

Kontroldarbs Nr. 1

B variants

Informācijas Sistēma “Mana bibliotēka”

Izveidot informācijas sistēmu “Mana bibliotēka”, kurā ir iespējams nodrošināt CRUD darbības ar bibliotēkā paņemtajām lietām (grāmatas, e-grāmatas, diski, žurnāli, spēles), lietu rezervēšanu, filtrāciju un statistikas attēlošanu.

Katram studentam tiks piešķirta atšķirīgu lietu kopa, kura jārealizē sistēmā. Variantus skatīt tabulā (dokumenta pielikumā Nr.2). Ar **dzeltenu krāsu** iezīmētajās vietās ir jāievieto Jūsu varianta klase vai mainīgais.

Pēc saviem ieskatiem varat papildināt aprakstā minētās klases, lai nezaudētu sistēmas funkcionalitāti. Jūs drīkstat lietot masīvus vai citas piemērotas datu struktūras tajos gadījumos, kad aprakstā minēts saraksts jeb *ArrayList*.

Par katras klases realizāciju norādīts maksimālais punktu skaits pie ideālas klases realizācijas (norādīts kā parametrs: **max**). Par katru neizpildīto vai nepareizi izpildīto punktu un par katru nerealizēto vai nepareizi realizēto pārbaudi, tiek noņemts punktu skaits no maksimālā (norādīts kā parametrs: **neizpilde**).

Iespējamā UML klašu diagramma atrodama dokumenta pielikumā Nr.1.

Veikt sekojošo klašu/enumeratoru realizāciju:

1. Abstrakta klase *AbstractItem* (**max: 1.4p, neizpilde: a, c, e, f, h - 0.1, b, g: -0.25p, d - 0.4**)
 - a. Mainīgie: `int id`, `String title`, `String itemCode`, `int howManyCopies`;
 - b. Nodrošiniet unikāla id piešķiršanu;
 - c. Get funkcijas visiem uzskaitītajiem mainīgajiem;
 - d. Set funkcijas visiem uzskaitītajiem mainīgajiem (neaizmirstam pievienot pārbaudes set funkcijās, kur tas ir nepieciešams. Piemēram, pārbaudīt min un max robežas skaitliskām vērtībām, referenču vērtībām - vai nav null. Pēc saviem ieskatiem, varat izveidot regex masku `String` vērtībām (nav nepieciešams sarežģīt));
 - e. `setItemCode` funkciju izveidot abstraktu, jo šī funkcija būs jāpārraksta bērnu klasēs. Funkcijai nebūs neviens ieejas parametrs, jo `itemCode` mainīgā vērtība tiks uzstādīta no citiem parametriem.
 - f. Bezargumentu konstruktors;
 - g. Argumentu konstruktors;
 - h. `toString` funkcija.
2. Klase ***Class1 VarX*** - tiek mantota no *AbstractItem* klases (**max: 1.15p, neizpilde: a, b, e, g - 0.1p, c, d, f -0.25p**)
 - a. Mainīgie: ***varX_class1_1*** un ***varX_class1_2***;

- b. Get funkcijas;
- c. Set funkcijas (ar pārbaudēm, piemērus skatīt 2.d. punktā);
- d. Pārrakstīt *setItemCode* funkciju pēc sekojošas shēmas:

`<id>_<šīs klases nosaukums>_<title>`

Piemēram, ja ***Class1 VarX*** ir Disks, tad *itemCode* varētu tikt uzstādīts `1_disks_Batman`. `<šīs klases nosaukums>` nosaukumam varat iestatīt konkrētu String vērtību (piemēram, Disks) vai arī izmantot `getClass().getSimpleName()` funkciju.

- e. Bezargumentu konstruktors;
- f. Argumentu konstruktors;
- g. *toString* funkcija.

3. Klase ***Class2 VarX*** - tiek mantota no *AbstractItem* klases (**max: 1.15p, neizpilde: a, b, e, g - 0.1p, c, d, f - 0.25p**)

- a. Mainīgais: ***varX_class2_1*** un ***varX_class2_2***;
- b. Get funkcijas;
- c. Set funkcijas (ar pārbaudēm, piemērus skatīt 2.d. punktā);
- d. Pārrakstīt *setItemCode* funkciju pēc sekojošas shēmas:

`<id>_<šīs klases nosaukums>_<title>`

Piemēram, ja ***Class2 VarX*** ir Disks, tad *itemCode* varētu tikt uzstādīts `1_disks_Batman`. `<šīs klases nosaukums>` nosaukumam varat iestatīt konkrētu String vērtību (piemēram, Disks) vai arī izmantot `getClass().getSimpleName()` funkciju.

