



ClassyCraft

Uvod

Urednik ClassyCraft je alat koji omogućava korisnicima da naprave sopstvene verzije klasnih dijagrama, koji služe da ilustrativno predstave arhitekturu, dizajn i implementaciju jednog softvera. Korisnik sam dodaje klase i povezuje ih po želji. ClassyCraft je aplikacija namenjena desktop korisnicima.

Rečnik

UML: grafički orijentisan jezik, koji služi da opiše sistem koji se razvija iz različitih uglova gledanja na sistem.

UML dijagram: obuhvata različite dijagrame i to: dijagram slučajeva korišćenja, dijagram aktivnosti, dijagram klasa, dijagram paketa, dijagram sekvenci, dijagram komponenti.

Klasni dijagram: Vizuelna reprezentacija klasa, interfejsa i njihovih odnosa u softverskom sistemu.

Export: čuvanje kopije trenutno otvorenog dokumenta, baze podataka, slike ili video zapisa u formatu datoteke.

Generisanje: automatsko stvaranje nečega.

Klasa: šablon za pravljenje objekata. Definiše svojstva (atribute) i ponašanja (metode) koje će njeni objekti imati.

Objekat: instance klase koja ima konkretne vrednosti za attribute definisane klasom i može izvršavati metode te klase.

Interfejs: skup apstraktnih metoda (bez implementacija) koje druge klase mogu implementirati. Služi kao garancija da će klase poštovati određeni standard.

Vidljivost: označava pristupačnost atributa ili metoda klase. Može biti public, private, protected i default.

Veza: opisuje odnos između dve ili više klasa.

Veza asocijacije: je najopštiji oblik veze u kome objekti jedne klase imaju neku strukturnu vezu ili odnos sa objektima druge klase.

Veza nasleđivanja (generalizacija): vrsta veze u kojoj jednu klasu “nasleđuje” druga klasa, pa se zato često naziva nasleđivanje.

Veza kompozicije: Predstavlja posebnu vrstu veze u kojoj objekat jedne klase, kao sastavni deo, sadrži jedan ili više objekata druge klase. Kod kompozicije, kada se obriše objekat koji sadrži druge objekte, brišu se svi objekti.

Veza agregacije: Slično kao kompozicija, klase su povezane, ali ako se obriše objekat, drugi objekti ne moraju biti obrisani.

Veza zavisnosti: Ukazuje na to da jedna klasa zavisi od druge. Ta veza je obično privremena (postoji samo za određeni period vremena kao što je tok izvođenja metode) i uvek jednosmerna.

Kardinalnost: broj koji označava koliko objekata jedne klase može biti u vezi sa objektom druge klase

Undo: radnja koja poništava najnoviju promenu ili operaciju, vraćajući sistem u prethodno stanje.

Redo: radnja koja ponovo primenjuje promenu ili operaciju nakon što je poništena.

Format: U programiranju odnosi se na raspored ili strukturu podataka, datoteka ili dokumenata.

Element: na nekim mestima gde stoji „element“, to se odnosi na klasu, koja može biti: enum, apstraktna, final, kao i na interfejs i različite tipove veze veze.

Konkurencija

U tabeli ispod je prikazana aplikacija ClassyCraft i druge konkurentne aplikacije čiji je zadatak pravljenje UML klasnih dijagrama, kao i neke od njihovih funkcionalnosti:

Funkcionalnost	Desktop klijent	Internet verzija aplikacije	Mobilna verzija aplikacije	Mogućnost exporta u .png formatu	Mogućnost generisanja koda	Kolaborati vnost	Intuitivan	Sve opcije su besplatne
Ime								
ClassyCraft	✓	✗	✗	✓	✓	✗	✓	✓
Gleek.io	✗	✓	✗	✓	✗	✗	✗	✗
Visual paradigm	✓	✓	✗	✓	✓	✓	✓	✗
Creately	✓	✓	✓	✓	✗	✓	✓	✗
Smartdraw	✓	✓	✗	✓	✗	✓	✓	✗
Astah	✓	✗	✗	✓	✓	✗	✓	✗

Korisnici

Za korišćenje softvera potrebno je osnovno razumevanje koncepta objektno orijentisanog programiranja i prethodno iskustvo sa klasnim dijagramima.

