



Курсовая работа по .NET Framework

Методические рекомендации к выполнению курсового проекта по .NET

Введение

В рамках процесса обучения в Компьютерной Академии «Шаг» студенты направления «Разработка программного обеспечения», кроме экзаменов по предметам, также сдают курсовые проекты. Давайте зададимся вопросом: почему в программе обучения есть курсовые? Есть несколько ответов на этот вопрос. Первый ответ: студентам важно поработать над комплексным проектом, объединяющим набор технологий. Например, если мы говорим о курсовом по .NET Framework, то в рамках проекта можно и нужно использовать максимальный спектр технологий .NET Framework. Второй ответ: студентам по окончании обучения необходимо иметь своё портфолио и курсовые работы отлично помогают в этом.

Утверждение темы курсовой работы

Тема курсовой работы студента должна быть утверждена заведующим кафедрой. Студент описывает тему работы и список возможностей курсового проекта в заявке объемом не более двух листов. На первом шаге полученную заявку утверждает преподаватель, отвечающий за реализацию курсового проекта. По курсовому .NET Framework, это тот преподаватель, который вел блок предметов .NET Framework. На втором шаге заявку студента получает менеджер учебного процесса конкретного направления. Заявка оформляется в двух экземплярах. Один экземпляр остается у студента, второй – у менеджера учебного процесса. Курсовой проект может быть выполнен одним студентом или командой студентов. В командном проекте, объём работы каждого студента должен быть равным объёму курсовой для одного студента. Работа над курсовой работой Студент выполняет курсовую работу дома, вне стен Академии. Если у студента возникают вопросы, то в процессе работы он задаёт их преподавателю. Для стационара выделено время на консультации и защиту курсового проекта.

Защита курсового проекта

На защите должны присутствовать менеджер учебного процесса направления (по возможности), преподаватель, отвечающий за курсовой студента, заведующий кафедрой. Заведующий кафедрой приходит на защиту с заявками студентов, которые хранились у менеджера учебного процесса.

Студент на защиту готовит распечатанный документ, в состав которого входит:

1. Титульный лист с названием темы, названием курсового проекта, ФИО студента;
2. Копия заявки, которая осталась у студента;
3. UML-диаграммы проекта. Обязательным условием является наличие диаграммы классов, диаграмма вариантов использования (use case diagram). Другие диаграммы прикладываются по желанию.
4. Схема базы данных (если в проекте есть база данных);

5. Исходный код курсового проекта.

Защита проводится в формате презентации. Студент рассказывает о своей работе перед комиссией. Использование презентаций в формате PowerPoint и других средств приветствуется. В рамках презентации студент должен рассказать о своем проекте, почему была выбрана данная тема, осветить наличие конкурентов, обосновать выбранные технологии и т.д. После рассказа о своей работе студент должен продемонстрировать работу проекта на практике.

Комиссия может задавать абсолютно любые вопросы по проекту, в рамках процесса защиты. Если комиссия захочет задать вопрос по исходному коду, студент обязан продемонстрировать место в коде, которое интересует комиссию. Максимальная длительность выступления студента 15 минут, рекомендуемое время для вопросов комиссии 15 минут. Итого: общее время работы с конкретным студентом на защите 30 минут. После выступления всех студентов комиссия устраивает коллегиальное обсуждение оценок (без присутствия студентов). Оценка конкретному студенту выставляется при достижении коллегиального согласия или большинства. После выставления всех оценок, заведующий кафедрой объявляет их перед студентами сразу после финального совещания. Студенты, не сдавшие курсовой или желающие получить более высокую оценку, отправляются на пересдачу. Дату пересдачи назначает менеджер учебного процесса после консультаций с преподавателями и заведующим кафедрой.

Общие требования.

1. Неработающая программа не проверяется и к защите не допускается!!!
2. Программа без пояснительной записки к защите не допускается.
3. Для защиты курсовой нужно подготовить краткое описание выбранной темы и программы (максимум 10 минут). Ошибка выполнения во время защиты – -1 балл к оценке.
4. У программы должен быть интуитивно понятный эргономичный пользовательский интерфейс. Его отсутствие снижает оценку.
5. Во всех программах обязательно должны присутствовать классы, наследование и база данных.
6. Каждая программа должна иметь более одной формы.
7. Перед запуском программа запрашивает имя сервера, имя базы, логин и пароль пользователя.
8. Каждая программа должна содержать главное и контекстное меню.