- e. Bezargumentu konstruktors;
- f. Argumentu konstruktors;
- g. *toString* funkcija.

4. Klase *Librarian* (**max: 0.9p, neizpilde: a, b, d, f - 0.1p, c, e - 0.25p**)

- a. Mainīgais: String *personCode*, String *name*, String *surname*, boolean *isAdminLibrarian*;
- b. Get funkcijas;
- c. Set funkcijas (ar pārbaudēm, piemērus skatīt 2.d. punktā);
- d. Bezargumentu konstruktors;
- e. Argumentu konstruktors;
- f. *toString* funkcija.

5. Klase *BorrowingItemsProcedure* (**max: 1.7p, neizpilde: a, b, d, f - 0.1p, e - 0.25p, c, g, h - 0.35p**)

- a. Mainīgie: *Librarian* *librarian*, *loanItems* - saraksts (kurā glabāsies visas lietas, ko bibliotēkārs ir iedevis kādam lietošanā), *LocalDateTime* *borrowingDateTime*, *LocalDateTime* *returningDateTime*;
- b. Get funkcijas;
- c. Set funkcijas (ar pārbaudēm, piemērus skatīt 2.d. punktā); Iekļaut sekojošu funkcionalitāti konkrēto mainīgo uzstādīšanā:
 - i. *borrowingDateTime* - uzstādīt šī brīža datumu un laiku;
 - ii. *returningDateTime* - uzstādīt datumu un laiku, kas ir 2 nedēļas pēc šī brīža laika;
- d. Bezargumentu konstruktors;
- e. Argumentu konstruktors;
- f. *toString* funkcija.

Turpmākās funkcijās būs jāizmanto 6.a punktā izveidotais lietu saraksts, jo tajā brīdī, kad bibliotekārs grob iedot kādam lietotājam lietas no bibliotēkas, tad šīs lietas ir jāizņem no kopējā bibliotēkas lietu saraksta. Noteikti jāpārbauda, vai konkrētā lieta vispār ir atrodamā bibliotēkā un ja ir, vai to skaits ir pietiekošs pēc *howManyCopies* parametra. Ieteiktu šīs divas funkcijas veidot pēc 6.a punkta izpildes.

- g. Funkcija *addItemInBorrowingItemsListByItemCode()*, kas pēc padotā *itemCode* to sameklē bibliotēkā (jeb 6.a.punktā izveidotajā *AbstractItem* tipa sarakstā) un pievieno to *loanItems* sarakstam. Vienkāršības labad - ir iespējams paņemt tikai konkrētas lietas vienu eksemplāru. Šajā darbībā šīs lietas kopskaits jeb *howManyCopies* jāsamazina par 1 (jeb 6.a.punktā izveidotajā *AbstractItem* tipa sarakstā konkrētās lietas daudzums jāsamazina par vienu). Var rasties situācija, ka pēc lietas iedošanas lasītājam, *howManyCopies* paliek 0, bet šajā gadījumā nav nepieciešams izdzēst pašu lietu no bibliotēkas.
- h. Funkcija *showAllClass2 VarX ()* atgriež visas lietas, kas ir ar *Class2 VarX* tipu un atrodas *loanItems* sarakstā.

