**ИT Кариера**

**Каталог за филми**

**Работа с Visual Studio**

**Разработили: Алкин Вежди,**

**Калина Йорданова,**

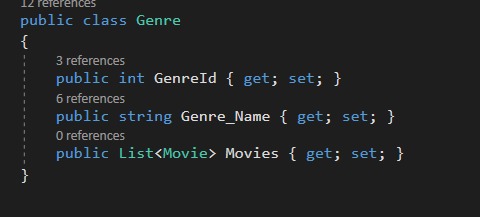
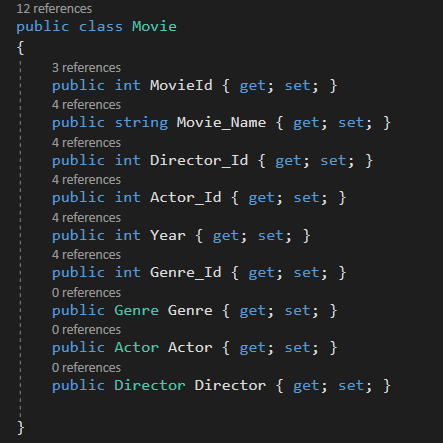
**Владимир Владимиров.**

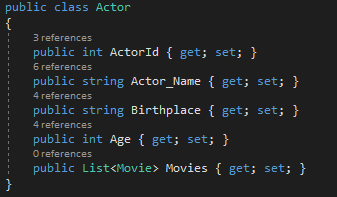
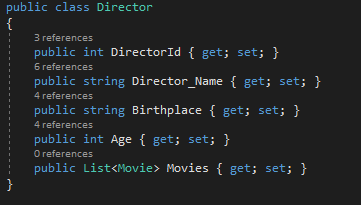
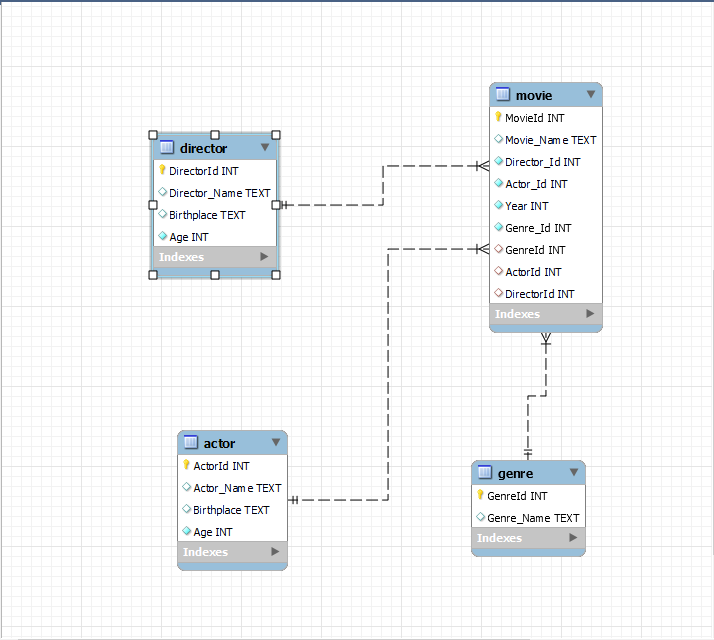
**Дата: 20.03.2022г.**

**Съдържание**

1. **Ентитита и връзки – 3 стр.**
2. **Свързване на Visual Studio и MySQL – 4 стр.**
3. **Примери за въвеждане на актьори, филми, жанрове и директори – 7 стр.**
   1. **Въвеждане – 7 стр.**
   2. **Редактиране – 10 стр.**
   3. **Извличане – 11.стр.**
   4. **Изтриване -12 стр.**
   5. **Показване на въведената информация – 13 стр.**

**Курсов проект на тема „Каталог за филми“.**

Това са 4 – те главни „entity“ – та ( таблици). Таблиците директор, актьор и жанр са свързани с таблицата филми едно към много.



**Свързване на Visual Studio с MySQL**

**Text

Description automatically generated**Връзката между Visual Studio и MySQL се случва чрез създаване на „db“ – сетове за да може да се направи връзка с базата данни.