Za uspešno korišćenje programa potreban je samo jedan korisnik, koji pomoću aplikacije ClassyCraft može napraviti željeni klasni dijagram sa opcijom dodavanja klase, interfejsa, atributa i metoda po želji. Korisnik sam bira koju će vezu izabrati da bi povezo klase i interfejse na klasnom dijagramu.

Interesna grupa ClassyCraft programa jesu programeri.

Zahtevi

U nastavku je lista funkcija koju ClassyCraft treba da ispuni. Radi preglednijeg opisa biće uvedena tri nivoa prioriteta:

- **Visok** (program ne može funkcionisati bez zadate funkcionalnosti)
- **Srednji** (poželjno je da program ispunjava zadatu funkcionalnost)
- **Nizak** (program može u potpunosti da radi bez zadate funkcionalnosti)

1.Osnovne operacije nad elementima

1.1. Pravljenje elemenata

Element podrazumeva bilo koju vrstu klase (na primer enum, apstraktna, final), interfejsa ili veze. Korisnik treba da bude u mogućnosti da bira koji će element dodati, kako bi po sopstvenim potrebama napravio klasni dijagram.

- **Prioritet:** visok
- **Rizik:** srednji, korisnik se jedino može zbuniti prilikom odabira elementa za dodavanje
- **Zainteresovana strana:** korisnik

1.2. Modifikovanje elemenata klasnog dijagrama

Postojeći elementi treba da budu takvi da korisnik može da im menja izgled i sadržaj.

- **Prioritet:** srednji
- **Rizik:** nizak
- **Zainteresovana strana:** korisnik

1.3. Brisanje elemenata

Elementi koje je korisnik dodao moraju da budu u stanju da se obrišu sa radne površine ukoliko korisnik primeti da je element suvišan.

- **Prioritet:** visok
- **Rizik:** srednji, zbog slučaja da korisnik slučajno obriše pogrešan element
- **Zainteresovana strana:** korisnik

1.4. Vizuelizacija klasnog dijagrama

Klase i interfejsi treba da budu grafički prikazani na radnoj površini u vidu pravougaonih polja, kako bi korisnik imao preglednost šta je do tada napravio.

- **Prioritet:** visok
- **Rizik:** nizak.
- **Zainteresovana strana:** korisnik

1.5. Dodavanje atributa i metoda

Korisnik ima opciju da dodaje attribute i metode u klasu ili metodu u interfejs koji je prethodno napravio.

- **Prioritet:** visok
- **Rizik:** srednji, postoji rizik da korisnik doda atribut ili metodu u pogrešnu klasu i takođe postoji rizik da korisnik pogrešno nazove atribut ili metodu, ili zaboravi da doda parametar u metodi.
- **Zainteresovana strana:** korisnik

1.6. Dodavanje i brisanje opisa veze

Korisnik je u mogućnosti da doda opis na početku ili kraju veze (na primer 0..*), ili da ga obriše.

- **Prioritet:** srednji
- **Rizik:** nizak, eventualno postoji ukoliko korisnik slučajno obriše vezu ili klasu / interfejs
- **Zainteresovana strana:** korisnik

2. Operacije nad fajlovima

2.1. Pravljenje novog projekta

Korisnik treba da bude u mogućnosti da napravi novi projekat koji može da sadrži više klasnih dijagrama.

- **Prioritet:** visok
- **Rizik:** nizak
- **Zainteresovana strana:** korisnik

2.2. Pravljenje novog klasnog dijagrama

Korisnik može da napravi novu radnu površinu na kojoj će da pravi novi klasni dijagram.