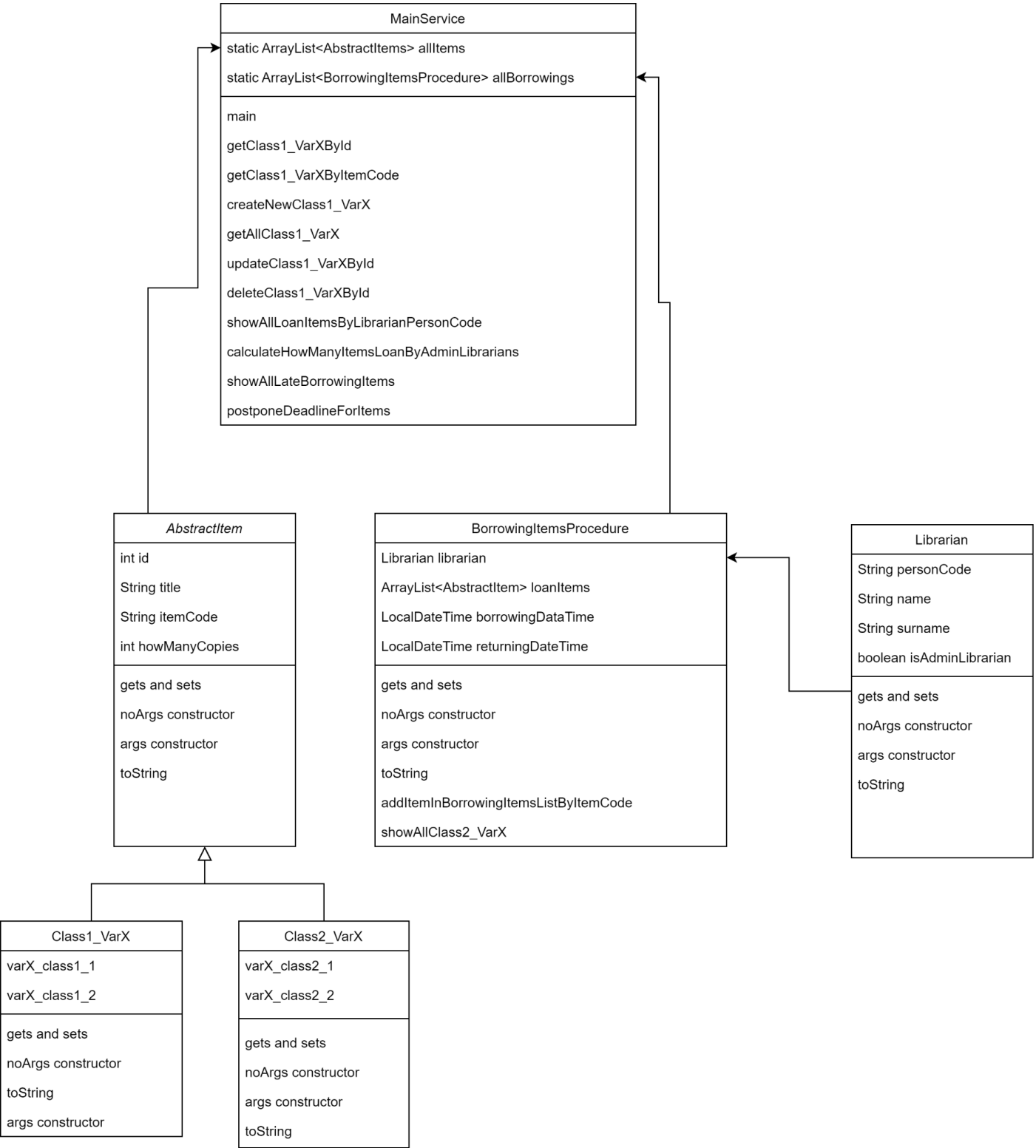
6. Klase MainService (max: 3.7p, neizpilde- a: -0.1p, b: -0.4p, c: -1.8p (jeb -0.3p par katru funkciju), d: -1.4p (jeb -0.35p par katru funkciju))

- a. Mainīgie (statiski): 1) *AbstractItem* tipa saraksts, kurā glabājas visas lietas, ko var paņemt no bibliotēkas. 2) *BorrowingItemsProcedure* tipa saraksts, kurā glabājas visi aizņemšanās procesi;
- b. main funkcijā izveidojiet vismaz divus *Class1 VarX* un divus *Class2 VarX* tipa objektus un tos ievietojat 6.a. punktā izveidotajā *AbstractItem* sarakstā. Izveidojiet arī trīs *BorrowingItemsProcedure* tipa objektus (kur katram piesaistat vismaz vienu *AbstractItem* objektu ar 5.g funkcijas palīdzību un kur katram piesaistat (vienkāršības labad) vienu un to pašu iepriekš izveidotu *Librarian* objektu) un ievietojiet šos objektus 6.a. punktā izveidotajā *BorrowingItemsProcedure* sarakstā.
- c. CRUD funkcionalitātes nodrošināšana darbam ar *Class1 VarX* objektiem:
 - i. Izveidojiet funkciju *getClass1 VarXById()*, kas atgriež konkrētu Jūsu varianta 2.punktā izveidotā klases tipa objektu no 6.a. punktā izveidotā *AbstractItem* saraksta, ja zināms šī objekta id.
 - ii. Izveidojiet funkciju *getClass1 VarXByItemCode()*, kas atgriež konkrētu Jūsu varianta 2.punktā izveidotā klases tipa objektu no 6.a. punktā izveidotā *AbstractItem* saraksta, ja zināms šī objekta *itemCode*.
 - iii. Izveidojiet funkciju *createNewClass1 VarX()*, kas ļauj pievienot jaunu Jūsu varianta 2. punktā izveidotās klases tipa objektu 6.a. punkta izveidotajā *AbstractItem* sarakstā. Neaizmirstam pārbaudīt, vai tāda lieta jau neeksistē 6.a. *AbstractItem* tipa sarakstā. Ja eksistē, tad nepievienojam jaunu, bet palielinām atrastā objekta *howManyCopies* mainīgā vērtību par vēlamo skaitu. Parametrus, kuri tiek izmantoti dublikātu pārbaudē, varat izvēlēties paši.
 - iv. Izveidojiet funkciju *getAllClass1 VarX()*, kas atgriež visus Jūsu varianta 2.punktā izveidotās klases tipa objektus no 6.a. punktā izveidotā *AbstractItem* saraksta. Neaizmirstam, ka šeit nepieciešams izfiltrēt tikai *Class1 VarX* tipa objektus no *AbstractItem* tipa saraksta.

