Programiranje u Javi1

Projektni zadatak

Ishodi učenja							UKUPNO
I1	I2	I3	I4	I5	I6	I7	
15	15	15	15	15	15	10	100

INSTRUKCIJE

- Obrana projekta održava se tijekom ispitnih rokova
- Student prijavljuje kao i pismeni ispit

ISHODI UČENJA:

I1 (15 bodova):

- *minimalni (10 bodova):* Implementirati osnovne koncepte objektno orijentirane paradigme u objektno orijentiranom programskom jeziku na virtualnoj platformi
- *željeni (5 bodova)*: Implementirati napredne koncepte objektno orijentirane paradigme u objektno orijentiranom programskom jeziku na virtualnoj platformi.

I2 (15 bodova):

- *minimalni (10 bodova):* Primijeniti funkcionalnu paradigmu i ocijeniti je u odnosu na objektno orijentiranu paradigmu
- *željeni (5 bodova)*: Odabrati i primijeniti odgovarajuću funkcionalnu ili objektno orijentiranu paradigmu pri implementaciji aplikacije

I3 (15 bodova):

- *minimalni (10 bodova):* Utvrditi potrebu upotrebe okvira kolekcije i tokove te ih implementirati u skladu s najboljim praksama
- *željeni (5 bodova)*: Koristiti napredne implementacije okvira kolekcije i tokove prilikom postavljanja aplikacije

I4 (15 bodova):

- *minimalni (10 bodova):* Usporediti i procijeniti tradicionalni i moderni način rada s podatkovnim sustavom
- *željeni (5 bodova)*: Implementirati moderni način rada s podatkovnim sustavom

I5 (15 bodova):

- *minimalni (10 bodova):* Primijeniti odgovarajuće biblioteke za izradu standardnih grafičkih softverskih rješenja
- *željeni (5 bodova)*: Primijeniti odgovarajuće biblioteke za izradu naprednih grafičkih softverskih rješenja

I6 (15 bodova):

- *minimalni (10 bodova):* Primijeniti odgovarajuće biblioteke temeljene na arhitekturi model-prikaz-upravljač (MVC) u izradi standardnih grafičkih softverskih rješenja
- *željeni (5 bodova)*: Primijeniti odgovarajuće biblioteke temeljene na arhitekturi model-prikaz-upravljač (MVC) u izradi naprednih grafičkih softverskih rješenja

I7 (10 bodova):

- *minimalni (7 bodova):* Usporediti različite pristupe u izradi grafičkih softverskih rješenja
- *željeni (3 boda)*: Koristiti napredne pristupe u izradi grafičkih softverskih rješenja

Napravite aplikaciju koja omogućuje ažuriranje podataka o povezanim entitetima po želji. U nastavku će za entitete biti odabran primjer entiteta **Film** i pripadajućih entiteta **Glumac, Redatelj, Žanr** i slično.

Prilikom izrade aplikacije važno je poštivati najbolje principe objektno orijentirane paradigme i korištenja biblioteka (**Class Library**), prema sljedećim uputama:

- Aplikacija sprema podatke u **Microsoft SQLServer** bazu podataka
- Potrebno je izraditi inicijalizacijsku **DDL** skriptu za kreiranje svih tablica koje će se koristiti u aplikaciji
- Potrebno je izraditi skriptu za brisanje svih podataka iz tablica
- Sav rad s bazom podataka odvija se pozivanjem procedura iz Java koda, korištenjem **Repository** obrasca
- Aplikacija sadrži više tipova korisnika pa je potrebno osigurati 2 uloge - **Administrator** i **Korisnik**
- **Administrator** mora biti barem jedan korisnik s pridruženim parom korisničko ime/lozinka koji će biti kreiran pomoću procedure nakon inicijalizacijske skripte
- Pristup aplikaciji ograničen je na **Login** formu za prijavu, gdje je moguća i jednostavna registracija korisnika uloge **Korisnik**
- **Administrator** postavlja aplikaciju u ispravno stanje za korištenje od strane **Korisnika**
    - **Administrator** prilikom ulaska u aplikaciju ima mogućnost brisanja svih podataka iz baze (čime se brišu i sve slike iz podatkovnog sustava) i uploada novih podataka u bazu, pozivom **RSS parsera** - to predstavlja jednostavno i potpuno administratorsko sučelje
- **RSS parser** je komponenta aplikacije koja analizira sve XML podatke s određenog URL-a i sprema ih u bazu podataka za kasniju upotrebu
    - Slike se moraju preuzeti u lokalni direktorij (primjer: **assets**) i njihove relativne putanje moraju biti spremljene u bazi podataka
    - Primjeri RSS izvora:
        - https://www.nasa.gov/rss-feeds/
        - https://photojournal.jpl.nasa.gov/rss/index.html
        - https://www.cedefop.europa.eu/en/news-and-events/rss-feeds
        - https://news.sky.com/info/rss
        - https://www.sciencedaily.com/newsfeeds.htm