- **Prioritet:** visok
- **Rizik:** nizak
- **Zainteresovana strana:** korisnik

2.3. Čuvanje i otvaranje projekta sa klasnim dijagramom

Korisnik treba da bude u mogućnosti da na određenom mestu na računaru, u nekom obliku, sačuva klasni dijagram, tj. projekat koji je napravio, da bi kasnije bio u mogućnosti da taj klasni dijagram može ponovo da otvori i nastavi sa radom.

- **Prioritet:** srednji
- **Rizik:** srednji, postoji mogućnost da klasni dijagram koji je korisnik sačuvao bude oštećen usled lošeg cuvanja klasnog dijagrama.
- **Zainteresovana strana:** korisnik

2.4. Export dijagrama u .png formatu

Korisnik bi trebao da bude u mogućnosti da nakon završetka izrade klasnog dijagrama, taj isti dijagram exportuje u .png formatu.

- **Prioritet:** srednji
- **Rizik:** nizak
- **Zainteresovana strana:** korisnik, šef

2.5. Automatsko generisanje kôda

Da bi korisniku ubrzao rad, aplikacija treba da bude u mogućnosti da nakon što korisnik klikne opciju za generisanje kôda, generiše fajl u kome će se nalaziti kod sa svim klasama, i eventualno atributa i metoda u njima.

- **Prioritet:** nizak
- **Rizik:** srednji, postoji mogućnost da je korisnik pogrešno uneo neki atribut ili parametar metode, što će uzrokovati da kôd u novonastalom fajlu ne bude funkcionalan.
- **Zainteresovana strana:** korisnik

3. Organizacija strukture podataka projekta

3.1. Implementacija strukture podataka u vidu stabla

Komponente klasnog dijagrama treba da budu raspoređene u strukturu stabla kako bi se ograničilo dodavanje neželjenih elemenata na klasni dijagram. Svaka struktura ima svog roditelja u koji može da bude dodata ili dete koje može biti dodato u nju.

- **Prioritet:** srednji
- **Rizik:** nizak
- **Zainteresovana strana:** šef

3.2. Prikaz stabla struktura podataka

Korisnik prilikom rada na klasnom dijagramu, treba da bude u mogućnosti da vidi sve elemente svog dijagrama, raspoređene u stablo sa leve strane.