Първата стъпка която направихме в проекта беше да направим менюто, за да може да имаме идея с какво трябва да разполага нашата програма.

Text

Description automatically generated

‚

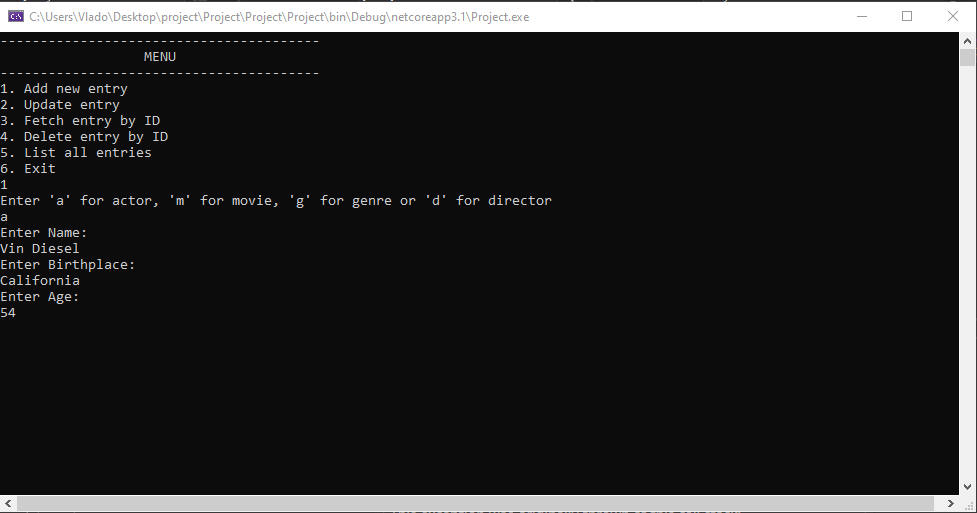
Менюто разполагащо със шест опции. Първата опция за вписване има няколко под теми.

Text

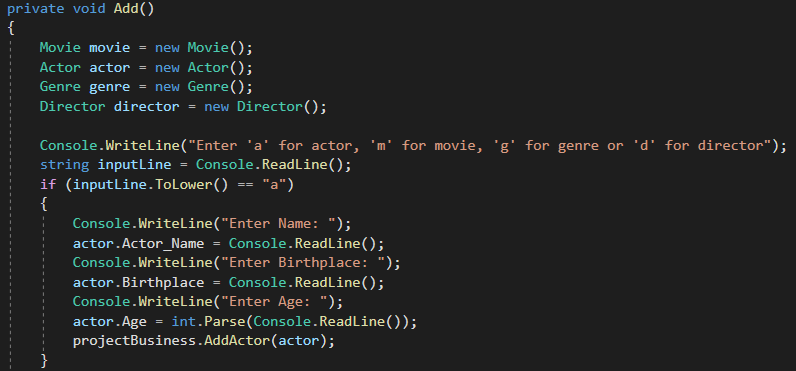
Description automatically generatedОпциите са за вписване на актьори, филми, жанрове и директори чрез изписване на определена буква в зависимост от това какво искате да въведете.

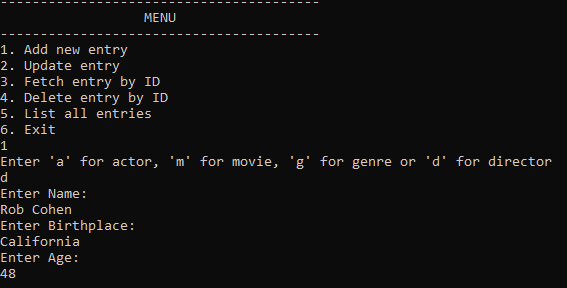
**Примери за въвеждане на актьори, филми, жанрове и директори .**

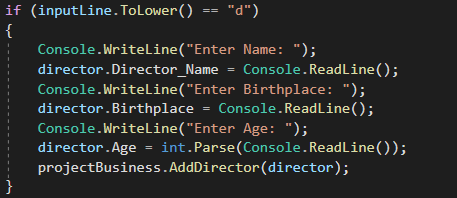
**Въвеждане**

****

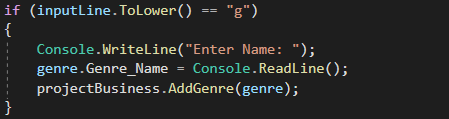
Въвеждането на актьори започва с въвеждането на буквата „а“, след което следва да се запишат имената, мястото на раждане и годините на актьора.

