

ClassyCraft

Uvod

ClassyCraft je program čija je glavna uloga vizualno prikazivanje i modelovanje klasnih dijagrama. Program preko klasnog dijagrama, UML-a, ima za namenu da vizualizovanjem olakša programerima konstruisanje i dokumentaciju projekta. Klasni dijagram predstavlja dijagram strukture, jer pored klasa prikazuje i vrstu veze, to jest odnos u kojim se one nalaze.

Rečnik

UML - (Unified Modeling Language) graficki orijentisani jezik koji vizuelno dokumentuje strukturu softvera

Klasa – model (šablon, obrazac) za objekat

Metoda – funkcija koja se poziva nad objektom neke klase

Konstruktor – operacija klase koja služi za kreiranje objekata klase, može biti bez argumenata, a može imati argumente

Interface – opisuje šta klase treba da rade, ali ne i kako

Dependency - veza zavisnosti

Konkurencija

U tabeli su prikazani konkurenti ClassyCraft aplikacije

	Besplatan za korišćenje	Desktop verzija	Mobilna verzija	Online verzija	Potreban nalog za korišćenje	Kreiranje dijagrama samo kucanjem koda
ClassyCraft	✓	✓	✗	✗	✗	✗
Visual Paradigm	✓	✓	✗	✓	✓	✗
Lucidchart	✗	✗	✓	✓	✓	✗
Gleek.io	✗	✗	✗	✓	✓	✓
Draw.io	✓	✓	✓	✓	✗	✗
Microsoft Visio	✗	✓	✓	✓	✓	✗

Korisnici

ClassyCraft je namenjen za rad na računaru i njegovi korisnici su programeri i studenti programiranja. Program se koristi radi dokumentovanja arhitekture softvera.

Zahtevi

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

- nizak(program može u potpunosti da radi bez zadate funkcionalnosti)
- srednji(poželjno je da program ispunjava zadata funkcionalnost)
- visok(program ne može da funkcioniše bez zadate funkcionalnosti)

1. Osnovne operacije nad poljima koje sadrži UML dijagram:

1.1 Kreiranje i brisanje klase

Mogućnost korisnika da kreira ili obriše već postojeću klasu. Prioritet je visok, dok je rizik nizak pri kreiranju klase, kao i pri brisanju ukoliko projekat nije u međuvremenu sačuvan. U suprotnom slučaju, opcija vraćanja koraka u nazad nije moguća i rizik pri slučajnom brisanja je visok.

1.2 Vizuelno prikazivanje dijagrama

Svi pojmovi i veze koje je korisnik napravio treba da budu grafički prikazani na radnoj površini, kako bi korisnik imao uvid u svoj rad. Prioritet je visok, dok je rizik nizak.

1.3 Dodavanje i brisanje atributa (objekata) klase

Korisnik treba da ima mogućnost da kreira atribut i doda ga željenoj klasi, kao i mogućnost brisanja već postojećeg atributa neke klase. Prioritet je visok, dok je rizik nizak pri dodavanju atributa, kao i pri brisanju ukoliko projekat nije u međuvremenu sačuvan. U suprotnom slučaju, opcija vraćanja koraka u nazad nije moguća i rizik pri slučajnom brisanja je visok.

1.4 Dodavanje i brisanje metoda (operacija) klase

Potrebno je da korisnik bude u mogućnosti da kreira operaciju i dodeli je klasi, kao i mogućnost brisanja već postojeće operacije neke klase. Prioritet je visok, dok je rizik nizak pri dodavanju metode, kao i pri brisanju ukoliko projekat nije u međuvremenu sačuvan. U suprotnom slučaju, opcija vraćanja koraka u nazad nije moguća i rizik pri slučajnom brisanja je visok.

1.5 Dodavanje i brisanje interfejsa

Korisnik treba da bude u mogućnosti da kreira ili obriše klasu interfejs kojoj ne može da dodeli objekte, već samo metode. Prioritet je visok, a sam rizik je nizak pri kreiranju, kao i pri brisanju ukoliko projekat nije u međuvremenu sačuvan. U suprotnom slučaju, opcija vraćanja koraka u nazad nije moguća i rizik pri slučajnom brisanja je visok.

1.6 Modifikovanje polja

Za već kreirane klase, interfejsse, kao i njihove atribute i metode, korisnik bi trebao da ima stalni pristup modifikovanju, to jest menjanju sadržaja. Prioritet je srednji, dok je rizik nizak.