Требование к программной документации

Все курсовые работы должны содержать следующие разделы:

1. Введение (описание необходимости разрабатываемой системы)
2. Постановка задачи (включает инженерную, математическую постановку задачи)
3. Требования к функциональным характеристикам системы (что система должна делать)
4. Требования к техническим характеристикам (описать необходимые технические средства для функционирования системы)
5. Требования к программным характеристикам (описать необходимые программные средства для функционирования системы)
6. Проектирование системы
 1. Проектирование интерфейса (описание правил и способов создания интерфейса, применяемых в системе)
 2. Проектирование диаграммы классов (описание классов, связей, примененных паттернов)
 3. Проектирование базы данных (описание структуры файла).

4. Проектирование структур системы (описание всех структур системы для хранения данных в упорядоченном виде: списки, стеки, очереди и т.д.)
5. Проектирование алгоритмов (представляется описание в виде блок-схем или UML “диаграмм действий” основных узлов системы)
7. Тестирование и отладка программы (описание примененной методики тестирования (черный, белый ящик), представление набора тестов)
8. Руководство пользователя (описание последовательности действий выполняемых пользователем)
9. Приложения
10. Список литературы

Темы:

1. Создать приложение «Книжный магазин».

Суть программы:

Основная задача проекта: учитывать текущий ассортимент книг в магазине.

Требования:

- Необходимо хранить следующую информацию о книгах: название книги, ФИО автора, название издательства, количество страниц, жанр, год издания, себестоимость, цена для продажи, является ли книга продолжением какой-то другой книги (например, вторая часть дилогии).
- Приложение должно позволять: добавлять книги, удалять книги, редактировать параметры книг, продавать книги, списывать книги, вносить книги в акции (например, неделя книг новогодней тематики со скидкой 10%), откладывать книги для конкретного покупателя.
- Приложение должно предоставить функциональность по поиску книг по таким параметрам: название книги, автор, жанр.
- Приложение должно предоставлять возможность просмотреть список новинок, список самых продаваемых книг, список самых популярных авторов, список самых популярных жанров по итогам дня, недели, месяца, года.
- Необходимо предусмотреть возможность входа по логину и паролю.

2. Создать приложение «Музыкальный магазин».

Суть программы:

Основная задача проекта: учитывать текущий ассортимент музыкальных пластинок в магазине.

Требования:

- Необходимо хранить следующую информацию о пластинках: название пластинки, название коллектива, название издателя, количество треков, жанр, год издания, себестоимость, цена для продажи.
- Приложение должно позволять: добавлять пластинки, удалять пластинки, редактировать параметры пластинок, продавать пластинки, списывать пластинки, вносить пластинки в акции (например, неделя пластинок джаза со скидкой 10%), откладывать диски для конкретного покупателя.
- Приложение должно предоставить функциональность по поиску дисков по таким параметрам: название диска, исполнитель, жанр.

- Приложение должно предоставлять возможность просмотреть список новинок, список самых продаваемых пластинок, список самых популярных авторов, список самых популярных жанров по итогам дня, недели, месяца, года.
- Необходимо предусмотреть возможность входа по логину и паролю.
- Также нужно сделать возможность регистрации постоянных покупателей и создать систему скидок в зависимости от накопленной суммы потраченных средств.

3. Создать приложение «Олимпиада».

Суть программы:

Основная задача проекта: учитывать информацию о результатах, показанных на той или иной олимпиаде.

Требования:

- Необходимо хранить такие данные: год проведения олимпиады, летняя или зимняя олимпиада, название страны хозяйки олимпиады, город, в котором проводилась олимпиада, название видов спорта, информацию по каждому виду спорта (количество участников, участники, результаты и т. д.), информацию об участниках (ФИО, страна, вид спорта, дата рождения, фото спортсмена и т. д.).
- Приложение должно позволять добавлять информацию, удалять информацию, редактировать информацию.
- Приложение должно разрешать пользователю отображать:
 - таблицу медального зачета по странам по конкретной олимпиаде, за всю историю олимпиад;
 - медалистов по разным видам спорта по конкретной олимпиаде, за всю историю олимпиад;
 - страну, которая собрала больше всего золотых медалей на конкретной олимпиаде, за всю историю олимпиад;
 - спортсмена, который выиграл больше всего золотых медалей в конкретном виде спорта;
 - название страны, которая чаще всех была хозяйкой олимпиады;
 - состав олимпийской команды спортсменов конкретной страны;
 - статистику выступления конкретной страны на конкретной олимпиаде, за всю историю олимпиад.

