

Лабораторная работа 1

Разработка программ на языке C# с графическим интерфейсом пользователя.

Цель работы:

- Изучить синтаксис языка C#.
- Изучить структуру класса в C#, создание и использование объектов класса в программе.
- Изучить приемы разработки Windows приложений в Visual Studio .Net.
- Освоить использование элементов графического интерфейса для работы приложения и построения иерархического меню.
- Изучить модель делегирования событий.

1. Для заданной структуры данных разработать абстрактный класс и класс наследник. В классе реализовать конструкторы. Создать методы, работающие с полями класса. Часть из них должны быть виртуальными. Добавить методы-свойства и индексаторы.

2. Разработать интерфейсные классы, добавляющие некоторые методы в класс-потомок. Изучить причины возникновения коллизий имен при наследовании и способы устранения.

3. Разработать классы исключительных ситуаций и применить их для обработки возникающих исключений.

4. Создать SDI-приложение (Single Document Interface) с элементами ввода и отображения полей класса. Для этого использовать различные элементы управления: текстовые поля, списки, независимые и радио- кнопки, а также панели.

5. Ввод новых данных и редактирование осуществлять через дополнительную диалоговую форму.

6. При изменении данных запрашивать подтверждение через окно диалога. В случае неполных данных или ошибок ввода сообщать об ошибке.

7. Объекты сохранять в коллекции.

8. Реализовать просмотр всей коллекции объектов через список. Для редактирования выбранного объекта создать дополнительную форму модального диалога.

9. Добавить на форму меню, позволяющее работать с пунктами: добавить, просмотреть, удалить, редактировать, справка.

10. Дублировать основные операции панелью инструментов.

Описания данных пользовательских типов:

3. ПЕЧАТНОЕ ИЗДАНИЕ →УЧЕБНИК: название, ФИО автора, стоимость, оглавление.

Лабораторная работа 2

Разработка многопоточных приложений

Цель работы:

- Изучить принцип разработки многопоточковых приложений на C# и способы управления потоками.

Задание

1. Разработать приложение с применением многопоточности.
2. В качестве объектов отображения добавить в коллекцию, созданную в 1 лабораторной работе объекты 2-х наследников вашего класса, добавив в них случайные координаты области отображения (x и y) и картинки для отображения (можно выбрать 2 любые картинки из интернета по заданию 1 лабораторной работы). Количество объектов – случайное число в диапазоне от 5 до 10 на каждый тип.
3. В программе обязательно должен реализовываться графический интерфейс, состоящий из двух частей: панель управления и панель отображения работы потоков.
4. Реализовать изменение приоритетов потоков.
5. Реализовать паузу в работе потоков и возобновление работы (через класс Monitor), запуск и полный останов осуществлять через меню и кнопки на панели управления.
6. Статусная строка в окне приложения должна отображать какой из потоков активен.

Вариант 3

1. 1-е объекты двигаются по оси X от одного края области обитания до другого со скоростью V.
2. 2-е объекты двигаются по оси Y от одного края области обитания до другого со скоростью V.

Программирование движения объекта

Для того чтобы запрограммировать движение объекта из одной точки в другую, вспомним равномерное движение.

Пусть $(x_1; y_1)$ – начальная координата объекта, а $(x_2; y_2)$ – конечная координата.

Направление движения задается вектором $(dx; dy) = \text{Нормализация}(x_2 - x_1; y_2 - y_1)$, где $\text{Нормализация}(a, b) = (a / \sqrt{a^2 + b^2}; b / \sqrt{a^2 + b^2})$.

Таким образом, координаты объекта в момент времени T от начала движения, движущегося со скоростью V по прямой, равны: $(x; y) = (x_1; y_1) + (dx; dy) * V * T$;

Скорость V измеряется в пиксель/сек, т.е. количество пикселей пройденных объектом на 1 сек.

Когда $(x; y)$ станет равно $(x_2; y_2)$, тогда объект дошел до конечной точки. На практике сравнивать текущие координаты с концом отрезка не корректно, так как промах на 1 пиксель из-за округления, например, отправит наш объект в бесконечное путешествие по прямой. Можно, например, прикинуть время, которое потребуется объекту, чтобы дойти до конца отрезка, поделив длину отрезка на скорость объекта. Условием продолжения движения будет время от начала движения T меньше времени, которое требуется, чтобы пройти отрезок.

Движение по окружности программируется иначе.

Пусть $(x_0; y_0)$ – центр окружности движения. Тогда координаты объекта в момент времени T от начала движения, движущегося со скоростью V по окружности, равны:

$$(x; y) = (x_0; y_0) + R * (\cos(V*T); \sin(V*T)),$$

где R – радиус окружности.

В этом случае в отличие от предыдущего V – угловая скорость, т.е. измеряется в радиан/сек.

Лабораторная работа 3

Разработка "клиент-серверных" приложений

Цель работы:

- Изучить принцип разработки "клиент-серверных" приложений на C#.

Задание

1. Разработать "клиент-серверное" приложение.
2. Создать отдельное консольное приложение, которое будет играть роль TCP-сервера. Клиентом будет приложение, которое было создано в предыдущих работах.
3. TCP-сервер должен ожидать подключения клиентов и подключении или отключении клиента рассылать список подключенных клиентов. Необходимо также внести изменения в интерфейс клиента, так чтобы в панели управления отображался список всех подключенных к серверу клиентов.
4. Запрограммировать специальное взаимодействие сервера по TCP протоколу с другими клиентами через сервер по варианту. Для действия необходимо предусмотреть элементы управления в интерфейсе.

Вариант 3

TCP: Реализовать возможность обмена всех объектов одного вида на объекты такого же вида из другого подключенного клиента.

Лабораторная работа 4

Разработка web-приложений на ASP .NET

Цель работы:

- Изучить принцип разработки "клиент-серверных" web-приложений на C#.

Задание

1. Разработать web-приложение.
2. Приложение должно иметь страницы, заполняющие данные объектов из 1-й лабораторной работы, которые обязательно проверяются на клиенте с помощью валидаторов.
3. Созданные объекты должны сохраняться в таблице базы данных.
4. По окончании ввода пользователь может просмотреть таблицу с введенными данными.
5. Реализовать возможность поиска данных по разным критериям, редактирование и удаление данных из базы данных.