

# Библиотека

## Общ преглед

Във вашата фирма постъпва проект за създаване на приложение, обслужващо „библиотека“. Библиотеката, с която ще работим е нестандартно малка и притежава само един рафт. Съответно, вашият софтуер трябва да описва **рафт (Shelf)** и **книга (Book)**.

Трябва да реализирате функционалност, която да позволява добавяне и премахване на книги, проверка за налични книги, премахване на книги и прочее – всичко това ще работи чрез **команди**, които вие ще получавате. Поредицата от команди приключва с „**END**“.

За Ваше удобство ще получите готов Program.cs файл, и ще трябва да реализирате само необходимите класове **Shelf.cs** и **Book.cs**

**Основната идея се базира на това, че т.нар. рафт е структура, която ще съдържа n на брой книги. Структурата не трябва да пази книгите в колекция! Всяка книга пази референция към следващата в поредицата.**

## Подзадача 1: 30 точки

### Book

Всички книги имат идентификационен код и референция към следваща книга:

- id – низ, съставен от малки и/или големи латински букви и цифри
- next – референция към следваща книга

Book.cs
<pre>public Book(string bookId) {     //TODO: Добавете вашия код тук ... }  public string BookId {     //TODO: Добавете вашия код тук ... }</pre>

```
public Book Next
```

```
{
```

```
    //TODO: Добавете вашия код тук ...
```

```
}
```

```
public override string ToString()
```

```
{
```

```
    //TODO: Добавете вашия код тук ...
```

```
}
```

## Shelf

„Рафтът“ държи информация за Book head, Book tail, Count:

head – **Book**, първа в поредицата

tail – **Book**, последна в поредицата

count – **брой** книги

Shelf.cs

```
public Shelf()
```

```
{
```

```
    //TODO: Добавете вашия код тук ...
```

```
}
```

```
public int Count
```

```
{
```

```
    //TODO: Добавете вашия код тук ...
```

```
}

public void AddBook(string bookId)
{
    //TODO: Добавете вашия код тук ...
}

public void AddSpecialBook(string bookId)
{
    //TODO: Добавете вашия код тук ...
}

public Book CheckBookIsPresent(string bookId)
{
    //TODO: Добавете вашия код тук ...
}

public bool ReleaseBook(string bookId)
{
    //TODO: Добавете вашия код тук ...
}

public bool ReleaseBook(int index)
{
    //TODO: Добавете вашия код тук ...
}
```

```

}

public StringBuilder ShelfInformation()
{
    //TODO: Добавете вашия код тук ...
}

```

Класът Shelf трябва да имплементира следните методи

- **Метод за добавяне на книга**

**Add <bookId>** - този метод има за цел да добави книга с нейният идентификационен код в края на редицата от книги, тоест като последен елемент.

- **Метод за генериране на информация относно структурата**

**ShelfInformation**– Трябва да съберете информация за всички налични книги в структурата в StringBuilder като спазвате следния формат:

```

Book {id}
Book {id}
..

```

**В случай на празна колекция от книги добавете само един ред - <Shelf is empty!>**

За успешна реализация трябва да реализирате ваша версия на **ToString()** метода за класа **Book**.

- **Getter за осигуряване на информация относно броя книги в колекцията**

**Count** – Трябва да върнете информация за актуалния брой книги в колекцията

## Подзадача 2: 30 точки

- **Метод за премахване на книга по индекс**

**ReleaseBook <index>** - Трябва да бъде премахнат елемент, който се намира на посочения индекс. Тъй като вашата структура не използва индексирание, удобен похват би бил използването на брояч. При успешно намиране и премахване на **Book** трябва да върнете **true**, което ще бъде доведе до изпечатване на текст **< Released book >** на конзолата от Main метод-а. При ненамиране на такъв **Book**, трябва

да бъде върната **false** стойност.

При получаване на индекс извън обхвата на колекцията – напр. отрицателно число – върнете отново **false**.

- **Метод за премахване на книга по ID**

**ReleaseBook < bookId >** - Трябва да бъде премахнат първият елемент, на който **id**-то отговаря на подаденото. При успешно намиране и премахване на **Book** трябва да върнете **true**, , което ще бъде доведе до изпечатване на текст **< Released book >** на конзолата от Main метод-а. При ненамиране на такъв **Book**, трябва да бъде върната **false** стойност.

## Подзадача 3: 20 точки

- **Метод за проверка за наличност на книга**

**CheckBookIsPresent < bookId >** - Трябва да бъде намерен елемент, на който **id**-то отговаря на подаденото. При успешно намиране **Book** трябва да върнете **обекта** в обратен случай **null** стойност.

## Подзадача 4: 20 точки

- **Метод за добавяне на книга най-отпред в поредицата**

**AddSpecial < bookId >** - Трябва да добавите нова книга най-отпред в поредицата от книги.

Команди подавани на конзолата

Вашето приложение реализира следните команди:

- **Add < bookId >** - Добавя Книга към структурата
- **AddSpecial < bookId >** - Добавя Книга като първи елемент в структурата
- **Info** – Изпечатва се информация за всички налични книги
- **Release <int index>** - Опитва да премахне елемент по **index**
- **Release < string bookId >** - Опитва да премахне елемент по **bookId**
- **Check < bookId >** - При намерена книга – **<Book < bookId > is present>** в обратен случай – **< Book is not present>**
- **Count** - Отпечатва **< Books count: <брой книги > >**

- Програмата ще получава множество редове с информация. Всеки ред представлява команда. Самият вход се обработва изцяло от примерния Program.cs.
- Всички **bookId** ще бъдат уникални
- Всички команди приключват с въвеждането на END

## Вход / Изход

Вход

Изход

За някои от командите не е нужно да извеждате нищо. За други е необходимо форматиране на изход – напр. **Info**

Ограничения

- Имената няма да съдържат интервал

Примери

Вход	
Add book	Books count: 2
Add book1	Book: book
Count	Book: book1
Info	Released book
Release book1	Books count: 1
Count	Books count: 2
Add book2	Book: book
Count	Book: book2
Info	Released book
Release 0	Books count: 1
Count	Book: book2
Info	Book: book2 is present.
Check book2	Book is not present
Check book	Book not found

Release 125	Book not found
Release pesho	Book: specialBook
AddSpecial specialBook	Book: book2
Info	Book: specialBook2
AddSpecial specialBook2	Book: specialBook
Info	Book: book2
Release specialBook2	Released book Book: specialBook
Info	Book: book2
END	