Частное учреждение образование

«Колледж бизнеса и права»

КУРСОВОЙ ПРОЕКТ

ПО РАЗРАБОТКЕ И СОПРОВОЖДЕНИЮ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ОП Т.693011.401

Выполнил К.В. Максимук

Руководитель курсовой Н.В. Ржеутская

2019

# Объектно-ориентированный анализ и проектирование системы

### Сущность задачи

Основной задачей автоматизируемой предметной области является автоматизация учета проданных дверей и расчёт стоимости выбранной двери.

Необходимо разработать программное приложение, используя язык объектно-ориентированного программирования C#.

В приложении должны быть реализованы следующие модули:

* ввод информации с клавиатуры;
* вывод информации в виде отчетов;
* сохранение информации в базе данных;
* вывод информации из базы данных;
* поиск по выбранному параметру:
* удаление из базы заданной информации.

Также должен быть предусмотрен вывод сообщений об ошибках, в случае введения неверных команд или данных.

### Проектирование модели

Главной целью проектирования моделей является отображение функциональной структуры объекта, то есть производимые ими действия и связи между этими действиями. Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD), которые предназначены для графического представления моделей данных разрабатываемой программной системы и предлагают некоторый набор стандартных обозначений для определения данных и отношений между ними. С помощью этого вида диаграмм можно описать отдельные компоненты концептуальной модели данных и совокупность взаимосвязей между ними, имеющих важное значение для разрабатываемой системы. Основными понятиями данной нотации являются понятия сущности и связи. При этом под сущностью понимается произвольное множество реальных или абстрактных объектов, каждый из которых обладает одинаковыми свойствами и характеристиками. В этом случае каждый рассматриваемый объект может являться экземпляром одной и только одной сущности, должен иметь уникальное имя или идентификатор, а также отличаться от других экземпляров данной сущности. Связь определяется как отношение или некоторая ассоциация между отдельными сущностями. Примерами связей могут являться родственные отношения типа «отец-сын» или производственные отношения типа «начальник-подчиненный». Другой тип связей задается отношениями «иметь в собственности» или «обладать свойством».

Графическая модель данных строится таким образом, чтобы связи между отдельными сущностями отражали не только семантический характер соответствующего отношения, но и дополнительные аспекты обязательности связей, а также кратность участвующих в данных отношениях экземпляров сущностей. Информационная модель базы представлена на диаграмме «Сущность-связь». Данная диаграмма представлена на рисунке 1.1.

Рисунок 1.1 - Диаграмма «сущность-связь» в нотации Баркера

Добавить сущность «Пользователь» и «Клиент»

Переделать связи как в БД

Рисунок 1.2 - Диаграмма ERD в нотации Ричарда Чена

Исходя из исследования предметной области, можно выделить следующие сущности разработки: квартира, договор, покупатель.

У сущности «» нет выделяемых атрибутов

Для сущности «» можно выделить следующие атрибуты:

Для сущности «» можно выделить следующие атрибуты:

# Вычислительная система

### Требования к аппаратным и операционным ресурсам

Основными минимальными требованиями, выдвигаемыми к аппаратному обеспечению персонального компьютера, являются:

* процессор 800 МГц и выше;
* оперативная память 128 Мбайт и более;
* свободное место на диске 100 Мбайт;
* интегрированная видеокарта на 512 Мбайт и более;
* монитор;
* мышь, клавиатура;
* принтер.

Компьютер должен работать под управлением операционной системы, начиная с Windows 7 и выше. Наиболее удобной операционной системой для проведения испытаний является Windows 10, так как она ориентированна на максимальное использование всех возможностей ПК, сетевых ресурсов и обеспечение комфортных условий работы.