1.7 Raspoređivanje polja po radnoj površini

Korisnik treba da ima mogućnost da pomera polja UML klasnog dijagrama radi preglednosti. Prioritet je srednji, a rizik je nizak.

2. Osnovne operacije nad vezama između polja UML klasnog dijagrama:

2.1 Kreiranje i brisanje veza između polja

Kako bi UML dijagram bio potpun, potrebno je da polja budu povezana određenim relacijama. Veza može biti jedna od četiri tipa: asocijacija, kompozicija, agregacija i dependency. Takođe, korisnik bi trebao da ima mogućnost da obriše željenu relaciju. Prioritet je visok, a rizik pri kreiranju i dodavanju je nizak, kao i pri brisanju ukoliko projekat nije u međuvremenu sačuvan. U suprotnom slučaju, opcija vraćanja koraka u nazad nije moguća i rizik pri slučajnom brisanja je visok.

2.2 Modifikovanje veza

Potrebno je da korisnik ima pristup izmenama veza između polja. Prioritet je visok, dok je rizik nizak.

2.3 Prepovezivanje veza koje su već napravljene

Kreirane veze korisnik treba da bude u mogućnosti da prepoveže sa jednog polja na drugo. Prioritet je nizak, kao i sam rizik.

3. Operacije nad celokupnim UML klasnim dijagramom:

3.1 Opcija koja vraća poslednju promenu

Potrebno je dodati opciju koja vraća poslednju izvršenu promenu nad projektom, koja bi olakšala korisniku rad ukoliko dođe do slučajnog brisanja nekog polja ili veze. Ukoliko je prvobitno projekat sačuvan, opcija vraćanja poslednjeg koraka neće moći da se izvrši. Prioritet je visok, dok je rizik nizak.

3.2 Opcija da se približi ili udalji UML dijagram na točkić miša

Korisniku je potrebno omogućiti da uz pomoć točkića miša može da približi ili udalji UML klasni dijagram. Prioritet je nizak, kao i sam rizik.

4. Operacije nad celokupnim projektom:

4.1 Kreiranje projekta

Potrebno je dodati opciju koja omogućava korisniku da kreira novi projekat. Prioritet je visok, a sam rizik je nizak.

4.2 Brisanje projekta

Pored postojeće opcije za kreiranje projekta, korisniku treba da se omogući i brisanje željenog projekta. Prioritet je visok, dok je rizik samo pri brisanju ako se slučajno obriše ceo projekat.

4.3 Kreiranje i brisanje paketa

Korisniku treba da bude omogućeno kreiranje, kao i samo brisanje paketa. Prioritet je visok, a rizik pri kreiranju je nizak, kao i pri brisanju zbog opcije koja vraća poslednju promenu.

4.4 Čuvanje projekta

Projekat koji je korisnik kreirao zajedno sa svim promenama, bi trebao da ima opciju čuvanja negde na računaru u odredjenom obliku, radi ponovnog pristupa. Prioritet je srednji, dok rizik postoji u slučaju lošeg čuvanja projekta.

4.5 Učitavanje postojećeg projekta

Ukoliko je projekat prethodno pravilno sačuvan, korisnik treba da bude u mogućnosti da ponovo pristupi projektu i vrši željene izmene na njemu. Prioritet je srednji, a rizik je nizak.

4.6 Mogućnost prikazivanja više projekata u različitim prozorima

Potrebno je da korisnik bude u mogućnosti da otvori više projekata od jednom u odvojenim prozorima. Prioritet je nizak, kao i sam rizik.

4.7 Mogućnost da se UML dijagram eksportuje

Za UML klasni dijagram koji korisnik napravi treba da postoji opcija za eksportovanje istog. Prioritet je nizak, kao i sam rizik.

