

## Невидимата библиотека

### Въведение

Студентите от невидимия университет се доказаха през годините, че имат способностите да разработват качествен софтуер. Най-случайно те решили да отидат да потърсят къде се намира библиотеката, защото им потрябвал учебник по математика. Те знаели, че трудно биха се мотивирали да го отворят и да видят какво има вътре, но за собствено успокоение, все пак били решили да се сдобият с такъв. Влизайки в библиотеката, място изпълнено със страх и прах, те решили директно да се втурнат през усойните коридори, над които бяха надвиснали високи рафтове с книги, и да потърсят учебното помагало. След часове безуспешно търсене прекарано в библиотеката, те все пак решили да се допитат до милата женица, която работи там. Тя с удоволствие им помогнала и вървейки им се оплаквала как трябвало да помни къде се намира всяка една книга, и дали изобщо е налична. На студентите им хрумнала гениална идея – да разработят софтуер, който да помогне на добрата дама от библиотеката. Грабнали учебниците и се втурнали към клавиатурите.

### Описание и характеристики

Нека да се разработи софтуер за управление на библиотека и наличните книги в нея. Дефинирайте класове съответно за библиотека и книга.

**Библиотеката** трябва да съдържа:

- наименование;
- адрес;
- брой служители;
- и списък с налични книги.

Всяка **книга** трябва да съдържа информация за:

- заглавие;
- жанр;
- автор/и;
- издателство;
- година на издаване;
- ISBN номер;
- брой страници;
- език, на който е написана книгата (*напр. английски език*);

## Самостоятелна работа 1

- наличност (дали книгата е налична или не);
- дата, на която е взета книгата от библиотеката;
- дата, на която трябва да се върне книгата от библиотеката;
- период на вземане (напр. 1 месец);
- колко пъти книгата е взимана от библиотеката;

*Бележка: Помислете и изберете подходящи модификатори за достъп и типове данни за всяка една от характеристиките на библиотеката и книгите. Реализирайте необходимите конструктори, методи, както и get/set методи. Това би се отразило на вашия точков актив.*

### Правила за четене от конзолата

Всички данни и характеристики, както за библиотеката, така и за книгите в нея, трябва да се въвеждат от потребителя чрез командния ред (от клавиатурата). Тъй като не трябва да се доверяваме на крайния потребител, то трябва да се реализират необходимите валидации, преди да запишем информацията за библиотека и/или книга.

### Правила за валидиране на данните

Валидации на данни за **библиотека**:

- *наименование* – задължително поле, не трябва да превишава 100 символа;
- *адрес* – задължително поле, не трябва да превишава 500 символа;
- *брой служители* – задължително поле, трябва да е по-голямо от 0 и по-малко от 50;

*Ако някое от условията не е удовлетворено, казваме на потребителя, че е въвел некоректни данни и го подканяме да опита отново, и така докато не въведе коректни данни.*

Валидации на данни за **книга**:

- *заглавие* – задължително поле, не трябва да превишава 100 символа;
- *автор/и* – единичен автор не бива да превишава 50 символа;
- *издателство* – задължително поле, не трябва да превишава 50 символа;
- *година на издаване* – задължително поле, трябва да бъде неотрицателно число и да **не** е в бъдещето;

## Самостоятелна работа 1

- ISBN номер – задължително поле, не трябва да превишава 10 символа;
- брой страници – задължително поле, трябва да бъде неотрицателно число и по-голямо от 0;
- наличност (*дали книгата е налична или не*) – задължително поле;
- колко пъти книгата е взимана от библиотеката – задължително поле, трябва да бъде неотрицателно число.

*Ако някое от условията не е удовлетворено, казваме на потребителя, че е въвел некоректни данни и го подканяме да опита отново, и така докато не въведе коректни данни.*

### **Правила за добавяне на книга**

Реализирайте функционалност, която да позволява на библиотекаря да добавя нова книга към списъка с книги на библиотеката. За целта библиотекарят трябва да избере опция от основното меню (виж по-долу) „добавяне на книга“ и след това ръчно да въведе данните за нея, като при неточност на данните, той трябва да ги въведе отново (виж секция „Правила за валидиране на данните“).

Ако книгата е въведена с коректни данни, добавяме я в списъка и извеждаме на потребителя съобщение, че книгата е успешно добавена и го връщаме към основното меню.

