

Департамент образования и науки Приморского края
Краевое государственное образовательное бюджетное учреждение
начального профессионального образования
«Профессиональный лицей № 16»

Программирование как один из этапов повышения профессионального мастерства.

(Сборник лабораторных работ)

Преподаватели:

Будякова Е.А.

Южанина Н.М.

Уссурийск

2011г.

Программирование как один из этапов повышения профессионального мастерства.

Будякова Е.А., преподаватель информатики

Южанина Н.М., преподаватель информатики

В данной работе представлены разработки лабораторных работ и их наглядное представление по темам «Объектно-ориентированное программирование», «Моделирование и формализация».

Материал, представленный в работе, призван помочь учителям и преподавателям глубже освоить новейшие информационные технологии, расширить их использование в повседневной работе с учащимися и при подготовке учебных материалов к урокам.

Будякова Е.А., Южанина Н.М., 2011

КГОБУ НПО «Профессиональный лицей №16» г. Уссурийск, 2011г.

Содержание.

Введение.....	4
1. Почему Visual Basic?.....	5
2. Язык объектно-ориентированного программирования Visual Basic.	6
3. Комплект лабораторных работ по теме «Алгоритмизация и объектно-ориентированное программирование».....	12
4. Комплект лабораторных работ по теме «Моделирование и формализация.....	61
Заключение.....	82
Список литературы.....	83

Введение.

Современные профессии становятся все более интеллектоёмкими, требующими развитого логического мышления. Опоздание с развитием мышления – это опоздание навсегда. Поэтому для подготовки учащихся к жизни в современном информационном обществе в первую очередь необходимо развивать логическое мышление, способности к анализу и синтезу. Наиболее доступный материал для развития мышления – это изучение темы «Основы алгоритмизации и объектно-ориентированное программирование». Алгоритмическое мышление является необходимой частью научного взгляда на мир. В то же время оно включает и некоторые общие мыслительные навыки, полезные и в более широком контексте. К таким относится, например, разбиение задачи на подзадачи.

Обучение учащихся основам алгоритмического мышления базируется на понятии исполнителя, которые они изучают в среднем звене общеобразовательной школы. Знания, полученные на уроках информатики, в среднем звене являются основой при изучении темы «Основы алгоритмизации и объектно-ориентированного программирования» на 2 курсе лица, что позволяет усложнить изучение темы и приложить практическое применение программирования в курсе информатики.

В качестве способа по формированию умений и навыков в учебной программе предусмотрены практические работы. Их выполнение является обязательным.

В данной работе представлены лабораторные работы по программированию. Эти работы не представляют особой сложности в выполнении. Однако, они охватывают большие возможности языка программирования V.B.

Работы составлены таким образом, что могут быть выполнены самостоятельно.

На начальном уровне учащиеся выполняют задания по образцу. Для сильных учащихся, предусмотрены задания для самостоятельного решения, которые преподаватель задает в зависимости от знаний учащегося и его профессии.

Преподаватель информатики имеет право изменить тему лабораторной работы или проекта.

Данный сборник призван помочь учителям и преподавателям глубже освоить новейшие информационные технологии, расширить их использование в повседневной работе с учащимися и при подготовке учебных материалов к урокам.

1. Почему Visual Basic?

Этот вопрос стоит перед преподавателями, которые не знают, с какого языка начать изучать тему «Объектно-ориентированное программирование».

Сейчас наиболее известны языки Бейсик, Паскаль, Си. До появления одноименных .NET-языков все они существовали и сейчас существуют в своих самых разных версиях. .NET-языки отличаются от своих тезок-предшественников, несмотря на то, что являются их прямыми потомками. Отличаются не столько внешним видом программ, сколько внутренним содержанием (использование механизмов .NET Framework). Особенно резко отличается Visual Basic .NET.

Рассмотрим, насколько тот или иной язык подходит для изучения азам программирования.

Си – высокопрофессиональный язык, программы, написанные на нем, имеют превосходное быстродействие, в среде программистов Си распространен очень широко. Но он слишком сложен для восприятия новичком, программы на нем пишутся со значительным трудом и с него лучше не начинать.

По мнению профессионалов наиболее легок и прост из упомянутых языков как раз Бейсик. Профессиональные программисты любят его за то, что для создания на нем приличного продукта требуется гораздо меньше времени, чем, скажем, на Си. Поэтому популярность у него огромная.

Паскаль по простоте программирования и эффективности получаемых программ занимает промежуточное положение между Бейсиком и Си. Паскаль существует сейчас в виде его современной версии Object Pascal и программирование на нем ведется в среде Delphi.

Из проведенного анализа мы для изучения темы «Объектно-ориентированное программирование» выбрали язык программирования Visual Basic.

2. Язык объектно-ориентированного программирования Visual Basic.

"Visual" относится к методу, используемому для создания того, что видит пользователь — *графического пользовательского интерфейса*, или GUI. Слово "Basic" относится к аббревиатуре BASIC (Beginners All-Purpose Symbolic Instruction Code — многоцелевой код символьных инструкций для начинающих) языка программирования, который используется программистами намного чаще, чем любой другой язык в истории вычислений. Для создания различных полезных программ достаточно изучить лишь некоторые из его возможностей.

Что такое программирование?

Изначально компьютер не обладает какими-либо функциями.

По существу, компьютер представляет собой множество небольших электронных переключателей, которые могут быть либо включены, либо отключены. Устанавливая различные сочетания этих параметров, можно выполнять различные действия с помощью компьютера. Например, можно отобразить какой-либо объект на экране или подать звуковой сигнал. Таким образом, в процессе программирования задается последовательность компьютерных инструкций, определяющих выполняемые действия.

Что такое язык программирования?

Обмен информацией между людьми осуществляется с использованием языка, содержащего большое количество слов. В компьютерах используется простой язык, состоящий только из нулей и единиц, определяющих соответственно состояния "выключено", и "включено". Теоретически можно обмениваться информацией с компьютером на двоичном языке. Однако это будет очень сложно и неудобно.

Язык программирования выступает в роли переводчика между вами и компьютером. Вместо изучения базового языка компьютера (называемого *машинным языком*) лучше использовать язык программирования, содержащий инструкции, определяющие конкретные действия компьютера.

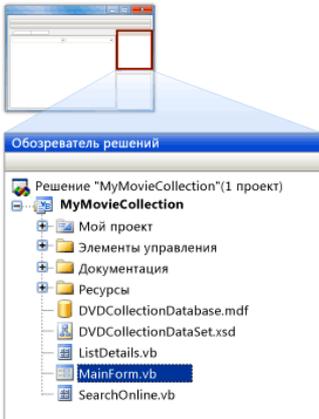
Специализированные программы, называемые *компиляторами*, принимают инструкции на языке программирования и преобразуют их в машинный язык. Это означает, что как программисту Visual Basic вам не нужно понимать, что или как делает компьютер. Достаточно понять, как работает язык программирования Visual Basic.

Начало работы

При первом открытии Visual Basic, большая часть среды IDE заполнена окном **Начальная страница**. **Начальная страница** содержит список последних проектов со ссылками, область **Начало работы** со ссылками на важные разделы справки, а также список ссылок на статьи в Интернете и другие ресурсы. Если ваш компьютер подключен к сети Интернет, этот список будет регулярно обновляться.

В правой части среды IDE отображается окно **Обозреватель решений**. Изначально это окно пустое, но впоследствии здесь будут отображаться сведения о проекте или группах проектов, называемых решениями.

Рисунок 1. Обозреватель решений



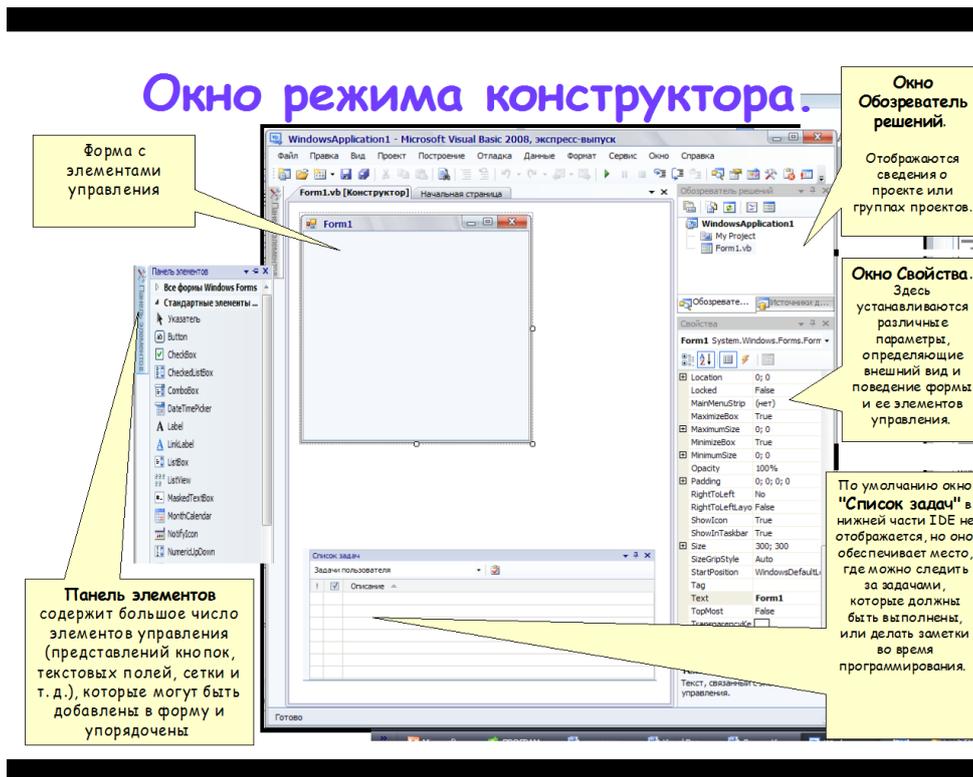
В левой части IDE отображается вертикальная вкладка с пометкой **Панель элементов**. Она также изначально пуста, но во время работы будет заполнена элементами, которые могут использоваться для выполняемой в данный момент задачи. Вдоль верхнего края IDE располагаются строка меню и панель инструментов. Доступные меню и кнопки панели инструментов изменяются в зависимости от текущей задачи — потратьте некоторое время и изучите различные варианты. Можно также настроить меню и панель инструментов в соответствии с личными предпочтениями. Для настройки панели инструментов выберите пункт меню **Сервис** и щелкните **Настройка**. Затем можно упорядочить команды или добавить новую панель инструментов. Вдоль нижнего края IDE располагается строка состояния, отображающая состояние **Готово**. При работе в интегрированной среде разработки строка состояния отображает сообщения, связанные с текущей задачей. Например, на ней отображаются сведения о ходе выполнения процесса при сборке проекта.