- v. Izveidojiet funkciju *deleteClass1_VarXById()*, kas dzēs konkrētu Jūsu varianta 2.punktā izveidotā klases tipa objektu no 6.a. punktā izveidotā *AbstractItem* saraksta, ja zināms šī objekta id.
 - vi. Izveidojiet funkciju *updateClass1_VarXById()*, kas rediģē konkrētu Jūsu varianta 2.punktā izveidotā klases tipa objektu 6.a. punktā izveidotā *AbstractItem* sarakstā, ja zināms šī objekta id. Kā rediģējamos parametrus var padot *title* un *howManyCopies*.
- d. Papildu funkcionalitātes nodrošināšana:
- i. Izveidot funkciju *showAllLoanItemsByLibrarianPersonCode()*, kas sameklē un atgriež visus *AbstractItem* objektus, kurus ir izdevis viens un tas pats bibliotekārs. Var rasties situācija, ka viena un tā pati lieta ir paņemta atkārtoti. Šajā gadījumā rezultātā parādīsies šī lieta vairākas reizes.
 - ii. Izveidojiet funkciju *calculateHowManyItemsLoanByAdminLibrarians()*, kas izfiltrē un atgriež skaitu, cik daudz lietu ir izdevuši tieši *admin* tipa bibliotekāri.
 - iii. Izveidojiet funkciju *showAllLateBorrowingItems()*, kas izfiltrē un atgriež visus *AbstractItem* objektus, kuriem kaut viena *BorrowingItemsProcedure* objekta nodošanas termiņš ir pirms šīs dienas, tādējādi nosakot gadījumus, kuros būtu jāpiešķir soda nauda par laikā nenodotām lietām.
 - iv. * Izveidojiet funkciju *postponeDeadlineForItems()*, kura kā parametru saņem bibliotekāra personas kodu un vairākus *itemCode* objektus (šajā gadījumā pielietojam “...” sintaksi). Visām padotajām lietām (kuru *itemCode* kodi sakrīt ar padotajiem) pagarināt *returningDateTime* par 2 nedēļām. Atceramies pārbaudīt, ka tikai *admin* tipa bibliotekārs var pagarināt lietas. Šeit var būt situācija, ka no konkrētā *loanItems* saraksta grib pagarināt tikai vienu lietu. Tad tā lieta ir jāizņem no esošā *BorrowingItemsProcedure* objekta un jāsaglabā jaunā *BorrowingItemsProcedure* objektā ar 2 nedēļu pagarinājuma termiņu.

Veiksmi!

Pielikums Nr.1.



Pielikums Nr.2

Realizēt klases un mainīgos, balstoties uz Jūsu variantu. Mainīgo datu tipu varat izvēlēties pēc saviem ieskatiem.

Nosaukumi rakstīti latviski, bet programmā varat tos saukt, izmantojot angļu tulkojumu.