- **Prioritet:** nizak
- **Rizik:** nizak
- **Zainteresovana strana:** korisnik

4. Pomoć i podrška korisnicima

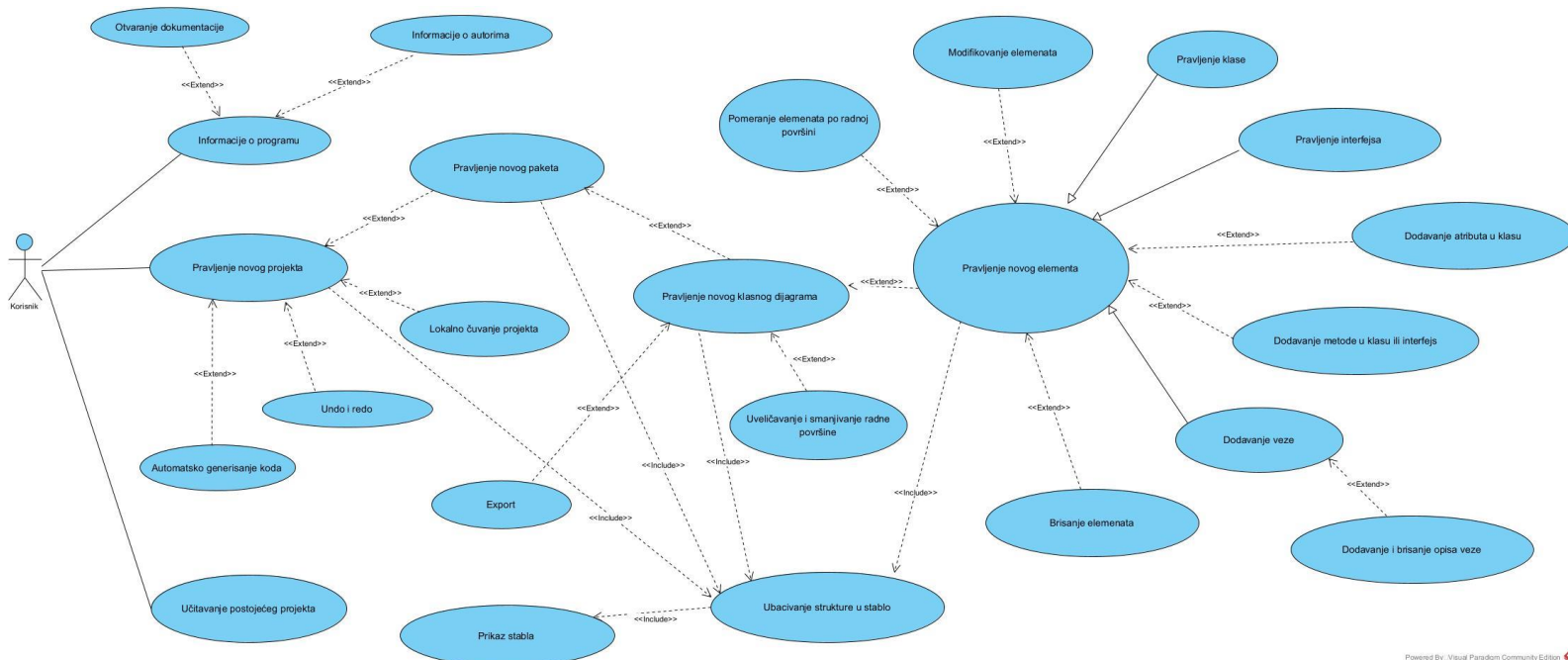
4.1. Vodič kroz aplikaciju

Da bi se olakšalo novim korisnicima da se snađu i brzo savladaju osnovne funkcionalnosti aplikacije, treba postojati interaktivni vodič koji će korisnicima pokazati kako da koriste osnovne funkcije aplikacije.

- **Prioritet:** nizak
- **Rizik:** nizak, ali može biti izazovno osigurati da je vodič razumljiv i efikasan za sve korisnike
- **Zainteresovana strana:** novi korisnici

Funkcionalni dijagram

Ispod se nalazi slika UseCase dijagrama projekta ClassyCraft



UseCase

Pravljenje novog projekta

- **Akter:** korisnik
- **Preduslov:** prethodno otvoren program ClassyCraft
- **Opis:** Korisnik pravi novi projekat koji može da sadrži više klasnih dijagrama
- **Izuzetak:** /
- **Koraci:** 1) korisnik bira opciju novi projekat 2) korisnik imenuje projekat 3) korisnik pravi projekat
- **Tačke proširenja:** pravljenje novog paketa, lokalno čuvanje projekta, undo i redo, automatsko generisanje koda
- **Stanje nakon završetka:** Korisnik je napravio novi projekat, prikazuje se ime na aplikaciji

Pravljenje novog paketa

- **Akter:** korisnik
- **Preduslov:** Prethodno napravljen projekat u okviru kog će se napraviti paket
- **Opis:** Korisnik pravi novi paket u okviru projekta.
- **Izuzetak:** /
- **Koraci:** 1) korisnik bira opciju novi paket 2) korisnik imenuje paket 3) korisnik pravi paket
- **Tačke proširenja:** pravljenje novog klasnog dijagrama
- **Stanje nakon završetka:** korisnik je napravio novi paket

Pravljenje novog klasnog dijagrama

- **Akter:** korisnik
- **Preduslov:** Prethodno napravljen projekat i paket u okviru kog će se napraviti dijagram
- **Opis:** Korisnik pravi novu radnu površinu.
- **Izuzetak:** Korisnik nije napravio paket a pokušava da napravi klasni dijagram
- **Koraci:** 1) korisnik bira opciju novi klasni dijagram 2) korisnik imenuje klasni dijagram 3) korisnik pravi novi klasni dijagram
- **Tačke proširenja:** pravljenje novog elementa, uveličavanje i smanjivanje radne površine, export
- **Stanje nakon završetka:** Korisnik je napravio radnu površinu na koju će biti u mogućnosti da dodaje klase i interfejs.