Режим конструктора

При открытии или создании проекта внешний вид интегрированной среды разработки изменяется, так как она переключается в *режим конструктора*. IDE в режиме конструктора показана на рисунке ниже. Это визуальная часть Visual Basic, где проектируется внешний вид приложения.

В режиме конструктора **Начальная страница** закрыта другим окном **Конструктор форм**, которое представляет собой пустое полотно, являющееся пользовательским интерфейсом для приложения. Обратите внимание, что **Начальная страница** остается доступной; на нее можно перейти, щелкнув ее вкладку выше **Конструктора форм**.

Рисунок 2. IDE в режиме конструктора



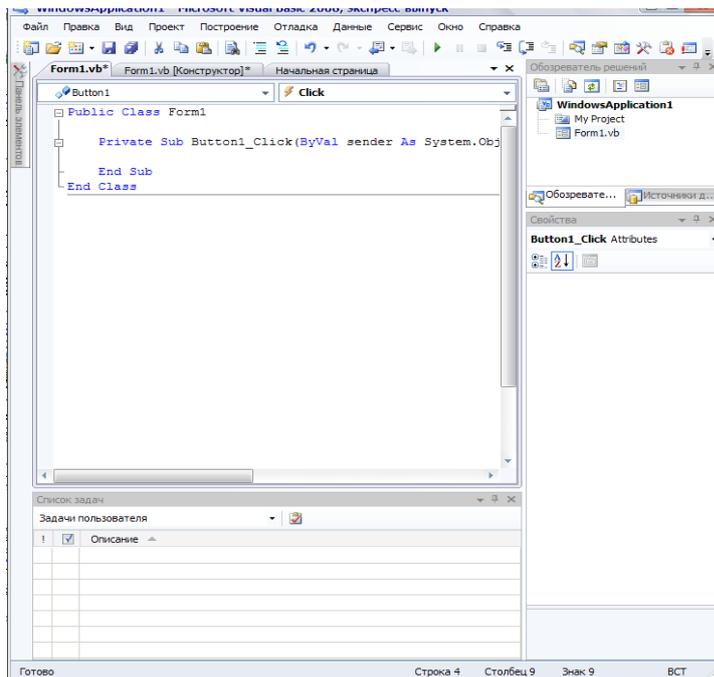
Когда **Конструктор форм** является видимым, **Панель элементов** содержит большое число элементов управления (представлений кнопок, текстовых полей, сетки и т.д.), которые могут быть добавлены в форму и упорядочены по вашему вкусу.

Также обратите внимание, что под окном **Обзорщика решений** отображается окно **Свойства**. Здесь будут устанавливаться различные параметры, определяющие внешний вид и поведение формы и ее элементов управления.

По умолчанию окно "Список задач" в нижней части IDE не отображается, но оно обеспечивает место, где можно следить за задачами, которые должны быть выполнены, или делать заметки во время программирования.

При двойном щелчке мышью на форме или элементе управления открывается новое окно, называемое редактором кода. В этом окне пишется фактический код для приложения. Редактор кода — это больше, чем текстовый редактор, он использует технологию, которая называется IntelliSense и помогает писать код, предоставляя необходимую информацию при вводе.

Рисунок 3. Окно редактора кода

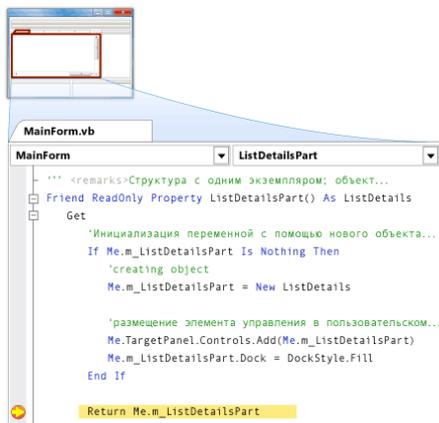


Режим выполнения

При выполнении или отладке приложения интегрированная среда разработки переходит в *режим выполнения*. При этом запускается само приложение, и появляются дополнительные окна, связанные с отладкой. В режиме выполнения нельзя вносить изменения в **Конструктор формы**, окно **Свойства** или **Обозреватель решений**, но можно изменить код в редакторе кода.

В режиме выполнения в нижней части IDE появляется новое окно, которое называется окном **интерпретации**. После перевода приложения в режим приостановки выполнения можно запрашивать значения и проверять код в окне **интерпретации**. Режим приостановки выполнения показан на приведенном ниже рисунке.

Рисунок 4. Форма Visual Basic в режиме приостановки выполнения



Дополнительные окна для просмотра значений переменных, просмотра выходных данных и других задач отладки можно просмотреть во время выполнения, выбрав их в меню **Отладка**.

Visual Basic, экспресс-выпуск позволяет создавать несколько типов приложений. Сюда входят приложения Windows Forms, Windows Presentation Foundation, консольные приложения и библиотеки классов. Каждое создаваемое приложение находится в своем собственном проекте, и предоставляются шаблоны проектов,

помогающие начать работу.

При создании нового проекта в диалоговых окнах **Создать проект** и **Добавить проект** отображаются значки, показывающие доступные типы проектов и их шаблоны. При создании нового проекта в Visual Basic, экспресс-выпуск доступны следующие шаблоны проектов.

Типы проектов

Шаблон проекта	Используется для создания
Шаблон приложения Windows	Windows-приложений, работающих локально на компьютерах пользователей. Можно создать что угодно, начиная от простого средства с одним окном, вроде калькулятора Windows, и заканчивая приложением, которое имеет несколько окон и дополнительные возможности.
Шаблон библиотеки классов	Применяется для создания многократно используемых классов или компонентов, которые могут использоваться совместно несколькими проектами.
Шаблон консольного приложения	Используется для создания приложений с командной строкой, которые запускаются из командной строки Windows и не имеют визуального интерфейса.
Приложение WPF	Используется для создания изолированных приложений Windows Presentation Foundation.
Приложение обозревателя WPF	Используется для создания приложений Windows Presentation Foundation, расположенных в выбранном месте.

Процесс разработки

Visual Basic, делает разработку приложения легкой. Основная ее часть состоит из следующих этапов.

- **Создание проекта.** Проект содержит все файлы, необходимые для приложения, и хранит сведения о приложении. Иногда приложение содержит несколько проектов, например, проект **Приложение Windows** и один или несколько проектов **Библиотека классов**. Такие приложения называются *решениями* (это просто другое название для группы проектов).
- **Создание пользовательского интерфейса.** Производится путем перетаскивания различных элементов управления, таких как кнопки и текстовые поля, в область конструктора, которая называется *формой*. Затем можно установить свойства, определяющие внешний вид и поведение формы и ее элементов управления.

- **Написание кода.** Далее необходимо написать код Visual Basic, который определяет, как будет вести себя приложение и как оно будет взаимодействовать с пользователем.
- **Проверка кода.** Приложение необходимо протестировать, чтобы убедиться, что оно ведет себя именно так, как требуется. Этот процесс называется *отладка*.
- **Распространение приложения.** После завершения приложения можно установить готовую программу на компьютер или распространить ее и передать другим лицам.

3. Комплект лабораторных работ по теме «Алгоритмизация и объектно-ориентированное программирование».

Лабораторная работа №1

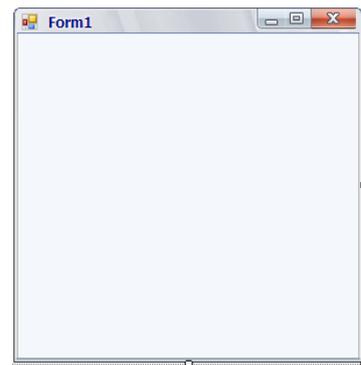
Тема: Создание и редактирование формы.

Цель: научиться создавать пользовательскую форму для будущих проектов, уметь настроить ее, применяя различные свойства формы, используя язык программирования Visual Basic.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

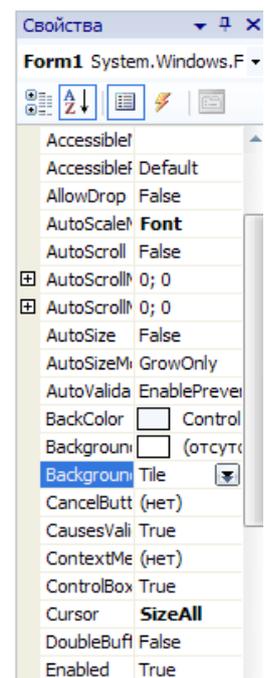
Теоретическая часть:

Формы являются основными блоками построения пользовательского интерфейса. Каждая форма в приложении представляет окно, которое появляется перед пользователем. При работе в IDE (интегрированной среде разработки) Visual Basic, форма является *конструктором*, предназначенным для разработки пользовательского интерфейса приложения.



Можно изменить внешний вид формы путем задания свойств формы в окне **Свойства**.

BackColor	<i>Используется для задания цвета фона.</i>
BackgroundImageLayout	<i>Макет фонового изображения, используемый для компонента.</i>
ShowInTaskbar	<i>Будет ли форма отображаться в Панели задач Windows при запуске программы.</i>
Text	<i>Изменяет название формы</i>
AutoScaleMode	<i>Определяет изменение масштаба формы или элемента управления при изменении разрешения экрана или</i>



	<i>шрифтов.</i>
Cursor	<i>Курсор, отображаемый при наведении указателя мыши на данный элемент управления.</i>
FormBorderStyle	<i>Определяет внешний вид и поведение границы и строки заголовка данной формы.</i>
HelpButton	<i>Определяет, содержит ли форма кнопку справки в строке заголовка.</i>
Icon	<i>Указывает значок для формы. Этот значок используется в системном меню данной формы, а также отображается, когда форма свернута.</i>
MaximizeBox	<i>Определяет, содержит ли форма в правом верхнем углу строки заголовка значок разворачивания до максимального размера.</i>
MaximumSize	<i>Максимальный размер, который может иметь форма.</i>
MinimizeBox	<i>Определяет, содержит ли форма в правом верхнем углу строки заголовка значок свертывания.</i>
MinimumSize	<i>Минимальный размер, который может иметь форма.</i>
StartPosition	<i>Определяет первоначальное положение формы при ее отображении.</i>
WindowState	<i>Определяет исходное видимое состояние данной формы.</i>

ПРАВИЛА БЕЗОПАСНОСТИ:

- 1. Не начинайте работу без разрешения преподавателя.**
- 2. Не удаляйте программы и файлы, которые установлены в компьютере.**
- 3. При работе с дискетами не допускайте их падения не кладите на монитор или системный блок. Это приводит к их перемагничиванию, порче.**
- 4. После выполнения работы верните первоначальный вид рабочего стола.**

Ход работы.