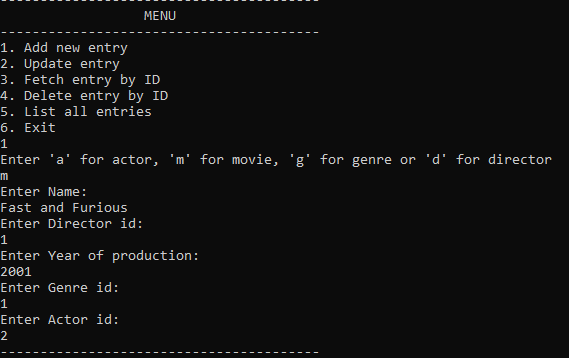
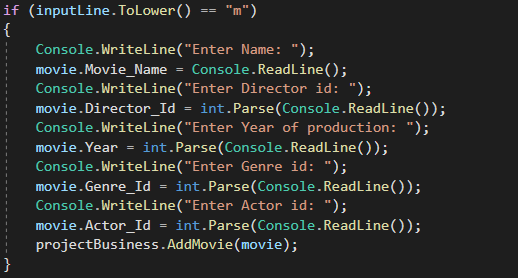
****

Същият метод се използва за въвеждане на другите три категории.

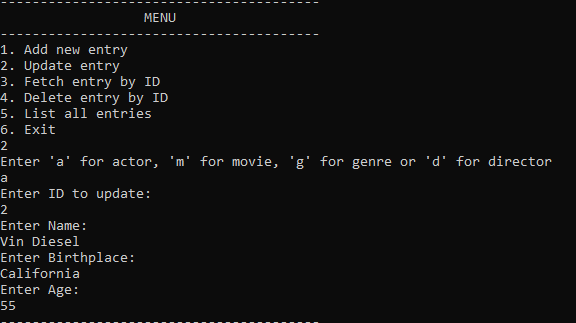
Text

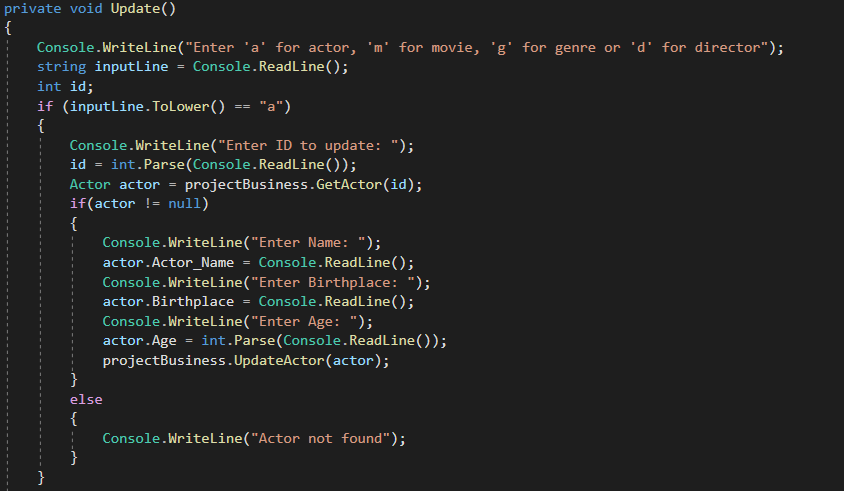
Description automatically generated



**Редактиране**

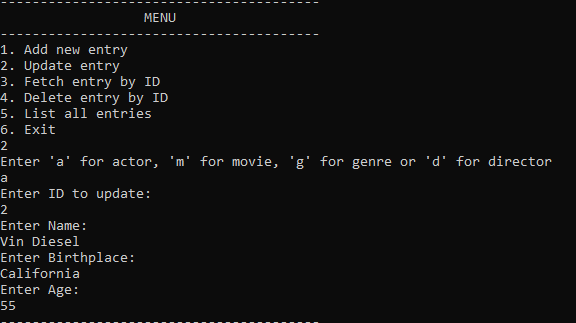
Втората опция е за промяна във вече записаните въвеждания. Като ще покажем пример с промяната не годините в категорията „актьори“, чрез въвеждане на „ID“ – то на актьора може да се редактира името, мястото на раждане и годините. Също има възможност за редактиране на всяка категория .