- Nakon ulaska u aplikaciju, **Korisniku** se prikazuje Forma za pregled i promjenu entiteta (**CRUD** operacije), u ovom primjeru **Filma**
- Forme unutar aplikacije moraju biti dobro organizirane (primjer: **JTabbedPane**, gdje je svaki **JPanel** predstavljen vlastitom klasom prema **Single-responsibility principle**)
- Potrebno je kreirati dodatne entitete (primjer: **Glumac, Redatelj**) koji također imaju **CRUD** forme
- Postojeći entitet treba ažurirati na takav način da se može povezati s novim entitetima (primjer: **Film** se može povezati s nekoliko entiteta **Glumac**, **Redatelj, Žanr**)
- Prilikom mijenjanja entiteta ili njegovog brisanja, potrebno je osigurati dosljednu promjenu / brisanje pridruženih slika iz direktorija slika (primjer: **assets**)
- Za prikaz entiteta potrebno je u najvećoj mjeri koristiti **JTable** i pridružene *view-models* poput **AbstractTableModel**
- Za navigaciju u aplikaciji potrebno je koristiti **JMenu**
- Potrebno je implementirati **Drag and drop** funkcionalnost na vlastitom odabranom primjeru u aplikaciji (primjer: dodavanje **Glumca** u **Film**)
- Potrebno je implementirati XML preuzimanje bilo kojeg entiteta pomoću **JAXB** biblioteke

Prilikom razvoja aplikacije važno je poštivati napredne principe objektno orijentirane paradigme (SOC - *separation of concerns*, SRP - *single responsibility principle*, *loose coupling*, *high cohesion*, *prefer composition over inheritance*, DRY - *don't repeat yourself*…)

Bodovi se boduju prema sljedećoj shemi:

I1 (15 bodova):

- (***Minimalni – 10 bodova***)
  - Aplikacija mora pokazati temeljito razumijevanje osnovnih OOP principa – Enkapsulacija, Nasljeđivanje, Polimorfizam i Apstrakcije na primjeru povezanih entiteta koji tvore arhitekturu (u primjeru – Film, Glumac, Redatelj, Žanr, Korisnik, Administrator…)
  - Upravljanje i pohranjivanje podataka mora se osigurati putem sučelja
  - Entiteti podataka u aplikaciji moraju biti ispravno enkapsulirani, implementirati metode *equals()*, *hashcode()* i *toString()*, kao i *Comparable* sučelje, gdje je potrebno
  - Aplikacija mora pokazati temeljito razumijevanje upravljanja iznimkama (***checked*** vs. ***runtime***)
  - *Utilities* u aplikaciji moraju se pružati kao statičke klase, koristeći privatne konstruktore za kontrolu instanciranja
  - Korištenje ispravno enkapsuliranih biblioteka klasa kao Maven modula za **Utilities** i **Dao** se preporučuje
- (***Željeni – 5 bodova***)
  - Aplikacija bi trebala koristiti Singleton (Lazy/Eager) i Repository uzorke
  - Enum bi se trebao koristiti za stvaranje statički tipiziranog RSS parsera
  - Generici bi se trebali koristiti kako bi se spriječila redundantnost

I2 (15 bodova):

- (***Minimalni – 10 bodova***)
  - Aplikacija mora pokazati temeljito razumijevanje funkcionalne paradigme u odnosu na objektno orijentiranu paradigmu – o čemu će se raspravljati na obrani projekta
  - Aplikacija mora pokazati korištenje
    - ***Consumer*** fukcionalnog sučelja
    - ***Predicate*** fukcionalnog sučelja
    - ***Function*** fukcionalnog sučelja
    - ***Filtera***
    - ***Optional*** implentiran kroz funkcionalnu paradigme
    - ***default methods***
- (***Željeni – 5 bodova***)
  - Tijekom obrane pokazati funkcionalnu paradigmu korištenja u odnosu na objektno orijentiranu paradigmu
  - Objasnite prednosti i nedostatke funkcionalne paradigme naspram objektno orijentirane paradigme koristeći konkretne primjere u aplikaciji

I3 (15 bodova):