Задание.

Составить пользовательскую форму.

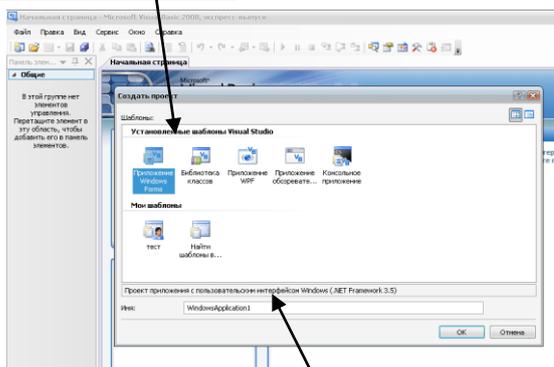
1) В меню Windows **Пуск** выберите **Visual Basic 2008**, экспресс-выпуск.

Появится экран "Добро пожаловать в Visual Basic Express". Это интерфейс для Visual Basic 2008, экспресс-выпуск, также называемый *интегрированной средой разработки* или *IDE*.

2) В меню **Файл** выберите пункт **Создать проект**.

3) Откроется диалоговое окно Создать проект.

4) В диалоговом окне Создание проекта в области Шаблоны выберите **Приложение Windows**.



5) В поле Имя введите **FirstFormВаша Фамилия** и нажмите кнопку **ОК**.

Создается новый проект Windows Forms. Новая форма появится в главном окне и ее свойства будут видны в окне Свойства в правом нижнем углу интегрированной среды разработки Visual Basic.

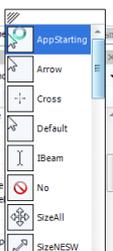
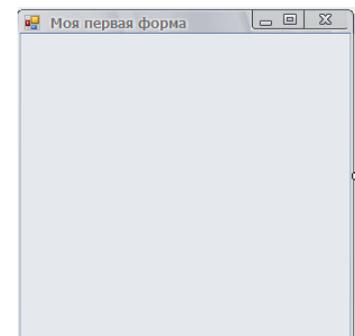
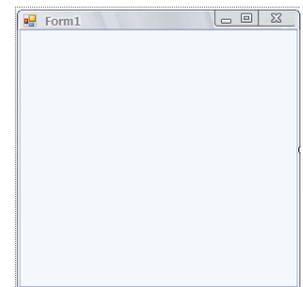
6) Нажмите на форму, чтобы выделить ее.

7) В окне Свойства измените свойство **Text** на "Моя первая форма" и нажмите клавишу **ENTER**.

Текст в верхней части формы изменится после ввода нового значения.

8) В окне Свойства измените свойство **BackColor** на другой цвет, выбрав цвет из выпадающего меню.

Обратите внимание, что свойство BackColor изменяется через специальный интерфейс. Этот интерфейс позволяет просмотреть цвет, прежде чем выбрать его, и позволяет выбирать системные, стандартные веб-цвета или пользовательские цвета.



9) Изменить вид курсора который будет изменяться при наведении на форму. Для этого в окне свойства выберите **Cursor**, затем выберите один из видов курсора на ваше усмотрение.

10) Установите такое свойство для формы чтобы при запуске программы форма отображаться в Панели задач Windows.

Для этого в окне свойства установите в свойстве **ShowInTaskbar** значение **True**.

11) Установите объемный внешний вид вашей программы. Для этого в окне свойства установите в свойстве **FormBorderStyle** значение **Fixed3D**.

12) Установите такое свойство форме, чтобы при запуске она отображалась в центре экрана. Для этого в окне свойства установите в свойстве **StartPosition** значение **CenterScreen**.

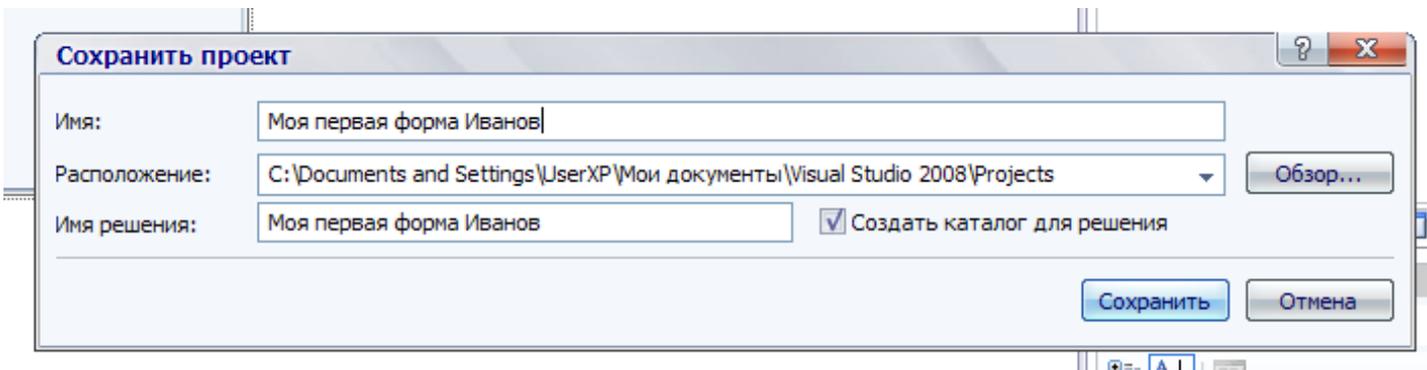
13) Запустите программу на исполнение.

Для этого необходимо в пункте меню **Отладка – Начать отладку**, или нажать  кнопку

Покажите созданную форму преподавателю.

14) Сохраните созданную форму в своей папке.

Для этого необходимо выполнить последовательность действий: **Файл- Сохранить все**- в появившемся окне в строке **Имя:** ввести название формы «Моя первая форма и ваша Фамилия»; в строке **Расположение** указать путь к вашей папке где будет храниться ваша программа – **Сохранить**.



Лабораторная работа №2

Тема: Создание простейшего проекта с помощью языка программирования Visual Basic.

Цель: научиться создавать, используя элементы Button, Label простые программы с помощью языка программирования Visual Basic.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

Использование кнопок

Наиболее простым способом работы с программой является работа при помощи кнопок. Например, многие программы имеют кнопку **Выход**. Кнопки, как правило, представляют собой прямоугольные элементы управления на форме, которые выглядят рельефными. Кнопка создается с помощью элемента **Button**. Существует множество свойств, с помощью которых можно изменить их внешний вид.

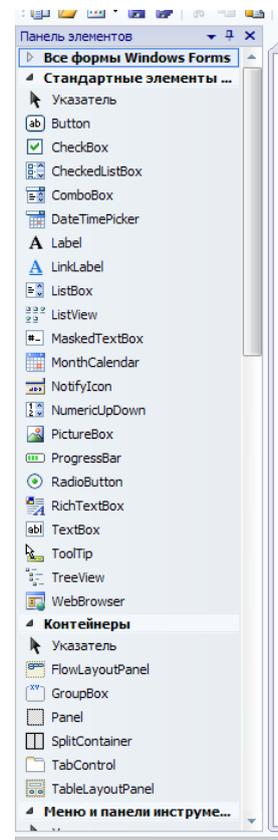
Свойства: **свойство Text**, определяет отображаемый текст, этот текст будет написан *шрифтом* или гарнитурой, задаваемой **свойством Font**. **Свойство BackColor** определяет цвет кнопки, а **свойство ForeColor** определяет цвет текста.

Когда пользователь нажимает кнопку во время выполнения, **Button** вызывает событие Click. При возникновении события элементы управления запускают код в ответ на эти события. Можно написать код, который должен будет выполняться, когда пользователь нажимает кнопку, посредством создания обработчика событий.

Обработчик событий — это метод, который выполняется при возникновении события. Когда пользователь нажимает кнопку, вызывается обработчик событий Click этой кнопки.

Отображение текста с помощью надписи

Элемент управления **Label** является основным элементом для отображения текста. Он отображается на форме в виде текста, заключенного в область



прямоугольной формы. Цвет этой области обычно совпадает с цветом формы. Таким образом, он отображается просто как текст на форме.

Поскольку элемент управления **Label** главным образом предназначен для отображения текста, наиболее важными свойствами **Label** являются свойства, определяющие его внешний вид. **Свойство Text** содержит текст, отображаемый в элементе управления Label. **Свойство Font** определяет шрифт для отображения текста, содержащегося в свойстве Text. **Свойство ForeColor** определяет цвет текста, а **свойство BackColor** – цвет окружающей его области.

ПРАВИЛА БЕЗОПАСНОСТИ:

1. Не начинайте работу без разрешения преподавателя.
2. Не удаляйте программы и файлы, которые установлены в компьютере.
3. Не открывайте чужие файлы и папки.
4. При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.
5. Сохраняйте свои файлы только в ваши папки.
6. После выполнения работы верните первоначальный вид рабочего стола.
7. Не пытайтесь самостоятельно устранять неисправности в работе компьютера.

Ход работы.

Задание.

Составить программу, в которой меняется текстовое сообщение.

- 1) В меню Windows **Пуск** выберите **Visual Basic 2008**, экспресс-выпуск.

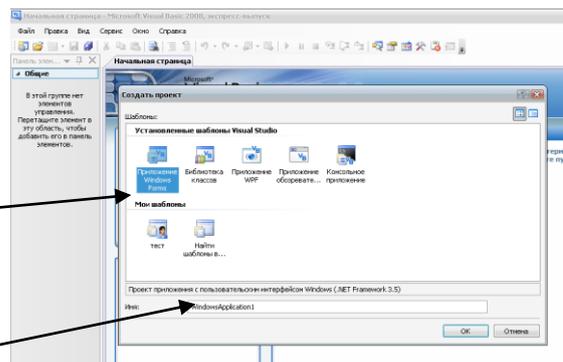
Появится экран "Добро пожаловать в Visual Basic Express". Это интерфейс для Visual Basic 2008, экспресс-выпуск, также называемый *интегрированной средой разработки* или *IDE*.

- 2) В меню **Файл** выберите пункт **Создать проект**.

3) Откроется диалоговое окно Создать проект.

4) В диалоговом окне Создание проекта в области Шаблоны выберите **Приложение Windows**.