4. Реализовать приложение, позволяющее искать некоторый набор запрещенных слов в файлах.

Требования:

- Пользовательский интерфейс приложения должен позволять ввести или загрузить из файла набор запрещенных слов.
- При нажатии на кнопку «Старт», приложение должно начать искать эти слова на всех доступных накопителях информации (жесткие диски, флешки).
- Файлы, содержащие запрещенные слова, должны быть скопированы в заданную папку.
- Кроме оригинального файла, нужно создать новый файл с содержимым оригинального файла, в котором запрещенные слова заменены на 7 повторяющихся звезд (*****).
- Также нужно создать файл отчета. Он должен содержать информацию о всех найденных файлах с запрещенными словами, пути к этим файлам, размер файлов, информацию о количестве замен и так далее.

- В файле отчета нужно также отобразить топ-10 самых популярных запрещенных слов.
- Интерфейс программы должен показывать прогресс работы приложения с помощью индикаторов (progress bars).
- Пользователь через интерфейс приложения может приостановить работу алгоритма, возобновить, полностью остановить.
- По итогам работы программы необходимо вывести результаты работы в элементы пользовательского интерфейса (нужно продумать, какие элементы управления понадобятся).
- Программа обязательно должна использовать механизмы многопоточности и синхронизации! Программа может быть запущена только в одной копии.
- Предусмотреть возможность запуска приложения из командной строки без отображения визуального интерфейса.

5. Программа для формирования экзаменационных билетов.

Суть программы:

Есть база вопросов. Вопросы разбиты по категориям. Программа выбирает случайным образом вопросы и выводит их на форму. Программа должна также позволять добавлять, удалять и изменять вопросы.

Требования:

- Каждый вопрос состоит из названия дисциплины, текста вопроса и картинки.
- Поле картинки в вопросе может быть пустым.
- В базе хранится путь к картинке.
- При добавлении вопроса, выбирается категория, вводится название дисциплины, текст вопроса и путь к картинке, при этом картинка копируется в специальную папку, а в базу заносится путь к этому файлу.
- Программа должна позволять редактировать, добавлять и удалять категории.
- Добавление и редактирование выполняется в отдельных окнах.
- При выборе вопросов, категории не должны повторяться.
- Количество выбираемых вопросов вводит (выбирает) пользователь.
- Перед началом генерации вопросов, программа должна запросить данные студента (ФИО, группа).
- Результаты выбора вопросов выводятся на панель (номер вопроса, название дисциплины, текст вопроса, рисунок). Элементы управления при этом генерируются динамически (если рисунка в вопросе нет, элемент управления для него не создается).
- Для отображения вопросов выбирается категория, все вопросы выводятся на панель, как указано пунктом выше, но при этом под каждым две кнопки (удалить, добавить).
- Программа должна также предусматривать поиск вопросов по тексту, выводить в формате указанном в предыдущем пункте.

Рекомендации:

Вопросы разных категорий лучше хранить в разных таблицах.

При поиске использовать шаблон like.

6. Студенческое общежитие

Суть программы:

Основная задача проекта: разработать приложение для учета информации о студентах, проживающих в общежитии (ВУЗа).

Требования:

- Информация о студентах (ФИО, пол, адрес, телефон, дата рождения, факультет, курс, группа, фотография);
- Информация о месте проживания(комната) (номер комнаты, сведения о студентах, закрепленная мебель с инвентарными номерами);
- Информация о родителях (ФИО, статус родства, адрес, телефон, место работы);
- Сведений о нарушении дисциплины в общежитии (сведения о нарушителях: студент, номер комнаты, дата происшествия и т. д.);
- На выходе пользователи системы могут получать следующую информацию: сортированный список студентов общежития, курс, факультет и т.д.; № комнаты, закрепленная мебель с инвентарными номерами; информация о нарушении дисциплины с описанием (нарушитель, № комнаты и дата); поиск информации о родителях по данным студента; количество студентов общежития по группам.

7. Проектирование ландшафтных дизайнов

Суть программы:

В базе располагается список всех растений и их характеристики. Пользователь проектирует обустройство своего участка, выбирая растения из списка и перемещая его на участок.

Программа не предусматривает редактирование базы. Программа должна сохранять полученный дизайн в файл.