### **Правила за премахване на книга**

Реализирайте функционалност, която да позволява на библиотекаря да премахва съществуваща книга от списъка с книги на библиотеката. За целта библиотекарят трябва да избере опция от основното меню (виж по-долу) „премахване на книга“ и след това ръчно да избере коя книга да премахне. Това може да се случи чрез избор на книга по:

- наименование (*когато търсите книга по наименование, малки и главни букви не трябва да имат значение за целите на търсенето*);
- ISBN номер.

Ако книгата е намерена, премахваме я от списъка и извеждаме на потребителя съобщение, че книгата е успешно премахната и го връщаме към основното меню. Ако книгата не е намерена, извеждаме подходящо съобщение и връщаме потребителя отново към основното меню.

## Правила на за търсене на книга/и

Реализирайте функционалност, която да позволява на библиотекаря да търси книга/и от списъка с книги на библиотеката. За целта библиотекарят трябва да избере опция от основното меню (виж по-долу) „търсене на книга“ и след това ръчно да избере коя книга ще търси. Това може да се случи по:

- наименование
- автор
- година на издаване
- ISBN

Ако бъде намерена книга/и, по зададените от библиотекаря критерии, то трябва да бъде изведена в конзолата пълната информация за нея/тях (трябва да се изведат в конзолата в табличен вид всички характеристики на всяка една книга). След което потребителят трябва да бъде отведен към основното меню. Ако не е намерена книга/и, извежда се подходящо съобщение и отново отвеждаме потребителя към основното меню.

## Правила на обновяване на информация за книга

Реализирайте функционалност, която да позволява на библиотекаря да обнови информацията за съществуваща книга от списъка с книги на библиотеката. За целта библиотекарят трябва да избере опция от основното меню (виж по-долу) „обновяване на книга“ и след това ръчно да избере коя книга да обнови. Това може да се случи чрез избор на книга по:

- ISBN номер.

Ако книгата **не е намерена**, извеждаме подходящо съобщение и връщаме потребителя отново към основното меню. Ако книгата е намерена, извеждаме меню за избор на коя точно информация за книгата иска да обнови библиотекарят. **Примерно меню:**

*Избрахте за обновите информация за книга с ISBN номер: XXX*

*Моля изберете какво искате да обновите:*

- 1. наличност*
- 2. дата, на която е взета книгата*
- 3. дата, на която трябва да се върне книгата*
- 4. период на вземане*
- 5. колко пъти книгата е взимана от библиотеката*

След като потребителят направи своя избор, трябва да му се предостави възможност да въведе информация за избраната от него категория. Трябва да се приложат всички валидации, както при добавянето на книга и при невалидни такива, потребителят трябва да опита отново. След въвеждане на коректни данни, извеждаме на потребителя съобщение, че успешно е обновил информацията за книгата и го отвеждаме към основното меню.

### **Правила на изпечатване на пълен списък с книги**

Реализирайте функционалност, която да позволява на библиотекарят да изведе в табличен вид в конзолата всички налични книги в библиотеката. След което го пренасочете обратно към основното меню.

### **Правила на изход от основното меню**

При избор на опция „Изход“ от основното меню, библиотекарят трябва да види на екрана си съобщение „Довиждане! До нови срещи!“ и програмата да приключи своето изпълнение.

### **Правила на основното меню**

За да може библиотекарят коректно и правилно да извършва своята работа, трябва да реализирате меню, което да му дава достъп до всяка една от реализираните функционалности. Примерна визуализация на основно меню:

1. *Добавяне на книга*
2. *Премахване на книга*
3. *Търсене на книга*
4. *Обновяване на книга*
5. *Пълен списък с книги*
6. *Изход*

### **Правила при стартиране на програмата**

При стартиране на програмата, библиотекарят трябва да получи подканящо съобщение за създаване на библиотека, чрез въвеждането на всички необходими данни за нея. При опит за въвеждане на невалидни данни, операцията се повтаря, докато потребителят не въведе, удовлетворяваща валидациите, информация.

## *Самостоятелна работа 1*

При успешно въвеждане на всички данни за библиотека, потребителят се отвежда към основното меню.

Примерно меню при стартиране на програмата и процес по създаване на библиотека:

*Здравейте, моля въведете информация за библиотека!*

*Въведете наименование: Библиотека на ФМИ*

*Въведете адрес: бул. „България“ 236, 4027 Пловдив*

*Въведете брой служители: 0*

*Броят на служителите в библиотеката трябва да е по-голям от 0 и по-малък от 10! Моля опитайте отново!*

*Въведете брой служители: 3*

След приключване на това действие, отвеждаме потребителя към основното меню, където може да се продължи с действия по добавяне на книга.