### Инструменты разработки

Инструментами разработки являются:

* операционная система Windows 10;
* система управления данными MS SQL Server;
* среда программирования Microsoft Visual Studio;
* язык программирования C#.
* Umlet
* MS Office
* Справка

Visual Studio меняет отношение к процессу разработки, делая его увлекательным и хорошо организованным. Новый продукт отличается повышенной скоростью загрузки рабочей среды и открывает разработчикам доступ к конкретным проектам буквально в считанные секунды. Кроме того, все длительные процессы выполняются в Visual Studio в фоновом режиме, что не замедляет скорость работы среды и не отвлекает разработчика от основных задач.

Также рабочая среда Visual Studio имеет новый контекстно-зависимый интерфейс. Главная его особенность заключается в том, что он предлагает разработчику только те функции и инструменты, которые ему нужны на данном этапе работы. Таким образом, панель инструментов не содержит ничего лишнего и не затрудняет поиск нужных функций.

Visual Studio позволяет эффективно управлять полным жизненным циклом приложения от этапа его разработки до стадии эксплуатации. Такой подход предполагает командную работу и участие в процессе большого количества специалистов разного профиля: от архитекторов и разработчиков до дизайнеров и заказчиков проекта.

Консолидацию всех циклов работы над приложением и взаимодействие рабочей группы в Visual Studio обеспечивает усовершенствованное решение Team Foundation Server. С его помощью все участники процесса разработки могут отслеживать состояние проекта, видеть его динамику, контролировать сроки и получать аналитические отчёты о каждом периоде работы.

Кроме того, Visual Studio содержит обновлённые инструменты проверки качества и работоспособности приложения, что позволяет тестировщикам моделировать поведение приложения в момент его использования, а также вовремя обнаруживать недочёты в разработке. А функция PowerPoint StoryBoarding позволяет техническим специалистам представлять макет будущего решения заказчику в понятном для него формате.

С# — это язык программирования, синтаксис которого очень похож на синтаксис Java (но не идентичен ему). Например, в С# (как в Java) определение класса состоит из одного файла (\*.cs), в отличие от C++, где определение класса разбито на заголовок (\*.h) и реализацию (\*.срр). Однако называть С# клоном Java было бы неверно. Как С#, так и Java основаны на синтаксических конструкциях C++. Если Java во многих отношениях можно назвать очищенной версией C++, то С# можно охарактеризовать как очищенную версию Java.

Синтаксические конструкции С# унаследованы не только от C++, но и от Visual Basic. Например, в С#, как и в Visual Basic, используются свойства классов как C++, С# позволяет производить перегрузку операторов для созданных вами типов (Java не поддерживает ни ту, ни другую возможность). С# — это фактически гибрид разных языков. При этом С# синтаксически не менее (если не более) чист, чем Java, так же прост, как Visual Basic, и обладает практически той же мощью и гибкостью, что и C++.

SQL Server является надежной базой данных для любых целей, может продолжать расширяться по мере наполнения информацией, без заметного уменьшения быстродействия операций с записями в многопользовательском режиме. Пользователи могут быть добавлены путем модернизации оборудования. В последнем тесте поддерживалось до 4600 пользователей базы данных.

Обеспечивается максимальная безопасность. Данные защищены от несанкционированного доступа за счет интеграции сетевой безопасности с сервером безопасности. Поскольку безопасность на уровне пользователя, пользователи могут иметь ограниченный доступ к записи данных, тем самым защищая их от модификации или поиска, указав доступ на уровне пользовательских привилегией. Кроме того, с данными, хранящимися на отдельном сервере, сервер работает как шлюз, который ограничивает несанкционированный доступ.

SQL Server обрабатывает запросы от пользователей и только отправляет пользователю результаты запроса. Таким образом, минимальная информация передается по сети. Это улучшает время отклика и устраняет узкие места в сети. Это также позволяет использовать SQL Server в качестве идеальной базы данных для сети Интернет.

Техническое обслуживание SQL Server очень простое и не требует больших знаний. Возможны изменения в структуре данных, а также резервное копирование во время работы сервера, без остановки.