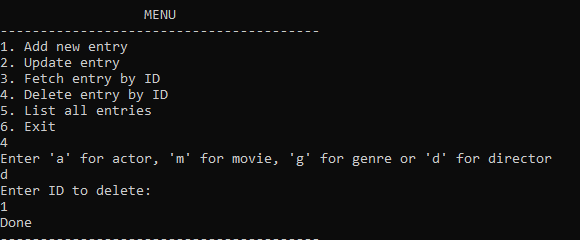
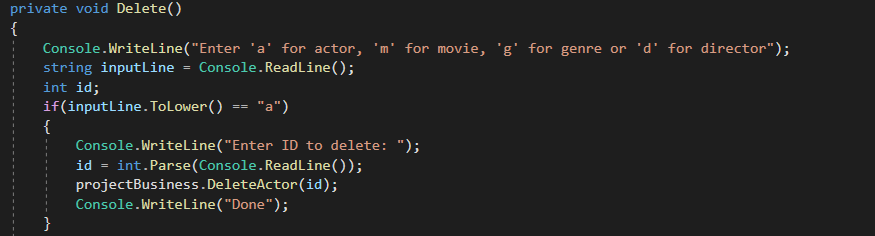
**Извличане**

Третата опция служи за извличане на информация от базата данни, след въвеждане на „3“ трябва да се избере от кои от въведените данни да се извлече информацията, ще дадем пример с извличане на актьор чрез вписване на „Id“ то му или номера под който е въведен



