7-изпитен билет ПП-Разработка на софтуер

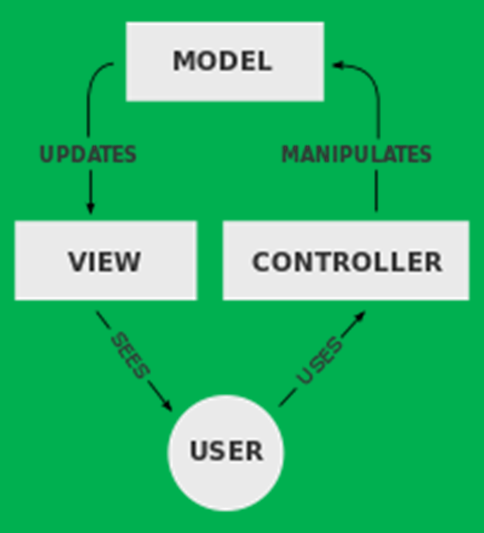
1.Трислойни модели Model-View-Controller и Model-View-ViewModel – графично представяне, начин на изпълнение

MVC-Разделя приложението на три части: модел, изглед и контролер. Шаблон за дизайн (Design Pattern) с три свързани един с друг компонента.

Модел (Model) – управлява данните и логиката на приложението

Изглед (View) – слой за представяне (presentation layer)

Контролер (Controller) – преобразува входните данни в команди и ги изпраща към Изгледа или към модела



Основните компоненти на MVVM са модел, изглед и модел. На първо място, моделът държи данните. Второ, viewmodel работи като връзка или връзка между модела и изгледа. Той преобразува обектите за данни от модела по такъв начин, че обектите лесно се управляват и представят. И накрая, изгледът съдържа потребителските интерфейси.



2.Рефакториране на код

Пренаписване на съществуващ изходен код, за да се подобри неговата разбираемост, повторна употреба или структура, без да се засяга неговото значение или поведение.

Рефакторирай дадения код и опиши начина по-който си го извършил.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static void Main(string[] args)

{

Console.Title = "Моят проект";

Console.ForegroundColor = ConsoleColor.Yellow;

Console.BackgroundColor = ConsoleColor.Blue;

Console.WriteLine("Здравейте,това е моят проект!");

Console.BackgroundColor = ConsoleColor.Black;

Console.ReadLine();

}

}

}

3.Видове тествания на кода (компонентен, интеграционен, регресионен и системен).

- компонентен тест- Компонентен тест е код, който обикновено се пише от разработчика, с цел да се провери и гарантира правилното поведение на компонент. Компонентните тестове се поставят в хранилището на кода (TFS / SVN / Git) заедно с кода, който тестват. Обикновено се обособяват в собствена папка и проект

-интеграционен тест- Основната цел на тестовете за интеграция е да се провери дали отделните модули работят правилно, когато се комбинират.

- регресионен тест-провеждаме тези тестове, за да гарантираме, че новите промени нямат отрицателни странични ефекти върху съществуващата функционалност. Целта е старият код да продължи да работи, след като са направени промени.

- системен тест- Системният тест анализира системата от друга гледната точна на клиента и на бъдещия потребител. Голяма част от функциите и системните характеристики са резултат от взаимодействието на всички системни компоненти.

За целта на системното тестване, на тестера се предоставя среда, която се доближава до тази на потребителя по:

- хардуер;

- системен софтуер;

- драйвери за определени устройства;

- мрежи;

- външни системи.

4.Писане на SQL заявки чрез език за програмиране- Създаването на SQL заявки се основава на прилагането на няколко основни правила за работа с релационна база данни. Писането на SQL заявка ще помогне да се реализират задачите за избор на конкретна информация от таблици, добавяне, промяна или изтриване на редове в таблица.

За SqlClient доставчик на данни се използва:

SqlConnection

Създава връзката към БД на SQL Server

SqlCommand

Изпълнява SQL команди върху SQL Server чрез създадената връзка

Може да приема параметри (SQLParameter)

SqlDataReader

Извлича данни (множество от записи) от SQL Server  
като резултат от изпълнението на SQL заявка

5.Използване на системи за обектно-релационно свързване (ORM frameworks) – разлики в подходите Code First и Database First.

Entity Framework опростява достъпа до данни във вашето приложение, като ви позволява да пишете код за извършване на CRUD (Създаване, четене, актуализиране и изтриване) операции, без необходимостта от директно взаимодействие с основния доставчик на база данни. Има три подхода за моделиране на вашите обекти в Entity Framework: Code First, Model First и Database First

Подходът Code First ви помага да създадете обекти във вашето приложение, като се фокусирате върху изискванията на домейна. При този подход обикновено създавате класове обекти.

Недостатъкът на този подход е, че всички промени в основната схема на базата данни ще бъдат загубени; при този подход вашият код определя и създава базата данни. Подходът Code First ви позволява да използвате Entity Framework и да дефинирате модела на обекта без дизайнерските или XML файлове.

Можете да използвате подхода Database First, ако базата данни вече е проектирана и е готова. Ръчните промени в базата данни са възможни лесно и винаги можете да актуализирате, ако е необходимо.

8.Дефинира понятието пакет и пакетен мениджър Системата за управление на пакети (известен също като „мениджър на пакети“) е програма, която ви позволява автоматично да инсталирате и премахвате софтуерни пакети, както и да ги актуализирате и конфигурирате.