Требования:

- Все растения разбиты на категории.
- В базе содержатся также описание основных строений (дом, гараж, сарай и т.д.).
- Каждое растение содержит категорию, название, описание, занимаемая площадь (ширина, высота), рисунок. В базе храниться путь к картинке, находящейся в определенной папке.
- При запуске программы пользователь вводит размеры участка (ширина, высота).
- При добавлении пользователем растения оно отображается на участке с учетом масштаба.
- При добавлении строения пользователь может указать его размеры и оно отображается с учетом масштаба.
- При наведении мышкой на соответствующий участок в строке состояния отображается описание растения находящегося на этом участке.
- Программа должна позволять сохранение в файл и чтение из файла

Рекомендации:

Необходимо предварительно заполнить базу.

Для отображения картинок можно использовать элементы управления PictureBox, которые генерировать во время выполнения программы.

8. Тестирующая система

Суть программы:

База данных содержит тесты. Пользователь может выбрать тест из определенной категории, пройти его и увидеть результат. Программа также позволяет редактировать тесты, но только пользователю с определенными правами.

Требования:

- Тест может иметь сколько угодно вопросов, каждый вопрос может иметь сколько угодно вариантов ответа. При этом один из вариантов ответа помечается как правильный.
- База данных содержит двух пользователей (teacher, student).
- Пользователю teacher доступны окна для редактирования вопросов и добавления новых тестов.
- Пользователю student доступны окна для сдачи теста.
- Программа должна выводить список всех доступных тестов (под пользователем student).
- После прохождения теста система выдает оценку в процентном соотношении.
- Для отображения вопросов выбирается дисциплина, все вопросы выводятся на панель (номер, вопрос, варианты ответа) при этом под каждым две кнопки (удалить, добавить). Такие же кнопки под каждым вариантом ответа (под пользователем teacher).
- Добавление и редактирование выполняется в отдельных окнах.

Рекомендации:

Создайте в базе данных таблицу, содержащую названия всех тестов.
Создавайте элементы управления динамически для отображения вопросов.

9. Программа для составления психологического портрета

Суть программы:

База данных содержит тесты. Пользователю может пройти набор тестов из определенной категории и увидеть результат. Программа также позволяет редактировать тесты, но только пользователю с определенными правами.

Требования:

- Каждый тест может иметь сколько угодно вопросов, каждый вопрос может иметь сколько угодно вариантов ответа. Каждый вариант ответа имеет определенное количество баллов.
- База данных содержит результаты тестов в виде текстового описания, картинки и диапазона баллов.
- В базе данных храниться путь к картинке.
- При добавлении ответа, вводится путь к картинке и она копируется в специальную папку.
- База данных содержит двух пользователей (admin, user).
- Пользователю admin доступны окна для редактирования вопросов и добавления новых тестов.

- Пользователю user доступны окна для сдачи теста.
- После прохождения теста система выдает оценку в виде текстового описания психологического портрета. Оценка определяется исходя из суммы баллов.
- Для отображения вопросов выбирается категория, название теста, все вопросы выводятся на панель (номер, вопрос, варианты ответа) при этом под каждым две кнопки (удалить, добавить). Такие же кнопки под каждым вариантом ответа (под пользователем admin).
- Добавление и редактирование выполняется в отдельных окнах.

Рекомендации:

Создайте в базе данных таблицу, содержащую названия всех тестов.

Создавайте элементы управления динамически для отображения вопросов.

10. База сотрудников

Суть программы:

Программа просматривает список сотрудников и список их сертификатов, а также просматривать информацию о начислениях по зарплате. Программа также позволяет редактировать сотрудников, но под определенными правами.

Требования:

- О сотруднике необходимо хранить следующую информацию (ФИО, пол, адрес, телефон, дата рождения, специальность, зарплата, фотография). Для каждого сотрудника также храниться список его сертификатов.
- Добавление и редактирование выполняется в отдельных окнах.
- База данных содержит двух пользователей – director и admin. Директор может только просматривать сотрудников и их сертификаты (просматривать соответствующие представления и/или выполнять соответствующие хранимые процедуры). Администратор может не только просматривать, но и редактировать сотрудников.
- Программа должна предусматривать поиск сотрудника по фамилии, имени, отчеству, полу, возрасту, специальности и по любой комбинации из всего выше перечисленного.
- При добавлении сотрудника вводится путь к фотографии, фотография копируется в специальную папку, а в базе храниться путь к этому файлу.
- Если при добавлении фотография не выбрана, то сотруднику присваивается стандартная фотография силуэта мужчины или женщины в зависимости от выбранного пола.
- Программа должна выдавать отчет о начислении зарплаты с итогами по каждой специальности и по всей фирме. Отчет выводится в ListView на отдельном немодальном окне.