# Объектно-ориентированный анализ и проектирование системы

### Требования к приложению

Для того, чтобы к базе данных могли получать доступ сразу несколько пользователей, разумно разделить собственно базу данных (она будет храниться в базе MS SQL Server, и может храниться на сервере) и интерфейс пользователя (представлен при помощи форм).

Приложение должно обеспечивать следующие функции;

* редактирование, добавление и удаление данных о дверях;
* поиск двери;
* калькулятор расчета стоимости покупки и установки двери;
* пользователям системы должен быть предоставлен простой и интуитивно-понятный интерфейс;
* возможность для различных пользователей совместного доступа к базе данных;
* разграничение прав доступа к различным областям базы данных при помощи системы паролей;
* справка.

Одной из главных деталей программного продукта является пользовательский интерфейс. Он должен быть интуитивно понятным и максимально удобным для всех пользователей программы, а тем более для новичков в работе с компьютером. Кроме этого, он должен быть выполнен в одной цветовой гамме в спокойных тонах. Все формы данного программного продукта будут выполнены в одном стиле.

### Концептуальный прототип

При запуске программа отображает главное меню. Дальнейшие действия программы будут зависеть от выбора пользователя. Меню будет состоять из следующих пунктов:

* «Профили»;
* «Дополнительные услуги»;
* «Учет заказов»;
* «Оформить заказ»;
* «Калькулятор стоимости»;
* «Справка».

При выборе пункта меню «Справка» вызывается инструкция по эксплуатации программы.

При выборе любого пункта подменю на экран выводится соответствующая форма.

### Организация данных

Для определения структуры данных представим схему, которая содержит основные объекты и показывает взаимосвязь этих объектов и задач пользователя. В качестве модели данных системы управления базами данных MS SQL Server используется реляционная модель, так как она отображает объединение выявленных при анализе сущностей в схему, объединяющую упорядоченные данные определенными отношениями.

Для логического размещения данных в реляционной базе используются таблицы, созданные в приложении MS SQL Server.

В ходе разработки программного продукта были созданы следующие таблицы:

Таблица 3.1 - «Profili»

|  |  |  |  |
| --- | --- | --- | --- |
| ***Имя поля*** | ***Тип Данных*** | ***Формат поля/размер поля*** | ***Обязательное поле*** |
| id\_profil | Счётчик | Длинное целое | Да |
| profil | Текстовый | 50 | Нет |
| cena | Числовой | Вещественный | Нет |

Таблица 3.2 - «Dopolnitelno»

|  |  |  |  |
| --- | --- | --- | --- |
| ***Имя поля*** | ***Тип данных*** | ***Формат поля/размер поля*** | ***Обязательное поле*** |
| id\_dop | Счётчик | Длинное целое | Да |
| cena\_ustanovki | Числовой | Вещественный | Нет |
| cena\_nalichniki | Числовой | Вещественный | Нет |
| cena\_zamki | Числовой | Вещественный | Нет |
| cena\_ruchka | Числовой | Вещественный | Нет |
| cena\_petli | Числовой | Вещественный | Нет |

Таблица 3.3 - «Zakazy»

|  |  |  |  |
| --- | --- | --- | --- |
| ***Имя поля*** | ***Тип Данных*** | ***Формат поля/размер поля*** | ***Обязательное поле*** |
| id\_zakaz | Счётчик | Длинное целое | Да |
| data | Дата | - | Нет |
| id\_profil | Числовой | Длинное целое | Нет |
| vysota | Числовой | Длинное целое | Нет |
| shirina | Числовой | Длинное целое | Нет |
| kolvo | Числовой | Длинное целое | Нет |
| ustanovka | Логический | Да/Нет | Нет |
| nalichniki | Логический | Да/Нет | Нет |
| zamok | Логический | Да/Нет | Нет |
| ruchka | Логический | Да/Нет | Нет |
| petli | Логический | Да/Нет | Нет |
| id\_dop | Числовой | Длинное целое | Нет |

Схема данных приведена на рисунке 3.1.