- 5) В поле Имя введите **Первая программа**



Ваша Фамилия и нажмите кнопку **ОК**.

Создается новый проект Windows Forms. Новая форма появится в главном окне и ее свойства будут видны в окне Свойства в правом нижнем углу интегрированной среды разработки Visual Basic.

Настройка формы.

6) Нажмите на форму, чтобы выделить ее.

7) В окне Свойства измените свойство **Text** на "Моя первая программа" и нажмите клавишу **ENTER**.

Текст в верхней части формы изменится после ввода нового значения.

8) В окне Свойства измените свойство **BackColor** на другой цвет, выбрав цвет из выпадающего меню.

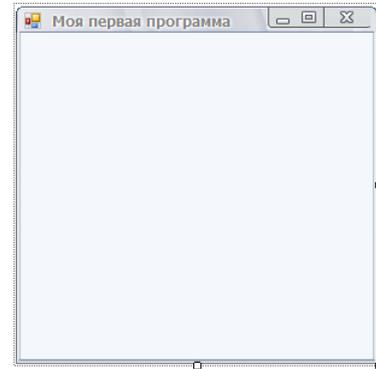
Обратите внимание, что свойство *BackColor* изменяется через специальный интерфейс. Этот интерфейс позволяет просмотреть цвет, прежде чем выбрать его, и позволяет выбирать системные, стандартные веб-цвета или пользовательские цвета.

9) Установите такое свойство для формы чтобы при запуске программы форма отображаться в Панели задач Windows.

Для этого в окне свойства установите в свойстве **ShowInTaskbar** значение **True**.

10) Установите объемный внешний вид вашей программы. Для этого в окне свойства установите в свойстве **FormBorderStyle** значение **Fixed3D**.

11) Установите такое свойство форме, чтобы при запуске она отображалась в центре экрана. Для этого в окне свойства установите в свойстве **StartPosition** значение **CenterScreen**.



Создание пользовательского интерфейса

12) Выберите панель элементов. Открывается панель элементов.

13) Щелкните вкладку **Все формы Windows Forms** в панели элементов, а затем четыре раза перетащите элемент управления **Button** на форму и расположите их как показано на рисунке.

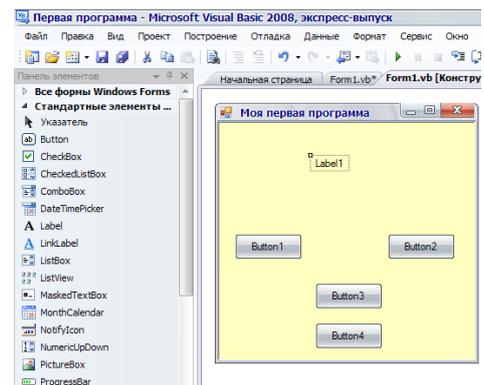
14) С той же вкладки перетащите элемент управления **Label**.

Настройка внешнего вида и поведения

Установка свойств элементов управления

15) В меню **Конструктор форм** выберите элемент управления **Button1**.

16) В окне **Свойства** все свойства этого элемента. Выберите свойство **Text** и нажмите кнопку со стрелкой. В открывшемся меню удалите **Button1** и введите слово «Первая фраза».



17)Проделайте те же действия для элементов **Button2, Button3, Button4**. Только на кнопке 2 напишите слово «Вторая фраза», на кнопке 3 «Очистить», а на кнопке 4 «Выход».

18)Измените цвет фона кнопок. Для этого выделите настраиваемый элемент, в окне Свойства выберите **BackColor**, и из открывшегося списка выберите понравившийся вам цвет.

19)Для кнопки «Выход» установите размер шрифта 10 полужирный. Для этого используйте свойство **Font** при этом кнопка должны быть активна (выделена).

20)Установите свойства для элемента **Label1**. Выберите свойство Text замените название на фразу «Моя программа приветствует Вас», с помощью свойства **Font** измените размер шрифта на 12 полужирный Monotype Corsiva , с помощью свойства **ForeColor** измените цвет текста.



Добавление кода

21) Дважды щелкните по кнопке «Первая фраза». **Откроется редактор кода.** В редакторе уже содержится некоторый код. Этот код является *обработчиком событий*, также называемым процедурой *Sub*. Весь код внутри этой процедуры (между **Sub** и **End Sub**) выполняется при каждом нажатии кнопки.

22)Внутри этой процедуры введите следующий код: Label1.Text = "Моя первая программа" В результате код примет вид:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
```