Varianta Nr.	Class1_VarX	varX_class1	varX_class	Class2_VarX	varX_class2	varX_class
1	Grāmata	ISBNumurs	enum pēc brīvas izvēles	EGrāmata	ISSNnumurs	String pēc brīvas izvēles
2	EGrāmata	ISSNnumurs	enum pēc brīvas izvēles	Disks	ietilpībaMB	String pēc brīvas izvēles
3	Disks	ietilpībaMB	enum pēc brīvas izvēles	Žurnāls	lpp	String pēc brīvas izvēles
4	Žurnāls	lpp	enum pēc brīvas izvēles	GaldaSpēle	vecumaIerob ežojums	String pēc brīvas izvēles
5	GaldaSpēle	vecumaIerob ežojums	enum pēc brīvas izvēles	Grāmata	ISBNumurs	String pēc brīvas izvēles
6	Grāmata	lpp	enum pēc brīvas izvēles	Disks	ietilpībaMB	String pēc brīvas izvēles
7	EGrāmata	kopējaisMinū šuSkaitis	enum pēc brīvas izvēles	Žurnāls	kopējaisRak stuSkaitis	String pēc brīvas izvēles
8	Disks	ietilpībaMB	enum pēc brīvas izvēles	GaldaSpēle	vecumaIerob ežojums	String pēc brīvas izvēles
9	Žurnāls	lpp	enum pēc brīvas izvēles	Grāmata	ISBNumurs	String pēc brīvas izvēles
10	GaldaSpēle	maxSpēlētāju Skaitis	enum pēc brīvas izvēles	EGrāmata	ISSNnumurs	String pēc brīvas izvēles

11	Grāmata	ISBNumurs	enum pēc brīvas izvēles	Žurnāls	lpp	String pēc brīvas izvēles
12	EGrāmata	kopējaisMinū šuSkaitis	enum pēc brīvas izvēles	GaldaSpēle	maxSpēlētāj uSkaitis	String pēc brīvas izvēles
13	Disks	ietilpībaMB	enum pēc brīvas izvēles	Grāmata	ISBNumurs	String pēc brīvas izvēles
14	Žurnāls	lpp	enum pēc brīvas izvēles	EGrāmata	ISSNnumurs	String pēc brīvas izvēles
15	GaldaSpēle	vecumaIerob ežojums	enum pēc brīvas izvēles	Disks	ietilpībaMB	String pēc brīvas izvēles
16	Grāmata	ISBNumurs	enum pēc brīvas izvēles	GaldaSpēle	vecumaIerob ežojums	String pēc brīvas izvēles
17	EGrāmata	ISSNnumurs	enum pēc brīvas izvēles	Grāmata	lpp	String pēc brīvas izvēles
18	Disks	ietilpībaMB	enum pēc brīvas izvēles	EGrāmata	kopējaisMin ūšuSkaitis	String pēc brīvas izvēles
19	Žurnāls	kopējaisRaks tuSkaitis	enum pēc brīvas izvēles	Disks	ietilpībaMB	String pēc brīvas izvēles
20	GaldaSpēle	maxSpēlētāju Skaitis	enum pēc brīvas izvēles	Žurnāls	lpp	String pēc brīvas izvēles
21	Grāmata	ISBNumurs	enum pēc brīvas izvēles	Disks	ietilpībaMB	String pēc brīvas izvēles
22	EGrāmata	ISSNnumurs	enum pēc brīvas izvēles	Žurnāls	lpp	String pēc brīvas izvēles
23	Disks	ietilpībaMB	enum pēc brīvas izvēles	GaldaSpēle	vecumaIerob ežojums	String pēc brīvas izvēles
24	Žurnāls	lpp	enum pēc brīvas	Grāmata	ISBNumurs	String pēc brīvas

			izvēles			izvēles
25	GaldaSpēle	vecumaIerob ežojums	enum pēc brīvas izvēles	Disks	ietilpībaMB	String pēc brīvas izvēles
26	Grāmata	ISBNumurs	enum pēc brīvas izvēles	Žurnāls	kopējaisRak stuSkaitis	String pēc brīvas izvēles
27	EGrāmata	kopējaisMinū šuSkaitis	enum pēc brīvas izvēles	GaldaSpēle	maxSpēlētāj uSkaitis	String pēc brīvas izvēles
28	Disks	ietilpībaMB	enum pēc brīvas izvēles	Grāmata	ISBNumurs	String pēc brīvas izvēles
29	Žurnāls	lpp	enum pēc brīvas izvēles	EGrāmata	ISSNnumurs	String pēc brīvas izvēles
30	GaldaSpēle	vecumaIerob ežojums	enum pēc brīvas izvēles	Žurnāls	lpp	String pēc brīvas izvēles