## ЗАДАЧИ ЗА ИМПЛЕМЕНТАЦИЯ

Всяка една от разработените задачи ще се оценява в точкова система с арбитрарни тежести. Максималното количество точки, които можете да получите от решението на поставените ви условия е **100 точки**. Повече информация за предаването на вашите решения можете да намерите в секция ПРЕДАВАНЕ НА РЕШЕНИЯТА.

| функционалност  | точки |
|---|-------|
| създаване на необходимите класове, характеристики, методи, конструктори, get/set методи, използване на правилни и подходящи ООП принципи, правила и парадигми | 20    |
| избор на подходящи модификатори за достъп и типове данни  | 5     |
| реализация на връзка между библиотека и книги   | 5     |
| прочитане на данните от конзолата   | 10    |
| реализиране на валидации на данните   | 10    |
| реализация на търсене на книга по дадени критерии   | 10    |
| реализация на добавяне на книга към библиотеката  | 10    |
| реализация на премахване на книга от библиотеката   | 10    |
| реализация на обновяване на книга от библиотеката   | 10    |
| реализация на изпечатване на списъка с книги в табличен вид   | 10    |

## КРИТЕРИИ ЗА ВАЛИДНА ОЦЕНКА

В тази част са описани критериите, които добавят положителен резултат към крайната оценка. За да бъдат приложени е необходимо да не е влязло в сила, нито едно от правилата описани в предходната част.

Едно домашно няма да бъде проверявано или ще бъде наказвано с **0 точки** при неговото предаване, ако налице са някои от следните обстоятелства.

| обстоятелство   | наказание                       |
|---|---------------------------------|
| <ul style="list-style-type: none"> <li>Програмата не се компилира;</li> <li>Програмата е предадена във вид, който изисква допълнителна намеса от страна на проверяващия, за да стартира;</li> <li>Изпълними файлове с грешно разширение.</li> </ul> | 0 точки                         |
| Домашна работа, предадено извън срока описан в заданието  | 0 точки                         |
| <i>Нагло плагиатство, дословно копиране и погазване на моралните принципи за справедливо предаване на домашна работа</i>  | <i>Слаб (2) за редовна дата</i> |
| Грубо погазване на описаните правила и тотално неспазване на условията на домашните работи  | 0 точки                         |

ПРОЦЕНТНО СЪОТНОШЕНИЕ НА КРИТЕРИИТЕ, СПРЯМО ДАДЕНИТЕ ТОЧКИ

| Графична имплементация  | проценти |
|---|----------|
| <ul style="list-style-type: none"> <li>Именуване на променливи и методи</li> </ul>              | 30%      |
| <ul style="list-style-type: none"> <li>Коректност на имплементираната функционалност</li> </ul> | 70%      |

## ИМЕНУВАНЕ НА ПРОМЕНЛИВИ И МЕТОДИ

Правилното именуване на методите и променливите са ключови за лесното разбиране на кода, което е свързано с предишните два критерия.

## КОРЕКТНА ФУНКЦИОНАЛНОСТ

Това е критерият с най-голяма тежест, тъй като целта на всяко задание в крайна сметка е коректно работещо приложение. Функционалността в текущото задание е разделена в няколко точки. Затова и всяка правилно реализирана функция прибавя резултат към крайната оценка.

## ПРЕДАВАНЕ НА РЕШЕНИЯТА

Всички участници трябва да предадат задачата си, използвайки съвременна и проверена схема за дистрибуция на софтуерни инструкции.

Задължително е да си направите нов проект в **GitLab**-а на курса, със заглавие: ***pu-fmi--java-oop-h1***



Името на хранилището не бива да бъде бъркано, преправено, импровизирано или подложено на творческа интерпретация. Ако това се случи, за ваше съжаление задачата ви няма да бъде приета и проверена, което ще донесе само и единствено тъга в сърцата на всички панди в средната земя.

Линк към хранилището трябва да бъде предоставен в **Google Classroom** преди крайния срок на заданието. За целта, трябва да го прикачите към заданието и след това да го предадете като натиснете бутон „**Предай**“. Хранилище, което е предоставено като коментар няма да бъде отворено и оценено!

Ако изпитвате затруднение комуникирайте с преподавател за правилния начин, по който да предадете решението. Като е добре това да се случва в срокове различни от последната минута на последния ден.

**Успех!**

---