```
Label1.Text = "Моя первая программа"
```

```
End Sub
```

23) Выполните аналогичные действия для кнопок «Вторая фраза» «Очистить» и «Выход».

Для кнопки «Вторая фраза» написать код: Label1.Text = "Моя программа работает"

Для кнопки «Очистить» написать код : Label1.Text = ""

Для кнопки «Выход» написать код: End

24)Запустить программу. Для этого выполните команду Отладка – Начать отладку. Либо нажмите F5

Продемонстрируйте работу преподавателю.

25)***Дополнительное задание**: Создайте еще одну кнопку в вашей программе, при нажатии на которую будет появляться информация о разработке данной программы.

Продемонстрируйте вашу программу преподавателю.

26)Сохраните созданную форму в своей папке.

Лабораторная работа №3

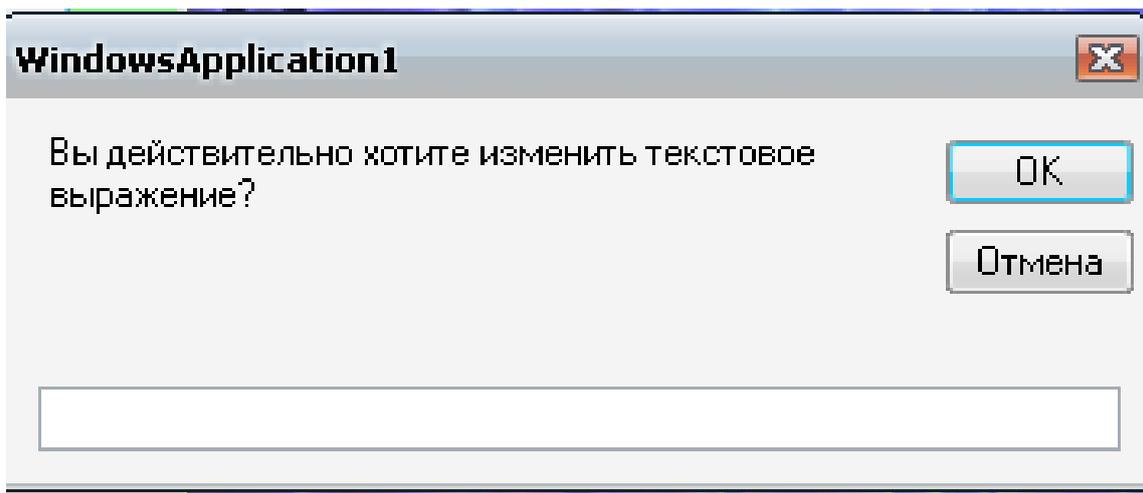
Тема: Специальное окно InputBox для ввода данных.

Цель: изучить специальное окно для ввода данных **объекта:** InputBox.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

INPUT (ввод) – оператор ввода. Используется для передачи в программу каких-либо значений. Использовании объекта InputBox при написании кода для элемента формы при запуске программы при нажатии на элемент которому задана процедура Input на экран выводится диалоговое окно предназначенное для ввода данных.



ПРАВИЛА БЕЗОПАСНОСТИ:

1. Не начинайте работу без разрешения преподавателя.
2. Не удаляйте программы и файлы, которые установлены в компьютере.
3. Не открывайте чужие файлы и папки.
4. При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.
5. Сохраняйте свои файлы только в ваши папки.
6. После выполнения работы верните первоначальный вид рабочего стола.
7. Не пытайтесь самостоятельно устранять неисправности в работе компьютера.

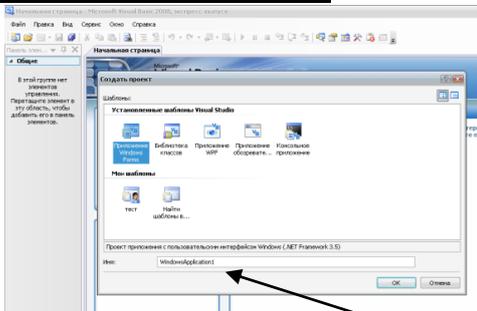
Ход работы.

Задание.

Составить программу, в которой открывается диалоговое окно.

Создание проекта для программы

- 1) В меню Windows **Пуск** выберите Visual Basic 2008, экспресс-выпуск.
- 2) Появится экран "Добро пожаловать в Visual Basic Express". Это интерфейс для Visual Basic 2008, экспресс-выпуск, также называемый *интегрированной средой разработки* или *IDE*.
- 3) В меню **Файл** выберите пункт **Создать проект**.
- 4) В диалоговом окне Создание проекта в области Шаблоны выберите **Приложение Windows**.

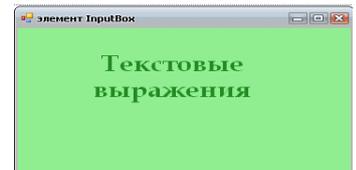


- 5) В поле Имя введите Программа с окном ввода Ваша Фамилия и нажмите кнопку **ОК**.

Создание пользовательского интерфейса

- 6) Настройте общий вид формы: придайте ей имя «Элемент InputBox», измените цвет фона по вашему усмотрению, установите свойство которое при запуске программы будет помещать окно в центре экрана.

Text	Элемент InputBox
BackColor	Любой цвет формы на ваше усмотрение
StartPosition	CenterScreen

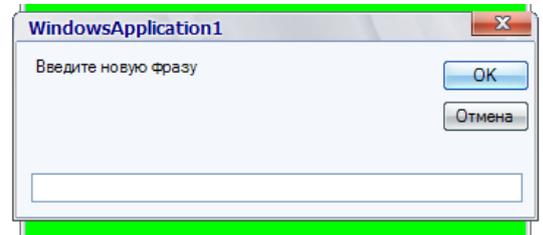


- 7) Выберите **панель элементов**. Открывается **панель элементов**.
- 8) Щелкните вкладку **Все формы Windows Forms** в панели элементов, перетащите элемент управления Label. Выполните следующие настройки:

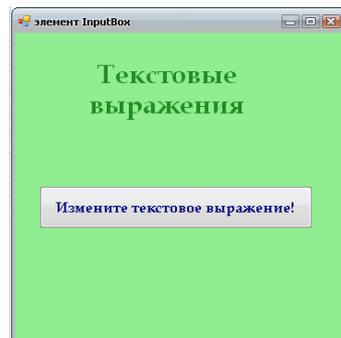
Text	Текстовые выражения
AutoSize	False
TextAlign	MiddleCenter (по середине в центре)
Font	Размер, стиль текста установите по-вашему усмотрению

ForeColor Цвет текста

- 9) Напишите программный код ситуационной процедуры, которая реагирует на щелчок (Click) мышкой по полю метки Label1: Label1.Text = InputBox("Введите новую фразу")
- 10) Запустите программу. Щелкните по надписи «Текстовые выражения». В открывшемся окне **InputBox** введите новую фразу «**Изучение Visual Basic**». Попробуйте изменить фразу несколько раз.



- 11) Поместите на форме кнопку с именем: "Измените текстовое выражение!". Для этого разместите на форме инструмент **Button**. Свойству **Text** присвойте значение - Измените текстовое выражение! Измените цвет кнопки, шрифта, стиль, размер шрифта, вид курсора который будет принимать стрелка при наведении на кнопку – по вашему усмотрению.



Text Текстовые выражения

BackColor Любой цвет кнопки по вашему усмотрению.

Font Размер, стиль текста установите по-вашему усмотрению

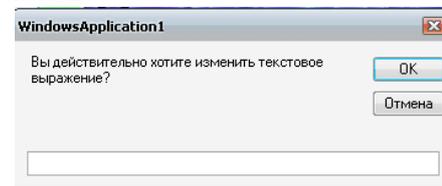
ForeColor Цвет текста

Cursor Выберите вид курсора по вашему усмотрению.

- 12) Напишите программный код для кнопки. Для этого двойным щелчком по кнопке «Измените текстовое выражение» откройте редактор кода и впишите в процедуру следующий код:

Label1.Text = InputBox("Вы действительно хотите изменить текстовое выражение?")

- 13) Запустите программу. Измените фразу.

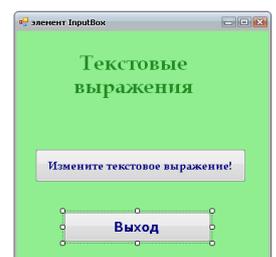


- 14) Добавьте на форму кнопку «Выход», используя панель свойств установите цвет кнопки, шрифта, стиль, размер шрифта, вид курсора который будет принимать стрелка при наведении на кнопку – по вашему усмотрению.

- 15) Добавьте для нее следующий код : End.

Продемонстрируйте работу преподавателю!

- 16) Сохраните проект.



Лабораторная работа №4

Тема: Использование переменных при составлении программ.

Цель: научиться использовать элемент управления TextBox для отображения и получения текста, введенного пользователем; использовать операции с переменными при составлении программ; использовать функцию MsgBox для вывода результатов в дополнительном окне сообщения.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

Получение текста с помощью текстового поля.

Если отображение и ввод текста требуются одновременно, удобно использовать элемент управления **TextBox**. В дополнение к возможностям отображения текста, элемент управления **TextBox** позволяет пользователям во время выполнения вводить в поле **TextBox** текст, который может извлекаться программой.

Наиболее важными для элемента **TextBox** свойствами являются свойства, определяющие его внешний вид. Одним из важнейших является **свойство Text**, которое представляет текст в элементе управления **TextBox**. Когда пользователь вводит текст в элемент управления **TextBox**, свойство Text обновляется в соответствии с изменениями. Таким образом, текст, отображаемый в элементе **TextBox**, всегда отражает значение свойства Text.

Существуют также свойства, влияющие на поведение элемента управления **TextBox**. **Свойство Multiline** определяет, можно ли отображать в элементе **TextBox** несколько строк. Если этому свойству присвоено значение **False**, элемент управления **TextBox** всегда будет вмещать только одну строку по высоте и не сможет быть увеличен по вертикали. Если же присвоено значение **True**, элемент управления **TextBox** сможет отображать несколько строк, а его высота может быть любой.

Функция

MsgBox – оператор вывода значения, который отображает результат в окне сообщения.

Переменные.

Переменные являются важными понятиями в компьютерном программировании. Переменная представляет собой букву или имя, которое может

хранить значение. При создании программ для компьютера переменные можно использовать для хранения чисел, например высоты здания, или слов, например имени пользователя. Проще говоря, переменные используются для представления любого рода сведений, необходимых программе.

Хранение информации в переменных

Для использования переменной необходимо выполнить три действия:

1. **Объявить переменную.** Сообщить программе имя и тип переменной, которую требуется использовать.
2. **Присвоить переменной значение.** Передать переменной значение для хранения.
3. **Непосредственно использовать переменную.** Извлечь значение, содержащееся в переменной, и использовать его в программе.

При объявлении переменной необходимо решить, как она будет называться, и какой *тип данных* будет ей присвоен. Имя переменной может быть любым, но оно должно начинаться с буквы или знака подчеркивания.

Чтобы грамотно использовать переменную, нужно её сначала объявить, зарезервировать под неё память. Для этого используются следующие ключевые слова: **Dim, Private, Public, Static, Global**. Переменные можно объявлять на уровне модуля и на уровне процедуры. Под уровнем модуля подразумевается часть формы General и стандартный модуль, а под уровнем процедуры подразумевается Sub, Function и т.д.

Строка кода **Dim aNumber As Integer** сообщает программе, что требуется использовать переменную с именем **aNumber**, и что она должна хранить целые числа (тип данных **Integer**). Поскольку **aNumber** является переменной типа **Integer**, она может хранить только целые числа. Если бы нужно было хранить, например, число 42.5, то использовался бы тип данных **Double**. А для хранения слов используется тип данных с именем **String**. Еще одним типом данных, который следует упомянуть здесь, является тип **Boolean**, который может хранить значения **True** или **False**.

Типы переменных.

При объявлении лучше всего объявить переменную, тем самым указать, сколько памяти нам потребуется:

Dim aName As String в этом случае мы объявляем строковую переменную. Необъявленная переменная автоматически стаёт типом **Variant**.

Тип данных	Размер	Диапазон значений	Пример использования
Integer (Целое)	2 байта	от -32 768 до 32 767	Dim Bird%

			Bird% = 37
Long Integer (Длинное целое)	4 байта	от -2 147 483 648 до 2 147 483 647	Dim Loan& Loan& = 350,000
Single - precision Floating point (Одинарной точности с плавающей десятичной точкой)	4 байта	от -3.402823E38 до 3.402823E38	Dim Price! Price! = 899.99
Double - precision Floating point (Двойной точности с плавающей десятичной точкой)	8 байт	от - 1.79769313486232D308 до 1.79769313486232D308	Dim Pi# Pi# = 3.1415926535
Currency (Денежные единицы)	8 байт	от - 922337203685477.5808 до 922337203685477.5807	Dim Debt@ Debt@ = 7600300.50
String (Строка)	1 байт на символ	от 0 до 65 535 символов	Dim Dog\$ Dog\$ = "pointer"
Boolean (Логический)	2 байт	True (Истина) или False (Ложь)	Dim Flag As Boolean Flag = True
Date (Дата)	8 байт	от January (Январь)1, 100, до December (Декабрь) 31, 9999	Dim Birthday As Date Birthday =#3-1-63#
Variant (Вариант)	16 байт (для чисел); 22 байт на символ (для строк)	для всех типов данных	Dim Total Total = 289.13

Присваивание переменным значений.

Значение переменной присваивается при помощи знака =, иногда называемого *оператором присваивания*.

```
Dim aName As Integer = 42
```

Эта строка кода присваивает переменной с именем `aNumber` значение 42.

Арифметические операторы

Оператор	Описание	Пример
+ (Сложение)	Возвращает сумму двух операндов	5 + 4
- (Вычитание)	Возвращает разность двух операндов	5 — 4
* (Умножение)	Возвращает произведение двух операндов	5 * 4
/ (Деление)	Возвращает частное двух операндов	5 / 4

Тип переменных, которые используются при выполнении арифметических действий, может повлиять на результат. При делении одного числа на другое часто возвращается значение, которое не является целым числом. Например, если вы делите 3 на 2, результатом будет 1,5. Если возвращаемое этим выражением значение присваивается переменной типа **Integer**, оно округляется до ближайшего целого числа, то есть до 2. При выполнении деления следует использовать переменную типа **Double** для хранения возвращаемого значения.

ПРАВИЛА БЕЗОПАСНОСТИ:

- 1. Не начинайте работу без разрешения преподавателя.**
- 2. Не удаляйте программы и файлы, которые установлены в компьютере.**
- 3. Не открывайте чужие файлы и папки.**
- 4. При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.**
- 5. Сохраняйте свои файлы только в ваши папки.**
- 6. После выполнения работы верните первоначальный вид рабочего стола.**
- 7. Не пытайтесь самостоятельно устранять неисправности в работе компьютера.**

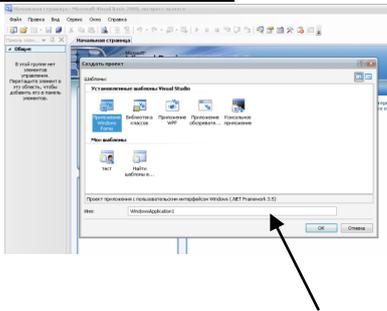
Ход работы.

Задания.

Составить программу, в которой вычисляются арифметические действия с числами.

Создание проекта для программы

- 1) В меню Windows **Пуск** выберите Visual Basic 2008, экспресс-выпуск.
- 2) Появится экран "Добро пожаловать в Visual Basic Express". Это интерфейс для Visual Basic 2008, экспресс-выпуск, также называемый *интегрированной средой разработки* или *IDE*.
- 3) В меню **Файл** выберите пункт **Создать проект**.
- 4) В диалоговом окне Создание проекта в области Шаблоны выберите **Приложение Windows**.



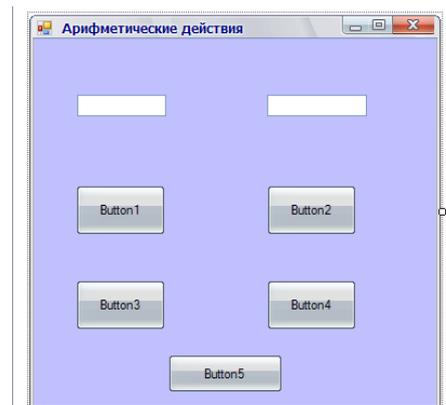
- 5) В поле Имя введите **Арифметические действия Ваша Фамилия** и нажмите кнопку **ОК**.
- 6) Настройте общий вид формы: придайте ей имя «Арифметические действия», измените цвет фона по вашему усмотрению, установите свойство, которое при запуске программы будет помещать окно в центре экрана.

Text	Арифметические действия
BackColor	Любой цвет формы на ваше усмотрение
StartPosition	CenterScreen

7) Из **Панели элементов** перетащите на форму два элемента управления **TextBox**.

8) Из **панели элементов** перетащите на форму 5 элементов управления **Button**. Как показано на рисунке.

- 9) Элементу **Button1** – присвойте имя «Сумма»
- Button2** – присвойте имя «Разность»
- Button3** – присвойте имя «Произведение»
- Button4** – присвойте имя «Частное»
- Button5** – присвойте имя «Выход»



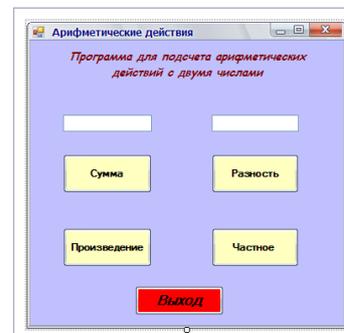
10) Настройте общий вид кнопок **Button** по вашему усмотрению

Text	Вводится текст, который должен размещаться на кнопках.
BackColor	Любой цвет кнопки по вашему усмотрению.
Font	Размер, стиль текста установите по-вашему усмотрению
ForeColor	Цвет текста
Cursor	Выберите вид курсора по вашему усмотрению.

11) С Панели элементов перетащите на форму элемент **Label**

12) Настройте общий вид элемента **Label**

Text	Вставьте текст: Программа для подсчета арифметических действий с двумя числами
AutoSize	false
Font	Размер, стиль текста установите по-вашему усмотрению
ForeColor	Цвет текста установите по вашему усмотрению.



13) Дважды щелкните элемент управления **Button1 (Сумма)**, чтобы открыть редактор кода.

В процедуру события **Button1_Click** добавьте следующий код.