Pravljenje novog elementa

- **Akter:** korisnik
- **Preduslov:** korisnik je napravio klasni dijagram
- **Opis:** korisnik dodaje novi element (klasa, interfejs ili neki tip veze)
- **Izuzetak:** /

- **Koraci:** 1) korisnik bira element koji dodaje 2) korisniku izlazi prozor sa podacima koje treba da popuni da bi napravio novi element 3) korisnik je napravio novi element
- **Tačke proširenja:** modifikovanje elemenata, brisanje elemenata, pomeranje elemenata po radnoj površini, dodavanje atributa u klasu, dodavanje metode u klasu ili interfejs
- **Stanje nakon završetka:** korisnik je napravio nov element, i dodao ga na radnu površinu

Pravljenje klase

- **Akter:** korisnik
- **Preduslov:** korisnik je napravio klasni dijagram
- **Opis:** korisnik pravi željeni tip klase i dodaje ga na radnu površinu
- **Izuzetak:** korisnik nije odabrao tip klase
- **Koraci:** 1) korisnik bira opciju za pravljenje nove klase 2) unosi podatke o klasi 3) dodaje se klasa na radnu površinu
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je dodao klasu na radnu površinu

Pravljenje interfejsa

- **Akter:** korisnik
- **Preduslov:** korisnik je napravio klasni dijagram
- **Opis:** korisnik pravi interfejs
- **Izuzetak:** /
- **Koraci:** 1) korisnik bira opciju za pravljenje interfejsa 2) dodaje mu se interfejs na radnu površinu
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je napravio interfejs i dodao ga na radnu površinu

Dodavanje atributa u klasu

- **Akter:** korisnik
- **Preduslov:** prethodno napravljena klasa u koju će se smestiti atributi
- **Opis:** korisnik dodaje attribute sa određenim karakteristikama (vidljivost, static, final) u klasu
- **Izuzetak:** korisnik izabere interfejs umesto klase ili ne odabere neku karakteristiku atributa
- **Koraci:** 1) korisnik bira klasu 2) korisnik unosi naziv atributa 3) korisnik popunjava podatke o vidljivosti i ostalim osobinama 4) dodaje se atribut u izabranu klasu
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je dodao atribut u klasu

Dodavanje metode u klasu ili interfejs

- **Akter:** korisnik
- **Preduslov:** prethodno napravljena klasa ili interfejs
- **Opis:** korisnik dodaje metodu sa određenim karakteristikama (vidljivost, static, final) u klasu ili interfejs
- **Izuzetak:** korisnik ne odabere neku karakteristiku metode
- **Koraci:** 1) korisnik bira klasu ili interfejs 2) korisnik unosi naziv metode 3) korisnik popunjava podatke o vidljivosti i ostalim osobinama 4) dodaje se metoda u polje
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je dodao metodu u klasu ili interfejs

Dodavanje veze

- **Akter:** korisnik
- **Preduslov:** korisnik je dodao dve klase, dva interfejsa, ili po jednu klasu i interfejs koju može povezati. Izuzetan je slučaj kada korisnik pravi listu objekata iste klase, pa može dodati vezu koja pokazuje na istu klasu.
- **Opis:** korisnik dodaje jedan od tipova veze
- **Izuzetak:** korisnik ne odabere tip veze, ili na radnoj površini ne postoji nijedna klasa ni interfejs
- **Koraci:** 1) korisnik bira opciju za dodavanje veze 2) korisnik bira tip veze 3) korisnik bira koja će polja povezati 4) polja su povezana
- **Tačke proširenja:** dodavanje i brisanje opisa veze
- **Stanje nakon završetka:** korisnik je povezao dve klase, dva interfejsa ili interfejs i klasu. Izuzetan je slučaj kad se dodaje veza na jednu istu klasu.