- (***Minimalni – 10 bodova***)
  - Prijava mora pokazati temeljito razumijevanje *Collections* okvira
  - Aplikacija mora demonstrirati paradigme programiranja prema apstrakcijama, a ne implementacijama
  - Koristite polimorfna rješenja za provjere valjanosti, koristeći sučelja **List** ili **Map**
  - Demonstrirajte korištenje **Set** sučelja kako biste spriječili dupliciranje
  - Tijekom obrane objasnite hijerarhiju *Collections* okvira i identificirajte scenarije za pravilnu upotrebu sljedećih implementacija
    - ***ArrayList*** vs. ***LinkedList***
    - ***HashSet*** vs. ***TreeSet***
    - ***HashMap*** vs. ***TreeMap***
- (***Željeni – 5 bodova***)
  - Aplikacija bi trebala demonstrirati napredne uslužne programe okvira zbirki
  - Koristite ***distinct()***, ***sorted()***, ***skip()***, ***min()***/***max()***, ***findAny()***/***findFirst()***, **anyMatch()**/**noneMatch()** i **Collectors** za demonstraciju naprednog korištenja ***Streams***

I4 (15 bodova):

- (***Minimalni – 10 bodova***)
  - Izradite inicijalizacijsku DDL skriptu za stvaranje svih tablica koje će se koristiti u aplikaciji
  - Izradite skriptu za brisanje svih podataka iz tablica
  - Podaci aplikacije moraju biti pohranjeni u bazi podataka, poštujući najbolja načela
  - Početni podaci moraju se preuzeti iz RSS izvora, korištenjem **XML parsera** i **HttpUrlConnection**, a zatim pohraniti u bazu podataka
  - CRUD se mora implementirati pozivanjem procedura iz Java koda, korištenjem **Repository** uzorka
  - Tijekom obrane projekta usporedite i procijenite korištenje CRUD upita sa i bez procedura
- (***Željeni – 5 bodova***)
  - Potrebno je implementirati ***XML download*** odabranog entiteta pomoću ***JAXB*** biblioteke
  - Preporuča se upotreba ***PreparedStatement***/***CallableStatement*** za prevenciju *SQL injectiona*

I5 (15 bodova):

- (***Minimalni – 10 bodova***)
  - Aplikacija mora pokazati razumijevanje standardnih grafičkih rješenja pravilnom upotrebom **Swing Application Framework** okruženja
  - Pokažite razumijevanje životnog ciklusa komponenti
  - Forme unutar aplikacije moraju biti dobro organizirane
  - Stvorite **Utilities** za enkapsuliranje i pojednostavljenje upotrebe
    - ***JOptionDialog***
    - ***JFileChooser***
  - Koristite ***JMenu*** to za navigaciju u aplikaciji
  - Implementirajte ***Drag and drop*** funkcionalnost u aplikaciji
- (***Željeni – 5 bodova***)
  - Koristite *Maven dependency* za korištenje ***FlatLaf*** biblioteke
  - Kreirajte ***Utility*** za korištenje ikona

I6 (15 bodova):

- (***Minimalni – 10 bodova***)
  - Aplikacija mora demonstrirati duboko razumijevanje ***Swing MVC*** arhitekture skladnim korištenjem **View** entiteta
    - **List**
    - ***Spinner***
    - ***ComboBox***
    - ***ButtonGroup***
  - **Modeli** spomenutih entiteta moraju biti upravljani sa aspekta sigurnosti te dinamički učitani ili enumerirani
  - Moraju se pokazati odgovarajući mehanizmi za rukovanje događajima u smislu komunikacije s **Controller**-om
- (***Željeni – 5 bodova***)
  - Za prikaz entiteta na ekranu, potrebno je koristiti **JTable**
  - Implementirajte **AbstractTableModel** za sofisticirano i dinamično učitavanje i korištenje ***MVC*** obrasca

I7 (10 bodova):

- (***Minimalni – 7 bodova***)
  - GUI aplikacije mora biti implemetirano poštujući SOC principe
    - ***JFrame*** mora biti samo *domaćin* ***JPanel*** instanci
    - ***JTabbedPane*** ili ***SplitPane*** mora biti korišten za organizaciju ***JPanel*** instanci
  - Tijekom prezentacije projekta dubinsko razumijevanje ***JFrame*** i ***JPanel*** životnih ciklusa mora biti demonstrirano
  - Aplikacija mora pokazati korištenje više **JFrame** instanci u ispravnom odnosu roditelj-dijete
  - Usporedba različitih upravljanja **Swing Layoutom** mora se pokazati na obrani projekta
- (***Željeni – 3 boda***)
  - GUI aplikacije ne smije se zamrzavati *tijekom* XML parsiranja i učitavanja podataka u bazu
    - Koristite različite dretve za učitavanje podataka
    - Dubinsko razumijevanje **Main** i **Event Thread management Swing GUI** treba biti demonstrirano tijekom predstavljanja projekta