```
Dim A As Double = TextBox1.Text
```

```
Dim B As Double = TextBox2.Text
```

```
MsgBox("Сумма чисел = " & A + B)
```

Первые две строки объявляют переменные *A* и *B*. *A* и *B* будут хранить числовые значения, используемые в этой программе, значения двух элементов управления **TextBox** (текста) будут присвоены переменным *A* и *B*.

Последняя строка создает выражение (сложение) с двумя переменными, а также отображают результаты этих выражений в окне сообщения «Сумма чисел =».

14) Дважды щелкните элемент управления **Button2 (Разность)**, чтобы открыть редактор кода.

В процедуру события **Button2_Click** добавьте следующий код.

```
Dim A As Double = TextBox1.Text
```

```
Dim B As Double = TextBox2.Text
```

```
MsgBox("Разность чисел = " & A - B)
```

15) Дважды щелкните элемент управления **Button3 (Произведение)**, чтобы открыть редактор кода. В процедуру события **Button3_Click** добавьте следующий код.

```
Dim A As Double = TextBox1.Text
```

```
Dim B As Double = TextBox2.Text
```

```
MsgBox("Произведение чисел = " & A * B)
```

16) Дважды щелкните элемент управления **Button4 (Частное)**, чтобы открыть редактор кода.

В процедуру события **Button4_Click** добавьте следующий код.

```
Dim A As Double = TextBox1.Text
```

```
Dim B As Double = TextBox2.Text
```

```
MsgBox("Частное чисел = " & A / B)
```

17) Дважды щелкните элемент управления **Button5 (Выход)**, чтобы открыть редактор кода.

В процедуру события **Button5_Click** добавьте следующий код: End

18) Редактор кода будет иметь вид:

19) Запустите программу на исполнение

Для этого выполните команду Отладка – Начать отладку. Либо нажмите F5.

```

Public Class Form1
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Dim A As Double = TextBox1.Text
        Dim B As Double = TextBox2.Text

        MsgBox("Сумма чисел = " & A + B)
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click
        Dim A As Double = TextBox1.Text
        Dim B As Double = TextBox2.Text

        MsgBox("Разность чисел = " & A - B)
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
        Dim A As Double = TextBox1.Text
        Dim B As Double = TextBox2.Text

        MsgBox("Произведение чисел = " & A * B)
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click
        Dim A As Double = TextBox1.Text
        Dim B As Double = TextBox2.Text

        MsgBox("Частное чисел = " & A / B)
    End Sub

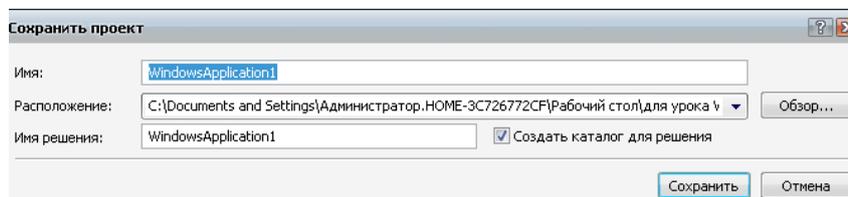
    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click
        End
    End Sub
End Class

```

***Дополнительное задание:** На основе созданного проекта создайте программу, которая будет осуществлять арифметические действия с тремя числами.

Продемонстрируйте работу преподавателю!

20) Сохраните проект. В программе Visual Basic выполните команду Файл-Сохранить все. В открывшемся окне укажите имя и путь к вашей папке.



Лабораторная работа №5

Тема: Компьютерные программы разветвленной структуры.

Цель: научиться использовать команды ветвления, при программировании используя язык Visual Basic.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

Программа ветвящейся структуры будем называть такую программу, которой выбирается один из нескольких возможных путей (вариантов) вычислительного процесса.

Признаком разветвляющейся структуры программы является наличие операций условного перехода, когда происходит проверка истинности некоторого логического выражения (проверяемое условие) и в зависимости от истинности или ложности проверяемого условия выполняется то или иное действие в программе.

Оператор If

Разветвляющая структура программы записывается при помощи оператора `if`

If	переводится "если"
Then	переводится "то"
Else	переводится "иначе"

Пример записи нового оператора:

`If a=28 Then Debug.WriteLine(f) Else k=44`

Переводится он так:

ЕСЛИ a=28 **ТО** пиши f **ИНАЧЕ** присвой k значение 44

Другими словами, компьютеру предлагается прежде, чем что-то делать, сначала подумать, правда ли, что `a=28`, и если правда, то выполнить оператор `Debug.WriteLine(f)`, в противном случае выполнить оператор `k=44`.

Оператор **If** включает в себя другие операторы, которые выполняются или не выполняются в зависимости от какого-то условия. Тем не менее, вся эта запись считается одним оператором *If*.

У VB есть две формы оператора **If**: **однострочная** и **многострочная**.

Синтаксическая схема однострочного оператора If:

If *условие* **Then** *операторы* **[Else** *операторы* **]**

Квадратные скобки означают, что их содержимое можно писать, а можно и не писать в операторе.

<i>операторы</i>	любой оператор VB или группа операторов, разделенных двоеточиями		
<i>условие</i>	<u>пока</u> под условием будем понимать два арифметических или строковых <i>выражения</i> , соединенных <i>знаком сравнения</i>		
<i>знак сравнения</i>	знаков сравнения шесть: > больше >= больше или равно = равно < меньше <= меньше или равно <> не равно		

Примеры записи оператора If.

ФРАГМЕНТ ПРОГРАММЫ	ПЕЧАТЬ
a=10 If a>2 Then Debug.WriteLine("!!!") Else Debug.WriteLine("!")	!!!
a=4 If a>5 Then a=a+10 Else a=a-1 Debug.WriteLine(a)	3
s=6 If s-8<>0 Then s=2*s Debug.WriteLine(s)	12

Синтаксическая схема многострочного оператора If:

```

If условие Then
    операторы
    операторы
    .....
[ ElseIf условие Then
    операторы
    операторы
    ..... ]
.....
[ Else
    операторы
    операторы
    ..... ]

```

End If

Ветвей *Elseif* может быть сколько угодно или совсем не быть. Ветвь *Else* может присутствовать, а может отсутствовать. Если она есть, то одна и стоит последней.

Определите без компьютера, что напечатает данная процедура:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Dim a As Integer = 0
    If 3 > 2 Then
        a = a + 1
        a = a + 2
    ElseIf 3 > 2 Then
        a = a + 4
    End If
    a = a + 10
    If 3 < 2 Then
        a = a + 20
    ElseIf 3 > 2 Then
        a = a + 40
        a = a + 100
    Else
        a = a + 200
    End If
    a = a + 500
    If 3 < 2 Then
        a = a + 1000
    ElseIf 3 < 2 Then
        a = a + 2000
    Else
        a = a + 4000
    End If
    a = a + 8000
    Debug.WriteLine(a)
End Sub
```

ПРАВИЛА БЕЗОПАСНОСТИ:

- 1. Не начинайте работу без разрешения преподавателя.**
- 2. Не удаляйте программы и файлы, которые установлены в компьютере.**
- 3. Не открывайте чужие файлы и папки.**
- 4. При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.**
- 5. Сохраняйте свои файлы только в ваши папки.**
- 6. После выполнения работы верните первоначальный вид рабочего стола.**

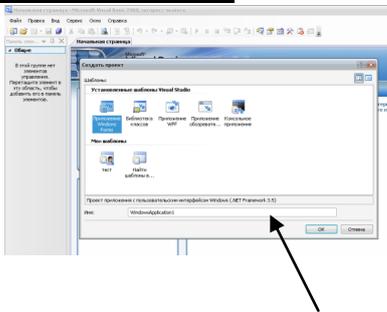
Ход работы.

Задания.

Составить программу, в которой будет решаться несколько задач с использованием оператора If.

Создание проекта для программы

- 1) В меню Windows **Пуск** выберите Visual Basic 2008, экспресс-выпуск.
- 2) Появится экран "Добро пожаловать в Visual Basic Express". Это интерфейс для Visual Basic 2008, экспресс-выпуск, также называемый *интегрированной средой разработки* или *IDE*.
- 3) В меню **Файл** выберите пункт **Создать проект**.
- 4) В диалоговом окне Создание проекта в области Шаблоны выберите **Приложение Windows**.



- 5) В поле Имя введите **Задачи Ваша Фамилия** и нажмите кнопку **ОК**.
- 6) Настройте общий вид формы: придайте ей имя «Решение задач», измените цвет фона по вашему усмотрению, установите свойство, которое при запуске программы будет помещать окно в центре экрана.

