

Тема 4, част 1

**Пример за използване на вектори
и списъци: библиотечна „база от
данни“**

Примерна библиотечна „база от данни“ (Example: A Library Database)

Цел: автоматизиране на работата с
библиотечен каталог

Emma Mitchell
borrowed
The Craft of Functional Programming

Michael Strang
borrowed
Algorithms and Data Structures

Представяне на картон от картотеката на заемната служба

Каква информация съдържат картоните от
картотеката на заемната служба?

Име на читателя

Заглавие на книгата

Emma Mitchell
borrowed
The Craft of Functional Programming

```
type Borrower = String
```

```
type Book = String
```

```
type Card = (Borrower, Book)
```

Представяне на картотеката

Какво съдържа картотеката?

➡ **поредица от картони за заети книги**

```
type Database = [Card]
```

```
example :: Database
```

```
example = [ ("Emma Mitchell", "The Craft of Functional Programming"),  
            ("Michael Strang", "Algorithms and Data Structures") ]
```

Заявки към базата от данни

Каква информация ще бъде полезно да бъде извлечена?

`books :: Database -> Borrower -> [Book]`

`borrowers :: Database -> Book -> [Borrower]`

`borrowed :: Database -> Book -> Bool`

`numBorrowed :: Database -> Borrower -> Int`

Кои книги са заети от даден читател?

books example "Emma Mitchell" =

["The Craft of Functional Programming"]

books example "Mary Sheeran" = []

Няма заети книги.

books db person =

[book | (borrower, book) <- db, borrower == person]

Отново съпоставяне с
образец (Pattern Matching).

Упражнение

Да се дефинира функция:

`numBorrowed :: Database -> Borrower -> Int`

Упражнение

Да се дефинира функция:

`numBorrowed :: Database -> Borrower -> Int`

`numBorrowed db person = length (books db person)`

Обновяване на базата от данни

Заемането и връщането на книги променя съдържанието на базата от данни.

`makeLoan :: Database -> Borrower -> Book -> Database`

`returnLoan :: Database -> Book -> Database`


Тези функции използват старото съдържание на базата от данни...

... „Пресмята“ се новото съдържание на базата от данни.

Заемане на книга

`makeLoan :: Database -> Borrower -> Book -> Database`

`makeLoan db borrower book = [(borrower, book)] ++ db`



Създаване (попълване)
на нов картон.

Връщане на книга (вариант 1)

`returnLoan :: Database -> Book -> Database`

`returnLoan db book =`

`[(borrower, book1) | (borrower, book1) <- db,
book1 /= book]`

Връщане на книга (вариант 2)

`returnLoan :: Database -> Borrower -> Book -> Database`

`returnLoan db borrower book =`

`[(borrower1, book1) | (borrower1, book1) <- db,`

`(borrower1, book1) /= (borrower, book)]`