Курс по информационни технологии

ДАВИД холдинг АД | гр.“Казанлък“, ул.“Стара река“ 2, ДК „Арсенал“, ет.4

Упражнения

Валери Дачев и Александър Далемски

2013

# Съдържание

[Съдържание 1](#_Toc366110223)

[Библиотека 2](#_Toc366110224)

[Задача 2](#_Toc366110225)

[Изисквания 2](#_Toc366110226)

[Допълнителни задачи 3](#_Toc366110227)

[Академия 3](#_Toc366110228)

[Задача 3](#_Toc366110229)

[Изисквания 3](#_Toc366110230)

[Допълнителни задачи 4](#_Toc366110231)

# Библиотека

## Задача

Създайте база данни, съхраняваща информация за книгите в библиотека, нейните членове и взетите от тях книги. Създайте и приложение, което да предоставя операции за взаимодействие с базата данни.

## Изисквания

* В предметната област съществуват следните типове обекти:
  + Автор – характеризира се с *име*, *година на раждане* и *година на смъртта* (ако е починал);
  + Книга – характеризира се със *заглавие*, *един или повече автори* и *година на издаване*;
  + Екземпляр – това е физически екземпляр на книга, тъй като в библиотеката може да има повече от един екземпляр на дадена книга; характеризира се с *книга*, на която е екземпляр, *уникален код на екземпляра* и това *дали в момента е взет от някой член на библиотеката*;
  + Член – това е човек, който е платил членския си внос в библиотеката и има право да взема екземпляри на книги от нея; характеризира се с *три имена*, *ЕГН* и *дата, на която е станал член*;
  + Вземане на екземпляр от член на библиотеката – характеризира се с *екземпляр*, който е бил взет, *член*, който го е взел, *дата на вземане на екземпляра*, *краен срок за връщане на екземпляра* и *дата, на която екземплярът е бил върнат* (ако това е така);
* В предметната област съществуват следните множества от зависимости:
  + Зависимости много-към-много между автори и книги;
  + Зависимости едно-към-много между книги и екземпляри;
  + Зависимости едно-към-много между екземпляри и вземанията им от членове на библиотеката;
  + Зависимости едно-към-много между членове и вземанията на екземпляри, извършени от тях;
* Да се създаде база данни в SQL сървър, която да може да съхранява в добре структуриран вид данните от предметната област;
* Да се създаде конзолно приложение, в което да се реализира меню със следните операции:
  + Извеждане на списъка от автори;
  + Извеждане на списъка от книги на конкретен автор;
  + Извеждане на списъка от екземпляри на конкретна книга;
  + Извеждане на списъка от вземания на конкретен екземпляр;
  + Извеждане на списъка от членове на библиотеката;
  + Извеждане на списъка от вземания, извършени от конкретен член на библиотеката;
  + Регистриране на автор;
  + Регистриране на книга;
  + Регистриране на екземпляр на книга;
  + Регистриране на член на библиотеката;
  + Регистриране на вземане на екземпляр от член на библиотеката;
  + Регистриране на връщане на екземпляр, който преди това е бил взет от член на библиотеката (като се променя съответния запис в таблицата с вземанията).

## Допълнителни задачи

* Да се реализират и операции за редактиране на вече регистрираните в базата данни обекти, както и за изтриването им;
* Да се създаде Entity/Relationship Model при проектирането на базата данни;
* Да се провери и осигури спазването на втора нормална форма за всяка от таблиците в базата данни;
* Манипулациите с базата данни да се осъществят чрез LINQ to SQL;
* Вместо конзолно, приложението за опериране с базата данни да бъде desktop приложение.

# Академия

## Задача

Да се създаде приложение, което служи за регистриране и разглеждане на информацията, свързана с протичащите курсове в академия за извънкласно обучение, участниците в тези курсове и поставените им курсови задачи.

## Изисквания

* Информацията да се съхранява в база данни или в подходящо структуриран(и) XML документ(и) във файловата система;
* Предметната област включва следните типове обекти:
  + Курс – характеризира се с *наименование*, *описание*, *ниво* (изброен тип със стойности начинаещи, напреднали и специалисти), *начална дата* и *крайна дата*;
  + Допълнителен материал към курс – характеризира се със *заглавие*, *описание*, *автор* и *курс, за който е предназначен*;
  + Курсова задача – характеризира се със *заглавие*, *условие*, *дали е задължителна* и *курс, към който принадлежи*;
  + Участник – характеризира се с *три имена*, *email адрес*, *дата на присъединяване* към академията и *курсове, които посещава (или е посещавал)*;
* Предметната област включва следните множества от зависимости:
  + Зависимост едно-към-много между курсове и допълнителни материали към тях;
  + Зависимост едно-към-много между курсове и курсови задачи към тях;
  + Зависимост много-към-много между курсове и участници в тях;
* Приложението да позволява следните операции:
  + Извеждане на списък с всички курсове;
  + Извеждане на списък с курсовете, в които участва конкретен участник;
  + Извеждане на списък с всички материали към конкретен курс;
  + Извеждане на списък с всички курсови задачи към конкретен курс;
  + Извеждане на списък с всички участници в академията;
  + Извеждане на списък с участниците в конкретен курс;
  + Регистриране на нов курс;
  + Регистриране на нов допълнителен материал към курс;
  + Регистриране на нова курсова задача към курс;
  + Регистриране на нов участник в академията;
  + Включване на участник в курс;
* Да бъдат спазени принципите на ООП;
* Да се предотврати възникването на грешки в програмата при въвеждането на некоректни стойности от потребителя;
* Ако данните се съхраняват в база данни, да се осигури защита от SQL Injection атаки;
* Ако данните се съхраняват в база данни, да се използва СУБД Microsoft SQL Server;
* Приложението може да бъде конзолно, графично или уеб базирано.

## Допълнителни задачи

* Да се реализират и следните типове обекти и множества от зависимости от предметната област (заедно с операции за манипулиране с тях):
  + Решение на курсова задача – характеризира се с *участник, който го е изработил*, *курсова задача, върху която е разработено*, *дата на предаване*, както и *дали е било одобрено*;
  + Сертификат за завършен курс – характеризира се с *участник, който го е получил*, *курс, за който е издаден*, *резултат* (цяло число между 0 и 100), както и *дата, на която е издаден*;
  + Зависимост едно-към-много между участници и решения на курсови задачи;
  + Зависимост едно-към-много между курсови задачи и решения;
  + Зависимост едно-към-много между участници и сертификати за завършени курсове;
  + Зависимост едно-към-много между курсове и сертификати;
* Приложението да позволява и модификация и изтриване на вече регистрирани обекти;
* Да се реализират операции за извеждане на филтрирани списъци с обекти (например всички курсове с конкретно ниво, или всички курсови задачи, чиито условия включват определена дума/символен низ);
* Ако данните се съхраняват в база данни, да се използва LINQ to SQL или Entity Framework за манипулациите с нея;
* Ако данните се съхраняват в XML документ(и), да се използва механизмът за сериализиране/десериализиране на обекти;
* Приложението да бъде с графично или уеб базирано приложение.