Рекомендации:

При поиске использовать шаблон like.

Для просмотра информации по начислению зарплаты используйте rollup или cube.

11. Риелторская контора

Суть программы:

Покупать может забронировать квартиру, может купить квартиру, может отказать выкупать забронированную квартиру. Программа также ведет подсчет общей суммы проданных квартир. Программа также позволяет редактировать квартиры под определенными правами.

Требования:

- Каждая квартира описывается номером, площадью, количеством комнат, этажом, стоимостью.
- Каждую квартиру можно забронировать.
- Забронированную квартиру может купить только пользователь, который ее забронировал.
- Программа должна осуществлять поиск квартир по количеству комнат, площади, стоимости или комбинации из всего выше перечисленного.
- Список всех квартир или найденных квартир отображать в ListView.
- Программа должна иметь возможность просмотреть список всех проданных квартир, всех забронированных квартир и всех свободных квартир.
- Программа должна показывать статистику по проданным квартирам (комнат в квартире, сумма) с подсчетом общей суммы всех проданных квартир.
- База данных содержит двух пользователей – риелтор и администратор. Риелтор может продавать квартиры, снимать/ставить бронь, просматривать списки квартир. Администратор может просматривать и редактировать квартиры.

Рекомендации:

Для просмотра информации по сумме продаж используйте rollup или cube.

12. Домашний бюджет

Суть программы:

Программа позволит отслеживать поступления и траты семьи, выяснять на какие нужды уходят деньги и экономить, убирая из бюджета не нужные расходы. Программа не предусматривает редактирование товаров и категорий расходов.

Требования:

- Программа содержит базу данных с развернутым перечнем товаров, разбитым на категории.
- Каждый месяц пользователь может открыть приход и расход.
- Каждая статья расходов содержит дату, сумму, категорию расходов, наименование товара, описание.
- Программа должна выдавать статистику расходов (категория, сумма) с общим итогом.
- Список расходов должен отображаться в элементе управления ListView.
- Программа должна предоставлять фильтр расходов по категориям.
- Пользователь может удалить или отредактировать статью расходов.
- Программа должна выводить статистику расходов в процентном соотношении (категория, проценты).
- Программа должна выполнять сравнение расходов по текущему месяцу с расходами по предыдущему месяцу.

Рекомендации:

Для просмотра информации по начислению зарплаты используйте rollup или cube.

13. Магазин компьютерных составляющих

Суть программы:

Покупатель может прийти в магазин и собрать себе компьютер из запчастей или выбрать один из предложенных вариантов собранного компьютера, а также купить отдельно любую составляющую к компьютеру. Программа будет учитывать продажу любого компонента или компьютера и выполнять подсчет собранных средств.

Требования:

- База данных содержит развернутое описание составляющих. Для каждой составляющей необходимо хранить (категорию, название, номер, стоимость, описание, количество).
- При покупке компьютера программа отображает список всех его составляющих.
- Список составляющих отображать с помощью ListView.
- Программа предусматривает поиск составляющих по категории, названию, номеру, цене или любой комбинации из всего выше перечисленного.
- Добавление и редактирование составляющих и компьютеров осуществляется в отдельных окнах.
- Программа выполняет подсчет всех средств за продажи определенного временного периода.
- Продавая товар, программа должна следить, что такой товар в нужном количестве присутствует на складе и уменьшать счетчик числа товаров.
- Продавая товар, покупателю выписывают чек, где указывается дата продажи, ФИО покупателя и проданные составляющие.

Рекомендации:

При поиске использовать шаблон like.

14. Программа для расчета стоимости металлопластикового окна

Суть программы:

Покупатель хочет заказать металлопластиковое окно и хочет заранее узнать его стоимость. Для оформления заказа необходимо замерить размеры окна и определиться с материалами (рама, стекла, фурнитура). Для учета форс-мажорных ситуаций после всех вычислений программа к полученной сумме прибавляет 10%. Программа предусматривает редактирование базы данных, но под определенными правами.