4.8 Mogućnost da se UML dijagram prikaže

Korisnik prilikom rada u programu ima mogućnost da

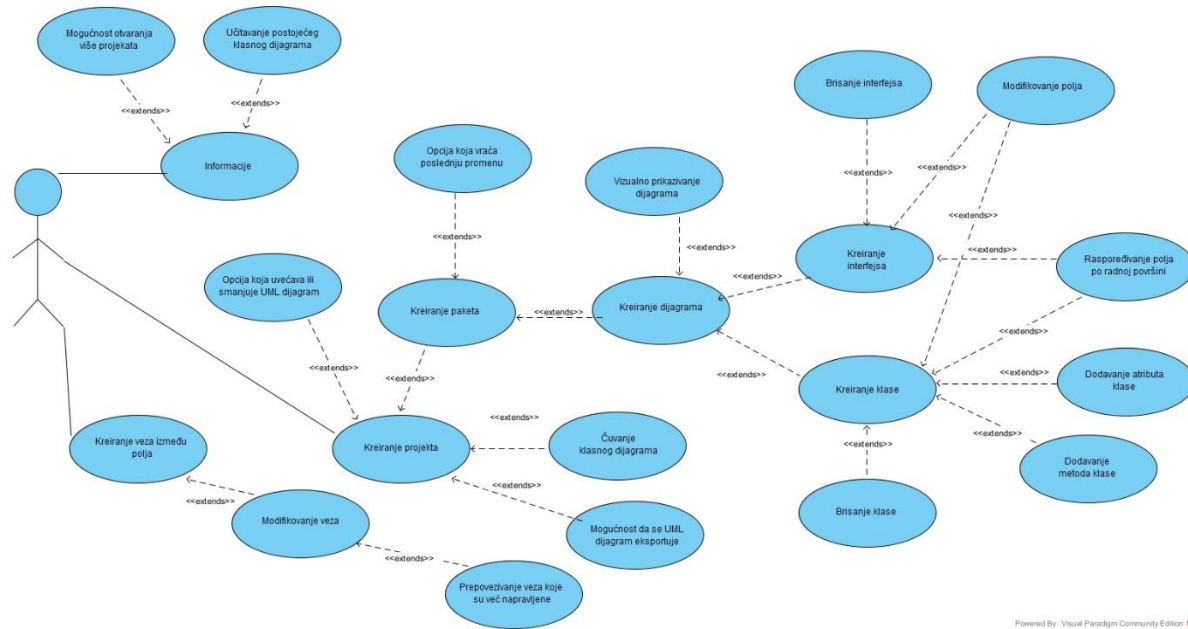
Korisnik prilikom rada na mapi uma treba da bude u mogućnosti da vidi sve elemente svoje mape raspoređene u stablo sa leve

strane. Prioritet je srednji, rizik je nizak dok je zainteresovana strana korisnik.

4.9 Kreiranje UML dijagrama

Korisnik ima mogućnost da kreira dijagram unutar projekta. Prioritet je visok dok je rizik nizak.

Funkcionalan dijagram



UseCase

Kreiranje projekta

Učesnik: Korisnik

Preduslov: Program ClassyCraFt je potrebno da bude otvoren

Opis: Korisnik pomoću opcije iz menija provi novi projekat

Izuzetak:Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Kreira se projekat i prikazuje se njegovo ime u prozoru

Opcija koja uvećava ili smanjuje UML dijagram

Učesnik: Korisnik

Preduslov: Mora da bude kreiran projekat kako bi postojala radna površina koju može približiti i odaljiti

Opis: Korisnik može da približi i odalji radnu površinu upotrebom točkića na mišu

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Stanje ostaje isto kao i posle izvršavanja ove funkcije

Čuvanje klasnog dijagrama

Učesnik: Korisnik

Preduslov: Korisnik mora da kreira projekat kako bi mogao da ga sačuva

Opis: Čuvanje projekta

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Stanje ostaje isto kao i pre izvršavanja ove funkcije

Mogućnost da se UML dijagram eksportuje

Učesnik: Korisnik

Preduslov: Korisnik mora da kreira projekat

Opis: Dijagram se eksportuje

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Stanje ostaje isto kao i pre izvršavanja ove funkcije

Kreiranje paketa

Učesnik: Korisnik

Preduslov: Korisnik mora da kreira paket

Opis: Korisnik može da kreira paket/podpaket u koji će kreirati klase/interfejse

Izuzetak: Korisnik je nazvao paket imenom kojim se već postojeći paket zove, pri čemu će morati da ga drugačije nazove

Stanje: Dodat je paket u projekat

Opcija koja vraća poslednju promenu

Učesnik: Korisnik

Preduslov: Mora da postoji prethodna promena

Opis: Program vraća za jedno stanje unazad

Izuzetak: N postoji prethodna promena pri čemu ova funkcija neće biti moguća

Stanje: Stanje se vraća na prethodno

Kreiranje interfejsa

Učesnik: Korisnik

Preduslov: Mora da bude kreiran projekat

Opis: Korisnik kreira interfejs

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Interfejs je dodat u projekat