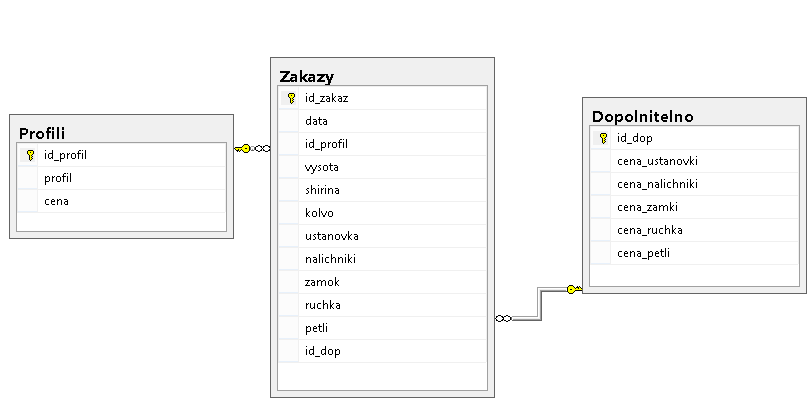


Рисунок 3.1 - Схема данных

# Список литературы

1. Иванов, А.А. Автоматизация технологических процессов и производств: Учебное пособие / А.А. Иванов. - М.: Форум, 2012. - 224 c.
2. Клюев, А.С. Наладка средств автоматизации и автоматических систем регулирования: Справочное пособие / А.С. Клюев, А.Т. Лебедев, С.А. Клюев. - М.: Альянс, 2009. - 368 c.
3. Пантелеев, В.Н. Основы автоматизации производства: Учебник для учреждений начального профессионального образования / В.Н. Пантелеев, В.М. Прошин. - М.: ИЦ Академия, 2013. - 208 c.
4. Лесневский, А. С. Объектно-ориентированное программирование: Бином. Лаб. знаний / Лесневский А. С. — М.: Бином. Лаб. знаний, 2010. — 232с
5. Балдин, К.В. Информационные технологии в управлении предприятием: Учеб. для студ. учреждений высш. проф. образования / К.В. Балдин. - М.: ИЦ Академия, 2012. - 288 c.
6. Венделева, М.А. Информационные технологии в управлении: Учебное пособие для бакалавров / М.А. Венделева, Ю.В. Вертакова. - М.: Юрайт, 2013. - 462 c.
7. Баронов, В.В. Информационные технологии и управление предприятием. – М: Компания АйТи, 2006. – 328с.
8. Автоматизированные информационные технологии в управлении предприятием: учебник / Под ред. И.Т.Трубилина.- М.: Финансы и статистика, 2001.- 416с.
9. Макаров, А.С. Работа с базами данных в С# / ДМК Пресс. – Минск, 2013.
10. Вычислительные системы, сети и телекоммуникации: Учебник. – 2-е изд., перераб. и доп. / Под ред. А.П. Пятибратова. – М.: Финансы и статистика, 2001.
11. Карминский, М.А., Нестеров, П.В. Информатизация бизнеса. – М.: Финансы и статистика, 1997.
12. Дик, В. В.Банковские информационные системы / В. В. Дик. – М.: Маркет ДС, 2010. – 816 с.
13. Зверев, В. С.Информационные системы: учебник / В. С. Зверев, В. Р. Банк. – М: ЭКОНО­МИСТЪ, 2008. – 477 с
14. Багласова Т.Г. Методические указания по оформлению курсовых и дипломных проектов / Т.Г.Багласова, К.О.Якимович. – Минск: КБП, 2013.



Масса

Лит.

Масштаб

Н.В.Ржеутская

Лист 1

Реценз.

Т. Контр.

*Программа автоматизации расчета стоимости установки дверей*

Диаграмма вариантов использования

Утверд.

Н. Контр.

К.В.Максимук

Разраб.

Провер.

*КП Т.693.401 ГЧ*

Инв.№подл.

КБиП

Взам.инв.№

Подп. и дата

Инв.№дубл.

Подп. и дата

У

Листов 11