Требования:

- База данных содержит описание всех материалов (категория, название, цвет, стоимость за квадратный метр, описание).
- База данных содержит двух пользователей – admin и manager. Администратор может редактировать все таблицы, просматривать данные и выполнять хранимые процедуры (если такие имеются). Менеджер может только просматривать таблицы и выполнять хранимые процедуры.
- Программа запрашивает размеры окна (ширина, высота), количество секций, ширина рамы, размер подоконника, количество открывающихся секций, тип пластика и тип стекла и выполняет расчет стоимости.
- Добавление и редактирование осуществляется в отдельных окнах.
- Программа предусматривает просмотр отдельно списка материалов.
- Список материалов отображается в ListView.

- Реализовать поиск материала по названию, стоимости, категории или любой комбинации из выше перечисленного.

Рекомендации:

При поиске использовать шаблон like.

15. База данных менеджера

Суть программы:

Программа предназначена для отслеживания выполнения доставки продукции, своевременной оплаты продукции и учета вырученных средств. Программа будет также содержать информацию о предприятиях и список продукции, которую закупает каждое предприятие. Программа предусматривает редактирование продукции, но только под определенными правами.

Требования:

- База содержит информацию о предприятиях, контактные лица по каждому предприятию, их телефоны: рабочий, мобильный.
- База данных содержит список предприятий, список закупаемой каждым предприятием продукции, по каждому виду продукции список трех последних цен, предложенных нашим предприятием и условия оплаты и поставки (с доставкой или без), учетный вес (физический, моногидрат), если работаем с предприятием, то цена, по которой работаем, отмечена галочкой (цена за одну тонну).
- База данных должна фиксировать отгрузки: дата отгрузки, №цистерны, вес и общая стоимость (подсчитывается автоматически: вес * цену).
- База данных должна учитывать проплаты предприятия и даты проплаты, оставшийся долг и срок до которого нужно оплатить.
- Если срок оплаты предприятием превышен, то программа отображает долг красный и рядом количество просроченных дней.
- Если долг полностью погашен, то данная запись хранится в базе течение месяца, после чего удаляется по запросу пользователя
- База данных содержит двух пользователей – admin и manager. Администратор может редактировать все таблицы, просматривать данные и выполнять хранимые процедуры (если такие имеются). Менеджер не может редактировать товар
- Программа должна выводить статистику по товарам за определенный период (товар, сумма) с подсчетом общей суммы по всем товарам.
- Добавление и редактирование осуществляется в отдельных окнах.
- Программа предусматривает просмотр отдельно списка продукции.
- Список материалов отображается в ListView.

Рекомендации:

Для просмотра статистики по товарам используйте rollup или cube.

16. Программа учета пациентов для участкового терапевта

Суть программы:

Программа предназначена для учета пациентов и сданных ими анализов, для хранения архива пациентов. Программа также предоставляет удобный и быстрый поиск сданных анализов.

Требования:

- База данных содержит информацию о пациентах, на каждого пациента оформляется карточка.
- Пациент может получить больничный, направление на процедуры, направление на анализы и результаты анализов.
- В случае переезда пациента его карточка со всеми анализами перемещается в архив.
- Программа предусматривает поиск пациента по фамилии, адресу, номеру карточки или любой комбинации из всего выше перечисленного.
- Список пациентов отображается в ListView.
- Добавление и редактирование осуществляется в отдельных окнах.
- Для каждого пациента реализовать поиск анализов или больничных по названию анализа или дате сдачи.

Рекомендации:

При поиске использовать шаблон like.

17. Программа для компьютерного клуба.

Суть программы:

Программа предназначена для хранения информации об оборудовании компьютерного клуба, учета занятости компьютеров, ремонта компьютеров. Программа будет позволять заказывать рабочее место за ранее и на определенное время. Программа будет также определять наиболее доходное время работы и делать статистику посещаемости за месяц.

Требования:

- База данных содержит информацию о компьютерах, о занятых и свободных машинах, о времени заказа.
- Программа должна позволять заказать машину на определенное время.
- Программа должна учитывать ремонт машин.
- Программа должна осуществлять анализ посещаемости за месяц и определять наиболее доходное время работы.
- Программа должна определять минимальное и максимальное количество посетителей за определенный период.
- Список компьютеров отображается в ListView.
- Добавление и редактирование осуществляется в отдельных окнах.

Рекомендации:

Для просмотра статистики используйте rollup или cube.