Dodavanje i brisanje opisa veze

- **Akter:** korisnik
- **Preduslov:** prethodno napravljena veza
- **Opis:** korisnik dodaje ili briše opis kardinalosti veze
- **Izuzetak:** /
- **Koraci:** 1) korisnik bira vezu na koju će dodati opis 2) korisnik piše opis 3) dodaje se opis na vezu
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je dodao ili obrisao opis kardinalnosti veze

Pomeranje elemenata po radnoj površini

- **Akter:** korisnik
- **Preduslov:** Prethodno napravljena klasa ili interfejs
- **Opis:** Korisnik može da pomeri klasu ili interfejs na željeno mesto
- **Izuzetak:** Korisnik može pokušavati da pomeri element van radne površine

- **Koraci:** 1) korisnik bira element koji želi da pomeri 2) korisnik pomera element mišem 3) element je pomeren
- **Tačke proširenja:** /
- **Stanje nakon završetka:** Željeni element je pomeren

Modifikovanje elemenata

- **Akter:** korisnik
- **Preduslov:** postoji element koji može da se modifikuje
- **Opis:** korisnik može da izmeni dodati element
- **Izuzetak:** korisnik može slučajno da izmeni nešto što nije nameravao
- **Koraci:** 1) korisnik bira element koji želi da modifikuje 2) korisnik modifikuje element
- **Tačke proširenja:** /
- **Stanje nakon završetka:** element koji je odabrao korisnik je modifikovan

Brisanje elemenata

- **Akter:** korisnik
- **Preduslov:** prethodno napravljen element
- **Opis:** korisnik uklanja željeni element sa radne površine
- **Izuzetak:** korisnik može ukloniti element koji nije želeo
- **Koraci:** 1) korisnik bira opciju za brisanje 2) korisnik briše element sa radne površine
- **Tačke proširenja:** /
- **Stanje nakon završetka:** željeni element je uklonjen sa radne površine

Uveličavanje i smanjenje radne površine

- **Akter:** korisnik
- **Preduslov:** prethodno napravljena radna površina
- **Opis:** Korisnik može da uveliča ili smanji radnu površinu zbog boljeg pregleda atributa i metoda
- **Izuzetak:** /
- **Koraci:** 1) korisnik mišem uveličava ili smanjuje radnu površinu
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je uveličao ili smanjio radnu površinu

Undo i redo

- **Akter:** korisnik
- **Preduslov:** U slučaju undo-a, prethodno mora biti napravljen bar jedan element, dok u slučaju redo-a, korisnik makar jednom mora kliknuti opciju undo
- **Opis:** korisnik se vraća jedan korak unazad ili je otišao korak unapred
- **Izuzetak:** /
- **Koraci:** 1) korisnik bira opciju undo ili redo 2) korisnik se vratio korak unazad ili korak unapred

- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je otišao korak unazad ili korak unapred

Export

- **Akter:** korisnik
- **Preduslov:** prethodno je napravljen klasni dijagram
- **Opis:** korisnik može da exportuje klasni dijagram u .png formatu
- **Izuzetak:** korisnik nije napravio klasni dijagram
- **Koraci:** 1) korisnik bira opciju export 2) korisnik bira koji klasni dijagram želi da exportuje 3) korisniku se na željenoj lokaciji dodaje .png format klasnog dijagrama
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je uspešno exportovao fajl u .png formatu