Raspoređivanje polja po radnoj površini

Učesnik: Korisnik

Preduslov: Moraju da postoje polja na radnoj površini

Opis: Korisnik može da pomera polja koja je kreirao po radnoj površini

Izuzetak: Na radnoj površini nema polja, samim tim ova funkcija neće biti moguća

Stanje: Pomerena polja na radnoj površini su promenile svoj položaj

Brisanje interfejsa

Učesnik: Korisnik

Preduslov: Mora da postoji interfejs u projektu

Opis: Brisanje interfejsa iz projekta

Izuzetak: U projektu nema interfejsa, samim tim ova funkcija neće biti moguća

Stanje: U projektu se briše interfejs i sve metode koje su se nalazile u njoj

Modifikovanje polja

Učesnik: Korisnik

Preduslov: Polja moraju da postoje u projektu

Opis: Modifikovanje polja

Izuzetak: Na radnoj površini nema polja, samim tim ova funkcija neće biti moguća

Stanje: Polja su promenila vrednost promenjenih elemenata

Kreiranje klasa

Učesnik: Korisnik

Preduslov: Projekat mora da bude kreiran

Opis: Kreira se klasa u projektu

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: U projekat je dodata klasa

Dodavanje atributa klase

Učesnik: Korisnik

Preduslov: Mora da postoji kreirana klasa u projektu

Opis: Dodavanje atributa klase u klasu

Izuzetak: U projektu ne postoji klasa, pa samim tim ova funkcija neće biti moguća

Stanje: Klasi su dodati željeni atributi

Dodavanje metoda klase

Učesnik: Korisnik

Preduslov: Mora da postoji kreirana klasa u projektu

Opis: Dodavanje metoda klase u klasu

Izuzetak: U projektu ne postoji klasa, pa samim tim ova funkcija neće biti moguća

Stanje: Klasi su dodate željene metode

Brisanje klase

Učesnik: Korisnik

Preduslov: Klasa mora da postoji u projektu

Opis: Brisanje Klase iz projekta

Izuzetak: Klasa ne postoji u projektu, samim tim ova opcija neće biti moguća

Stanje: Iz projekta se briše klasa i svi atributi i metode koje su se nalazile u njoj

Vizualno prikazivanje

Učesnik: Korisnik

Preduslov: Korisnik treba da kreira dijagram i na njemu polja

Opis: Grafički prikazivani pojmovi na radnoj površini

Izuzetak: Korisnik nema kreiranih polja, čime ova funkcija neće biti moguća

Stanje: Na radnoj površini se prikazuju pojmovi

Kreiranje veza između polja

Učesnik: Korisnik

Preduslov: Korisnik treba da kreira minimum dva polja

Opis: Povezivanje polja jednom od veza

Izuzetak: Korisnik nema barem dva polja, čime neće moći da poveže drugi kraj veze i kreira veza

Stanje: Polja su povezana

Modifikovanje veza

Učesnik: Korisnik

Preduslov: Korisnik mora da kreira veze

Opis: Korisnik može da menja vrstu veze kao i da je pomera na radnoj površini

Izuzetak: U projektu ne postoje veze, čime ova funkcija neće biti moguća

Stanje: Veze u projektu menjaju svoju vrstu ili poziciju na radnoj površini

Prepovezivanje veza koje su već napravljene

Učesnik: Korisnik

Preduslov: Mora da postoje veze i klase

Opis: Menjanje jedne od povezanih klasa drugom

Izuzetak: Nema treće klase koju ćemo da prepovežemo

Stanje: Klasa menja klasu sa kojom je povezana

Informacije

Učesnik: Korisnik

Preduslov: Korisnik mora da se nalazi u programu

Opis: Korisnik ima pristup svim projektima i njihovim informacijama

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Program prikazuje listu projekata i njihove informacije

Učitavanje postojećeg klasnog dijagrama

Učesnik: Korisnik

Preduslov: Mora da postoji već sačuvan projekat

Opis: Korisnik može da izabere već sačuvan projekat i otvori ga

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Program otvara izabrani projekat i učitava ga

Mogućnost otvaranja više projekata

Učesnik: Korisnik

Preduslov: Program mora da bude otvoren

Opis: Korisnik može da otvori više projekata u novim prozorima

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Program učitava novi projekat u novom prozoru

Kreiranje UML dijagrama

Učesnik: Korisnik

Preduslov: Korisnik mora da kreira projekat i paket unutar njega

Opis: Korisnik može da kreira dijagram na kojem će da kreira ostala polja

Izuzetak: Korisnik ne može napraviti nikakvu neočekivanu grešku u ovom koraku

Stanje: Program kreira novi dijagram