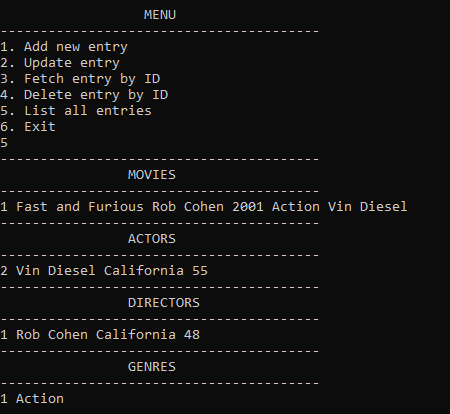


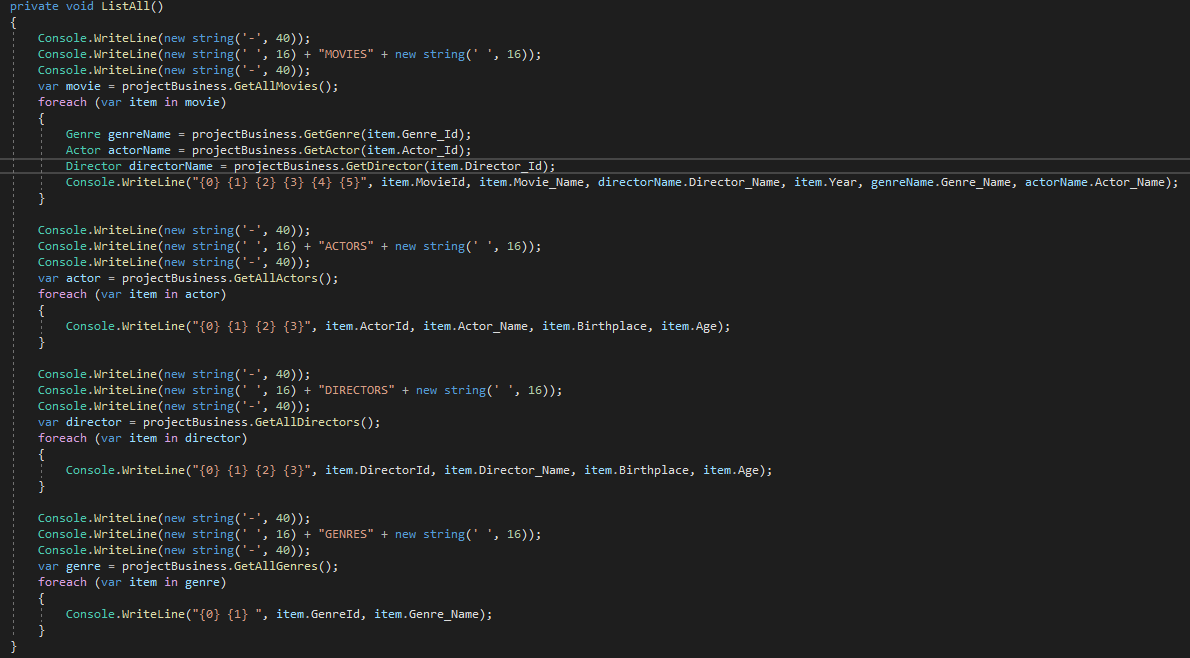
**Изтриване**

Четвъртата опция е за пълно изтриване на въведената от вас информация чрез въвеждане на „4“ и след това избор на категория от която искате да изтриете определените данни, въвежда се „Id“ – то и следва пълно изтриване от базата данни.

**Показване на въведената информация**

След като сте въвели дадената от вас информация можете да проверите дали има нещо изпуснато или грешно чрез въвеждане на „5“.





**XML File**

<doc>

<assembly>

<name>Project</name>

</assembly>

<members>

<member name="M:Project.Business.ProjectBusiness.GetAllMovies">

<summary>

Get all movies from the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.GetAllActors">

<summary>

Get all actors from the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.GetAllGenres">

<summary>

Get all genres from the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.GetAllDirectors">

<summary>

Get all directors from the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.GetMovie(System.Int32)">

<summary>

Get a single movie from the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.GetActor(System.Int32)">

<summary>

Get a single actor from the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.GetGenre(System.Int32)">

<summary>

Get a single genre from the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.GetDirector(System.Int32)">

<summary>

Get a single director from the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.AddMovie(Project.Data.Model.Movie)">

<summary>

Add a movie to the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.AddActor(Project.Data.Model.Actor)">

<summary>

Add an actor to the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.AddGenre(Project.Data.Model.Genre)">

<summary>

Add a movie to the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.AddDirector(Project.Data.Model.Director)">

<summary>

Add an director to the database

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.UpdateMovie(Project.Data.Model.Movie)">

<summary>

Update a single movie in the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.UpdateActor(Project.Data.Model.Actor)">

<summary>

Update a single actor in the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.UpdateGenre(Project.Data.Model.Genre)">

<summary>

Update a single genre in the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.UpdateDirector(Project.Data.Model.Director)">

<summary>

Update a single director in the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.DeleteMovie(System.Int32)">

<summary>

Delete a movie from the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.DeleteActor(System.Int32)">

<summary>

Delete an actor from the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.DeleteGenre(System.Int32)">

<summary>

Delete a genre from the database by id

</summary>

</member>

<member name="M:Project.Business.ProjectBusiness.DeleteDirector(System.Int32)">

<summary>

Delete a director from the database by id

</summary>

</member>

<member name="M:Project.Presentation.Display.#ctor">

<summary> Constructor </summary>

</member>

<member name="M:Project.Presentation.Display.ShowMenu">

<summary> Menu UI </summary>

</member>

<member name="M:Project.Presentation.Display.Input">

<summary> User Input </summary>

</member>

<member name="M:Project.Presentation.Display.Delete">

<summary> US for Deletion </summary>

</member>

<member name="M:Project.Presentation.Display.Fetch">

<summary>

UI for getting records from the database

</summary>

</member>

<member name="M:Project.Presentation.Display.Update">

<summary>

UI dor updating records in the database.

</summary>

</member>

<member name="M:Project.Presentation.Display.Add">

<summary>

UI for adding records to the database.

</summary>

</member>

<member name="M:Project.Presentation.Display.ListAll">

<summary>

UI to list all the products.

</summary>

</member>

</members>

</doc>