Lokalno čuvanje projekta

- **Akter:** korisnik
- **Preduslov:** korisnik je prethodno napravio projekat
- **Opis:** korisnik lokalno čuva projekat u računar, da bi mogao kasnije da nastavi sa radom na istom
- **Izuzetak:** korisnik nije napravio projekat
- **Koraci:** 1) korisnik bira opciju sačuvaj 2) korisnik bira koji projekat želi da sačuva 3) korisnik je sačuvao projekat
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik je uspešno sačuvao fajl na zadatoj lokaciji na računaru

Učitavanje postojećeg projekta

- **Akter:** korisnik
- **Preduslov:** prethodno sačuvan projekat na računaru
- **Opis:** korisnik može da učitaj projekat koji je ranije sačuvao na računaru
- **Izuzetak:** korisnik pokušava da otvori fajl koji nije napravljen u ClassyCraft-u
- **Koraci:** 1) korisnik bira opciju otvori 2) korisnik bira projekat koji želi da otvori 3) korisnik je učitao projekat
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik otvara projekat na kome je ranije radio

Automatsko generisanje koda

- **Akter:** korisnik
- **Preduslov:** prethodno tačno dodati svi atributi i metode u element bez grešaka
- **Opis:** korisnik može da automatski generiše kod na osnovu elemenata koje je napravio na radnoj površini. Kod će se nalaziti u fajlu sa kojeg može da kopira kod i ubaci u željeni program
- **Izuzetak:** klasa, interfejs, metoda ili atribut nisu tačno napravljeni
- **Koraci:** 1) korisnik bira opciju generiši kod 2) korisniku se stvara fajl sa kodom
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik dobija fajl sa generisanim kodom.

Informacije o programu

- **Akter:** korisnik, profesor
- **Preduslov:** prethodno otvoren program ClassyCraft
- **Opis:** akter dolazi do svih informacija o programu uz pomoć menija
- **Izuzetak:** /
- **Koraci:** 1) akter na meniju pronalazi padajući meni o informacijama
- **Tačke proširenja:** informacije o autorima, otvaranje dokumentacije
- **Stanje nakon završetka:** prikazuju se informacije o programu ClassyCraft

Informacije o autorima

- **Akter:** korisnik, profesor
- **Preduslov:** prethodno otvoren program ClassyCraft i padajući meni sa informacijama o programu
- **Opis:** akter može da otvori prozor da bi saznao ko je učestvovao u pravljenju programa
- **Izuzetak:** /
- **Koraci:** 1) akter otvara padajući meni o informacijama 2) akter bira opciju informacije o autorima 3) akteru se otvara prozor sa informacijama o autorima
- **Tačke proširenja:** /
- **Stanje nakon završetka:** otvoren je prozor sa informacijama o autorima

Otvaranje dokumentacije

- **Akter:** korisnik, profesor
- **Preduslov:** prethodno otvoren program ClassyCraft i padajući meni sa informacijama o programu
- **Opis:** akter otvara dokumentaciju o programu ClassyCraft
- **Izuzetak:** /
- **Koraci:** 1) akter otvara padajući meni o informacijama 2) akter bira opciju otvaranja dokumentacije 3) akteru se otvara prozor sa dokumentacijom
- **Tačke proširenja:** /
- **Stanje nakon završetka:** akter je otvorio dokumentaciju o programu ClassyCraft

Ubacivanje strukture u stablo

- **Akter:** korisnik
- **Preduslov:** prethodno napravljen klasni dijagram
- **Opis:** svaki novi element koji korisnik napravi, ubacuje se u stablo podataka
- **Izuzetak:** /
- **Koraci:** /
- **Tačke proširenja:** /
- **Stanje nakon završetka:** nakon pravljenja bilo kog elementa, element je dodat u stablo podataka

Prikaz stabla

- **Akter:** korisnik
- **Preduslov:** prethodno napravljen klasni dijagram
- **Opis:** prikazuje se stablo na delu ekrana namenjeno za prikaz stabla
- **Izuzetak:** /
- **Koraci:** /
- **Tačke proširenja:** /
- **Stanje nakon završetka:** korisnik može videti stablo sa leve strane ekrana