Text	Решение задач
BackColor	Любой цвет формы на ваше усмотрение
StartPosition	CenterScreen

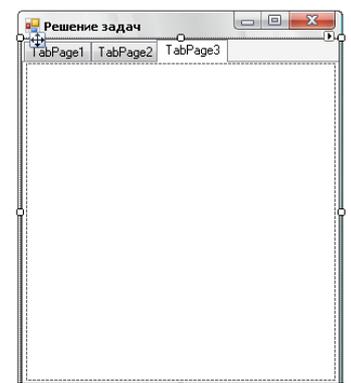
- 7) Из **Панели элементов** перетащите на форму TabControl растяните элемент на все форму.



- 8) Добавьте еще одну вкладку на данном элементе. Для этого необходимо щелкнуть правой клавишей мыши по строке с вкладками и выбрать пункт **Добавить вкладку**. (У вас появится формы с тремя вкладками на ней).

- 9) Настройте внешний вид первой вкладки формы.

Text	Задача №1
BackColor	Любой цвет формы на ваше усмотрение
Cursor	Настройте по своему усмотрению



10) Из панели элементов перетащите на форму элемент **Label**. Введите в эту метку условие задачи, настройте шрифт и цвет текста.

Text *Сравнить два числа. Если первое больше второго, то вывести их сумму, иначе – произведение. После решения задачи компьютер должен вывести надпись «ЗАДАЧА РЕШЕНА УСПЕШНО!»*

AutoSize False

TextAlign MiddleCenter (по середине в центре)

Font Размер, стиль текста установите по-вашему усмотрению

ForeColor Цвет текста

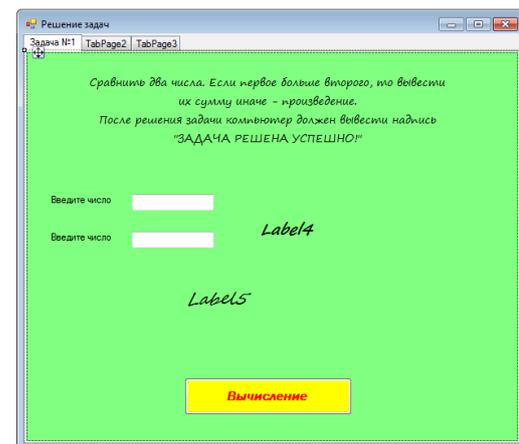
11) Установите на форме две пары элементов **Label2 – Textbox1; Label3 – Textbox2**. Элементам **Label2 ; Label3 –** присвойте текст **Введите число** (Свойство Text)

12) Установите на форме элементов **Label4, Label5**. Настройте общий вид этих элементов

Text Удалите надпись, элемент будет без названия

Font Размер -14, стиль Segoe Print

ForeColor Цвет текста любой



13) Установите на форме элемент **Button1** – присвойте имя «Вычисление»

14) Настройте общий вид кнопки **Button1** по вашему усмотрению

Text Вычисление

BackColor Любой цвет кнопки по вашему усмотрению.

Font Размер, стиль текста установите по-вашему усмотрению

ForeColor Цвет текста

Cursor Выберите вид курсора по вашему усмотрению.

15) Дважды щелкните элемент управления **Button1 (Вычисление)**, чтобы открыть редактор кода.

В процедуру события **Button1_Click** добавьте следующий код.

```
Dim A As Double = TextBox1.Text
```

```
Dim B As Double = TextBox2.Text
```

```
If A > B Then Label4.Text = "Сумма чисел=" & A + B Else Label4.Text = "Произведение чисел=" & A * B
```

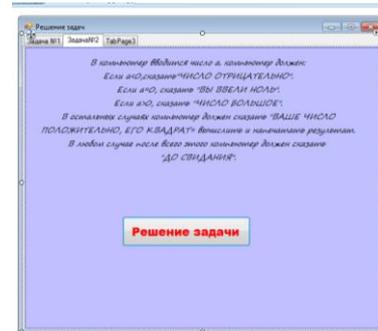
```
Label5.Text = "ЗАДАЧА РЕШЕНА УСПЕШНО!"
```

16) Запустить программу на исполнение .

17) Перейдите в форме на вкладку **TabPage2**.

18) Настройте внешний вид второй вкладки формы.

Text	Задача №2
BackColor	Любой цвет формы на ваше усмотрение
Cursor	Настройте по своему усмотрению



19) Из панели элементов перетащите на

форму элемент **Label**. Введите в эту метку условие задачи, настройте шрифт и цвет текста.

Text *В компьютер вводится число а. Компьютер должен:*

- Если $a < 0$, сказать «Число отрицательно».
- Если $a = 0$, сказать «Вы ввели нуль».
- Если $a > 100$, сказать «Число большое».
- В остальных случаях компьютер ничего не должен сказать «Ваше число положительное, его квадрат = вычислить и напечатать его квадрат».

В любом случае после всего этого компьютер должен сказать «До свидания».

AutoSize	False
TextAlign	MiddleCenter (по середине в центре)
Font	Размер, стиль текста установите по-вашему усмотрению
ForeColor	Цвет текста

20) Установите на форме элемент **Button2** – присвойте имя «Вычисление»

21) Настройте общий вид кнопки **Button2** по вашему усмотрению

Text Решение задачи

BackColor	Любой цвет кнопки по вашему усмотрению.
Font	Размер, стиль текста установите по-вашему усмотрению
ForeColor	Цвет текста
Cursor	Выберите вид курсора по вашему усмотрению.

22) Дважды щелкните элемент управления **Button2 (Решение задачи)**, чтобы открыть редактор кода.

В процедуру события **Button2_Click** добавьте следующий код.

```

Dim a As String

a = InputBox("Введите число")

If a < 0 Then
    MsgBox("Число отрицательно")
Elseif a = 0 Then
    MsgBox("Вы ввели нуль")
Elseif a > 100 Then
    MsgBox("Число большое")
Else
    MsgBox("Число положительное и его квадрат =" & a ^ 2)
End If

MsgBox("До свидания!") Запустить программу на исполнение .

```

23) **Запустить программу на исполнение**

24) Перейдите в форме на вкладку TabPage3.

25) Настройте внешний вид третьей вкладки формы.

26)	Text	Задача №3
	BackColor	Любой цвет формы на ваше усмотрение
	Cursor	Настройте по своему усмотрению

Попробуйте самостоятельно на третьей вкладке составить программу решения задачи:

В компьютер вводится число (пусть для конкретности это будет дальность какого-нибудь выстрела). Если оно находится в интервале от 28 до 30, то напечатать текст ПОПАЛ, иначе – НЕ ПОПАЛ.

