Министерство образования и науки

Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

УТВЕРЖДАЮ

|  |  |  |
| --- | --- | --- |
| Зав. профилирующей кафедрой | | |
|  | | |
| уч. ст., уч. зв. | | |
|  | | |
|  |  |  |
| подпись, дата |  | иниц., фамилия |

**ОТЧЕТ ПО ПРАКТИКЕ**

Студента\_\_\_\_ курса \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ факультета

наименование факультета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

вид практики

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

кафедра

курс\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

семестр\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

продолжительность\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

кол. недель, сроки практики

Руководитель практики от университета

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. ст., уч. зв. |  | личная подпись, дата |  | инициалы, фамилия |

Руководитель практики от организации (учреждения, предприятия)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. ст., уч. зв. |  | личная подпись, дата |  | инициалы, фамилия |

Тема практики: «Трёхслойная архитектура»

Оглавление

[Введение 4](#_Toc526899819)

[Трёхслойная архитектура 5](#_Toc526899820)

[Слой клиента 5](#_Toc526899821)

## Введение

Цель практики – изучение трёхслойной архитектуры и паттерна MVC на платформе .Net.

В ходе практики были изучены такие понятия, как:

1. уровень работы с базой данных;
2. уровень бизнес логики;
3. пользовательский интерфейс;
4. хранимые процедуры в MSSQL;
5. построение веб-приложения на основе MVC.

# Трёхслойная архитектура

Трёхслойная архитектура – это архитектурная модель программного комплекса, включающая следующие компоненты: клиент, сервер приложений, сервер базы данных.

К достоинствам можно отнести следующее:

1. масштабируемость;
2. конфигурируемость;
3. высокая безопасность;
4. высокая надёжность;
5. низкие требования к скорости канала между терминалами и сервером приложений;
6. низкие требования к производительности и техническим характеристикам терминалов, как следствие снижение их стоимости.

Из достоинств вытекают следующие недостатки:

1. более высокая сложность создания приложений;
2. сложнее в разворачивании и администрировании;
3. высокие требования к производительности серверов приложений и сервера базы данных, а, значит, и высокая стоимость серверного оборудования;
4. высокие требования к скорости канала между сервером базы данных и серверами приложений.

Сначала приложение создаётся как просто консольное приложение. Клиент является тем самым консольным приложением. Дальнейшие компоненты добавляются как библиотеки классов.

## Слой клиента

Этот компонент отвечает за взаимосвязь с пользователем, то есть за интерфейс, который пользователь будет видеть на экране компьютера или телефона.

В программе, которая была реализована во время летней практики, этот компонент называется PL (Presentation Layer). Интерфейс представляет собой консоль, в которой пользователь может выбирать определённые действия, такие как добавление пользователя, медали или награждение пользователя определённой наградой:

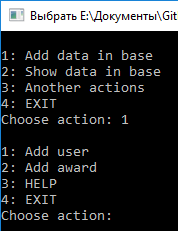


Рисунок 1 - Пример консольного интерфейса

Слой клиента взаимодействует с бизнес логикой, используя слабое связывание, то есть через интерфейс.

Добавить ещё что-нибудь про вызов методов.

## Слой бизнес логики

Этот компонент отвечает за логику самого приложения, то есть обработку и выдачу данных в нужном формате.

В программе, которая была реализована во время летней практики, этот компонент называется BLL (Business Logic Layer). Слой представляет из себя набор методов для обработки входящих данных. Например, метод UpdateUser(string id, string name, string birthday), который обновляет данные пользователя и возвращает либо true, либо false, сообщая о том, прошла операция или нет. На вход метод получает номер пользователя – id, новое имя пользователя – name и новую дату рождения – birthday. Внутри данные преобразуются в нужный формат для работы с базой данных. После обработки данных метод передаёт параметры другому методу UpdateUser, расположенный в слое базы данных. Если данные введены пользователем неправильно, то метод возвращает false и сообщение об ошибке. В параметры не входит возраст пользователя, так как он высчитывает с помощью отдельного приватного метода SetAge(DateTime birthday), который возвращает целочисленное значение, то есть возраст пользователя.

…..

public int SetAge(DateTime birthday)

{

if (DateTime.Today.Month > birthday.Month)

{

return (DateTime.Today.Year - birthday.Year);

}

else

{

return (DateTime.Today.Year - birthday.Year) - 1;

}

}

…..

…..

public bool UpdateUser(string id, string name, string birthday)

{

DateTime dateTime;

int userId;

if (DateTime.TryParse(birthday, out dateTime) && (Int32.TryParse(id, out userId)))

{

if (GetUserById(userId) != null)

{

\_userDao.UpdateUser(userId, name, dateTime, SetAge(dateTime));

return true;

}

else

{

Console.WriteLine($"DB has no information");

return false;

}

}

else

{

Console.WriteLine($"Incorrect id or birthday");

return false;

}

}

…..

Слой бизнес логики взаимодействует со слоем базы данных, используя слабое связывание, то есть интерфейс. В данном случае используется интерфейс слоя базы IUserDao, в котором перечислены методы для работы с базой данных. Для взаимодействия со слоем базы данных через интерфейс в конструкторе инициализируется переменная \_userDao.

…..

private readonly IUserDao \_userDao;

public UserLogic(IUserDao userDao)

{

\_userDao = userDao;

}

…..

## Слой базы данных

Этот компонент отвечает за работу с самой базой данных, которая хранит все себе данные о пользователях и наградах.

В программе, которая была реализована во время летней практики, этот компонент называется DAL (Data Access Layer). Слой представляет собой набор методов для работы с базой данных. Например, метод UpdateUser(int wantedId, string wantedName, DateTime wantedBirthday, int wantedAge). С помощью SqlConnection осуществляется подключение к базе данных. Далее, используя метод CreateCommand(), указывается используемая хранимая процедура. В данном случае используется хранимая процедура под именем UpdateUser. После определяются параметры для хранимой процедуры: @ID, @NAME, @BIRTHDAY, @AGE, которые соотносятся с входящими параметрами соответственно. С помощью метода Open() открывается подключение к базе данных, а метод ExecuteNonQuery() выполняет процедуру, то есть обновление пользователя в базе данных.

…..

public int UpdateUser(int wantedId, string wantedName, DateTime wantedBirthday,

int wantedAge)

{

using (var connection = new SqlConnection(\_connectionString))

{

var command = connection.CreateCommand();

command.CommandType = System.Data.CommandType.StoredProcedure;

command.CommandText = "UpdateUser";

var id = new SqlParameter("@ID", SqlDbType.Int)

{

Value = wantedId

};

command.Parameters.Add(id);

var name = new SqlParameter("@NAME", SqlDbType.VarChar)

{

Value = wantedName

};

command.Parameters.Add(name);

var birthday = new SqlParameter("@BIRTHDAY", SqlDbType.DateTime)

{

Value = wantedBirthday

};

command.Parameters.Add(birthday);

var age = new SqlParameter("@AGE", SqlDbType.Int)

{

Value = wantedAge

};

command.Parameters.Add(age);

connection.Open();

return (int)(decimal)command.ExecuteNonQuery();

}

}

…..