinica	7	6	Funkcije Rekurzija Osnovni slučaj Korak rekurzije

						Hanojski tornjevi 8 kraljica
31.	Programski jezik	Programski jezik	Nastavna cjelina	0	58	* + 8 kraljica <string>[:] <string>[] a="Dobar dan" abs(x) append() ceil() chr() extend() Funkcije Funkcije sa brojevima Funkcije sa stringovima Globalne varijable Hanojski tornjevi in Instrukcija Integer Korak rekurzije len() len(<string>) len(L) Liste Liste instrukcija Liste varijabli Lokalne varijable Matematičke funkcije max max max(L) Metode sa stringovima min min() min(L) Moduli not in Operacija Operator ord Osnovni slučaj Podstring Podstring Potprograma Procedure Programski jezik Programski jezik visoke razine Python

						Rekurzija round(broj x, broj decimala) str() String string.find(podstring) string.lower() string.replace(str1, str2) string.upper() Ugrađene funkcije Varijabla
32.	Programski jezik	Programski jezik	Nastavna tema	0	14	Programski jezik Programski jezik visoke razine Liste instrukcija Liste varijabli Varijable String Integer Lokalne varijable Procedure Instrukcija Operacije Python Modul Liste
33.	Programski jezik	Programski jezik	Nastavna jedinica	0	4	Programski jezik Programski jezik visoke razine Liste instrukcija Liste varijabli
34.	Programski jezik visoke razine	Programski jezik	Nastavna jedinica	1	5	Programski jezik visoke razine Python Modul Liste Procedure
35.	Liste instrukcija	Programski jezik	Nastavna jedinica	1	3	Liste instrukcija Instrukcija Operacije
36.	Liste varijabli	Programski jezik	Nastavna jedinica	1	5	Liste varijabli Varijable Lokalne varijable Integer String
37.	Liste	Programski jezik	Nastavna jedinica	3	6	Liste Append() Extend() Len() Max() Min()
38.	Moduli	Programski jezik	Nastavna jedinica	3	3	Moduli Ugrađene funkcije Matematičke funkcije

39.	Procedure	Programski jezik	Nastavna tema	3	10	<p>Procedure</p> <p>Lokalne varijable</p> <p>Globalne varijable</p> <p>Potprogrami</p> <p>Funkcije</p> <p>Rekurzija</p> <p>Osnovni slučaj</p> <p>Korak rekurzije</p> <p>Hanojski tornjevi</p> <p>8 kraljica</p>
40.	Procedure	Programski jezik	Nastavna jedinica	3	5	<p>Procedure</p> <p>Lokalne varijable</p> <p>Globalne varijable</p> <p>Potprogrami</p> <p>Funkcije</p>
41.	Funkcije	Programski jezik	Nastavna jedinica	4	6	<p>Funkcije</p> <p>Rekurzija</p> <p>Osnovni slučaj</p> <p>Korak rekurzije</p> <p>Hanojski tornjevi</p> <p>8 kraljica</p>
42.	Integer	Programski jezik	Nastavna jedinica	3	7	<p>Integer</p> <p>Funkcije sa brojevima</p> <p>Round(broj x, broj decimala)</p> <p>Ceil()</p> <p>Max()</p> <p>Abs()</p> <p>Min()</p>
43.	String	Programski jezik	Nastavna tema	3	24	<p>String</p> <p>a="Dobar dan"</p> <p>Podstring</p> <p>Funkcija sa stringovima</p> <p>Min()</p> <p>Len()</p> <p>Str()</p> <p>Ord()</p> <p>Max()</p> <p>Chr()</p> <p>Operator</p> <p>&lt;string&gt;[]</p> <p>*</p> <p>Ln</p> <p>+</p> <p>Not in</p> <p>len(&lt;string&gt;)</p> <p>&lt;string&gt;[:]</p> <p>Podstring</p> <p>Metode sa stringovima</p> <p>string.find(podstring)</p> <p>string.upper()</p>

						string.lower() string.replace(str1, str2)
44.	String	Programski jezik	Nastavna jedinica	3	7	String a="Dobar dan" Podstring Funkcija sa stringovima Metode sa stringovima Operator Podstring
45.	Metode sa stringovima	Programski jezik	Nastavna jedinica	5	4	Metode sa stringovima string.find(podstring) string.upper() string.lower() string.replace(str1, str2)
46.	Operator	Programski jezik	Nastavna jedinica	4	8	Operator <string>[] * Ln + Not in len(<string>) <string>[:]





## Prilog C – Izlazni upitnik

1. Lekcije na Moodle-u su mi se sviđele:						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
2. Nastavni sadržaj unutar lekcija je prikazan na jasan način:						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
3. Tok lekcija je jasan i nedvosmislen:						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
4. Rad na lekcijama je intuitivan:						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
5. Prelasci na više stereotipove su ostvareni:						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
6. Pitanja unutar testa prate nastavni sadržaj unutar lekcija						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
7. Pitanja unutar testa odgovaraju trenutnoj razini znanja (stereotipovima)						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
8. Ostvareno je konzistentnost unutar stereotipova						
U potpunosti se ne slažem	1	2	3	4	5	U potpunosti se slažem
9. Svojim riječima opišite svoje dojmove rada na oblikovanim nastavnim sadržajima na Moodle:						