27) Запустить программу на исполнение

Решение задачи №3

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click
```

```
Dim d As Double = InputBox("Введите дальность выстрела")
```

```
If d < 28 Or d > 30 Then
```

```
    MsgBox("НЕ ПОПАЛ")
```

```
Else
```

```
    If d = 28 Or d = 29 Or d = 30 Then MsgBox("ПОПАЛ")
```

```
End If
```

```
End Sub
```

Продемонстрируйте работу преподавателю!

28) Сохраните проект. В программе Visual Basic выполните команду Файл-Сохранить все. В открывшемся окне укажите имя и путь к вашей папке.

Лабораторная работа №6

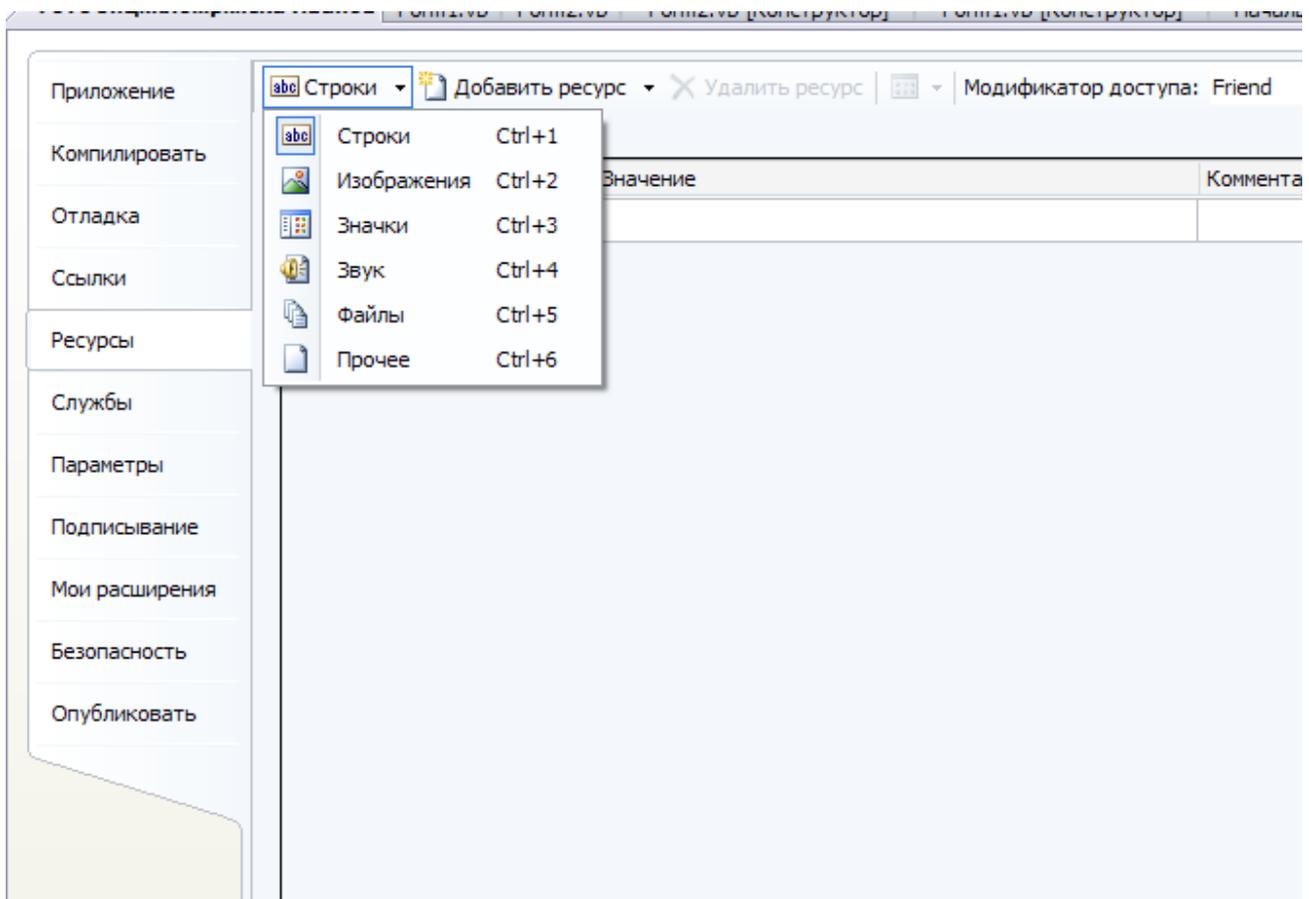
Тема: Использование графических элементов при составлении приложения.

Цель: научиться добавлять изображения в качестве ресурса в Visual Basic, использовать инструмент **PictureBox** для отображения рисунка, использовать несколько форм для создания приложения, применять свойство **Icon** для смены опознавательного значка программы.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

При составлении программ иногда приходится использовать различные рисунки и фотографии для этого сначала необходимо загрузить их в приложение. Для этого используется **Конструктор проектов**, где во вкладке ресурсы происходит загрузка графических, звуковых и др. элементов.



Элемент PictureBox.

Считается, что изображение стоит тысячи слов, и в действительности многие программы используют рисунки для обмена информацией. Существует несколько способов для отображения рисунков в Visual Basic, стандартный способ заключается в использовании элемента управления **PictureBox**.

Элемент управления **PictureBox** действует как контейнер для картинок. Можно выбрать изображение для вывода, присвоив значение свойству **Image**. **Свойство Image** может быть установлено в окне **Свойства** или в коде программы, указывая на рисунок, который следует отображать.

Элемент управления **PictureBox** содержит и другие полезные свойства, в том числе: **AutoSize** определяющее, будет ли изображение растянуто в элементе **PictureBox**, и **SizeMode**, которое может использоваться для растягивания, центрирования или увеличения изображения в элементе управления **PictureBox**.

Перед добавлением рисунка к элементу управления PictureBox в проект обычно добавляется файл рисунка в качестве ресурса. После добавления ресурса к проекту можно повторно использовать его. Например, может потребоваться отображение одного и того же изображения в нескольких местах.

Элемент управления PictureBox также поддерживает отображение изображений в качестве фона для формы. **Свойство BackgroundImage** формы используется для отображения изображения, которое находится за любыми другими элементами управления на форме, что напоминает фоновый рисунок на рабочем столе Windows.

Так же, как и в Windows, можно расположить изображение по центру, замостить или растянуть его на весь экран. В форме для этих целей используется **свойство BackgroundImageLayout**.

Использование нескольких форм в приложении.

В одном приложении (программе) может быть использовано несколько форм (Проект -Добавить форму) Переключение между формами можно использовать различными элементами например кнопкой (элемент Button). Для переключения нужно обязательно для элемента прописать код в редакторе кодов.

ПРАВИЛА БЕЗОПАСНОСТИ:

- 1. Не начинайте работу без разрешения преподавателя.**
 - 2. Не удаляйте программы и файлы, которые установлены в компьютере.**
 - 3. Не открывайте чужие файлы и папки.**
 - 4. При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.**
 - 5. Сохраняйте свои файлы только в ваши папки.**
 - 6. После выполнения работы верните первоначальный вид рабочего стола.**
- Ход работы.**

Задание.

Составить фото - энциклопедию.

Добавление изображения в качестве ресурса

1) В меню **Файл** выберите **Создать проект**.
 2) В диалоговом окне **Создание проекта** в области **Шаблоны** выберите **Приложение Windows**.

3) В поле **Имя** введите **Фото энциклопедия Уссурийска Ваша фамилия** и нажмите кнопку **ОК**.

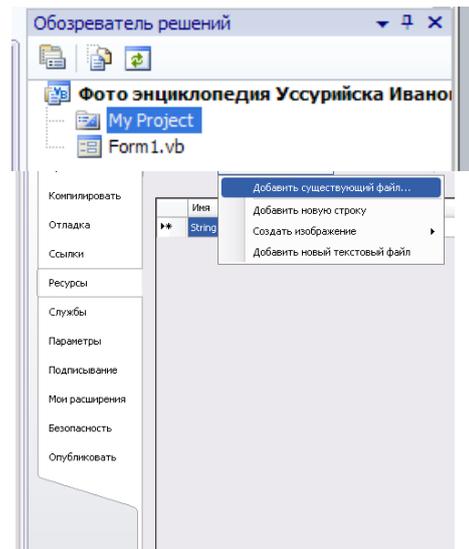
Откроется новый проект Windows Forms.

4) В обозревателе решений дважды щелкните узел **Мой проект (My Project)**, чтобы открыть **конструктор проектов**.

5) В **конструкторе проектов** перейдите на вкладку **Ресурсы**.

6) Выберите команду **Добавить ресурс**, а затем выберите пункт **Добавить существующий файл** в раскрывающемся списке.

Откроется диалоговое окно **Добавить существующий файл к ресурсам**. Перейдите в папку, содержащую изображения (Мои документы – Мои рисунки – Фото Уссурийска).



7) Выделите файл изображения (с расширением bmp, gif или jpg) и нажмите кнопку **Открыть**. Изображение добавляется в проект и будет отображаться в окне **диспетчера ресурсов**.

8) Повторите предыдущие два шага, чтобы добавить оставшиеся изображения папки Фото Уссурийска к проекту.

9) В меню **Файл** выберите **Заккрыть**. Если требуется сохранить изменения, нажмите кнопку **Да**.

Настройка общего вида формы.

10) Настройте общий вид формы: придайте ей имя «Фото энциклопедия», измените значок программы по вашему усмотрению, установите свойство, которое при запуске программы будет помещать окно в центре экрана.

Text Фото энциклопедия

Icon В папке Мои рисунки-Иконки - выберите значок по своему усмотрению

StartPosition CenterScreen

Отображение фонового изображения на форме

11) В окне Form1.vb[Конструктор]* выберите свойство, которое позволяет устанавливать фоновое изображение на форме.

Для этого в окне **Свойства** рядом со свойством **BackgroundImage** нажмите кнопку , чтобы открыть диалоговое окно **Выбор ресурса**.

12) В списке **Элементы** выберите одно из добавленных изображений и нажмите кнопку **ОК**.

Обратите внимание, что рисунок отображаемый на форме не помещается весь.

13) Выберите свойство **BackgroundImageLayout** и присвойте ему значение **Stretch**.

Обратите внимание, что изображение растягивается и заполняет всю форму.

14) Перетащите на форму элемент Label и настройте его свойства.

Text «Фото энциклопедия города Уссурийск»

Font Шрифт, начертание, размер выбрать по своему усмотрению

ForeColor Цвет шрифта выбрать по своему усмотрению

AutoSize False

TextAlign Выравнивание текста по центру

15) Перетащите на форму два элемента Button1 и Button2 и настройте их свойства.

Text **Ночной город**

Font Шрифт, начертание, размер выбрать по своему усмотрению

ForeColor Цвет шрифта выбрать по своему усмотрению

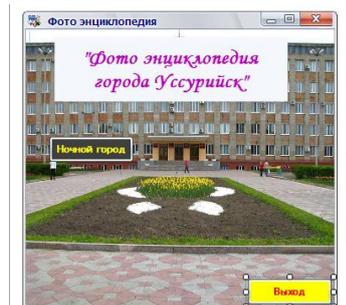
BackColor Цвет кнопки по своему усмотрению

Text **Выход**

Font Шрифт, начертание, размер выбрать по своему усмотрению

ForeColor Цвет шрифта выбрать по своему усмотрению

BackColor Цвет кнопки по своему усмотрению



Создание дополнительной формы.

16) В пункте меню **Проект** выбрать **Добавить форму Windows – Форма Windows Form – Добавить**. Появится дополнительная вторая форма.

17) Настройте общий вид формы 2: придайте ей имя «Ночной Уссурийск», измените значок программы по вашему усмотрению, установите свойство, которое при запуске программы будет помещать окно в центре экрана.

Text	Ночной Уссурийск
Icon	В папке Мои рисунки-Иконки - выберите значок по своему усмотрению
StartPosition	CenterScreen
BackColor	Любой цвет формы на ваше усмотрение

Отображение изображений с помощью элемента управления PictureBox на Form2

18) В **обозревателе решений** выберите **Form2.vb**, а затем в меню **Вид** выберите **Конструктор**.

19) С **панели элементов** перетащите в форму элемент управления PictureBox.

20) В окне **Свойства** нажмите кнопку рядом со свойством Image, чтобы открыть диалоговое окно **Выбор ресурса**.

21) В списке **Элементы** выберите одно из добавленных изображений ночного города и нажмите кнопку **ОК**.

22) Выберите свойство **SizeMode** и присвойте ему значение **StretchImage**.

23) Дважды щелкните элемент управления PictureBox, чтобы открыть обработчик событий **PictureBox1_Click** в редакторе кода.

24) В обработчик событий **PictureBox1_Click** добавьте следующий код.

```
PictureBox1.Image = My.Resources.MyPictureName2
```

Необходимо заменить "MyPictureName2" на фактическое имя второго изображения, которое было добавлено ранее.

25) В **обозревателе решений** выберите **Form2.vb**, а затем в меню **Вид** выберите **Конструктор**.

26) Добавьте на форму второй элемент управления PictureBox.

Image Выберите нужный рисунок

SizeMode StretchImage

27) Дважды щелкните второй элемент управления PictureBox, чтобы открыть обработчик событий **PictureBox2_Click** в редакторе кода.

28) В обработчик событий **PictureBox2_Click** добавьте следующий код.

```
PictureBox2.Image = My.Resources.MyPictureName2
```

Необходимо заменить "MyPictureName2" на фактическое имя второго изображения, которое было добавлено ранее.

29) В **обозревателе решений** выберите **Form2.vb**, а затем в меню **Вид** выберите **Конструктор**.

30) Перетащите на форму элемент Label и настройте его свойства.

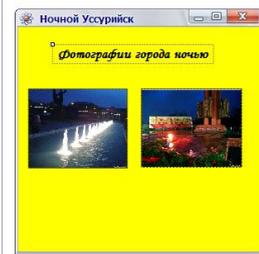
Text «Фотографии города ночью»

Font Шрифт, начертание, размер выбрать по своему усмотрению

ForeColor Цвет шрифта выбрать по своему усмотрению

AutoSize False

TextAlign Выравнивание текста по центру



Настройка взаимосвязи двух форм.

31) В **обозревателе решений** выберите **Form1.vb**, а затем в меню **Вид** выберите **Конструктор**.

32) Дважды щелкните элемент управления Button «Ночной город», чтобы открыть обработчик событий **Button1_Click** в редакторе кода.

33) В обработчик событий **Button1_Click** добавьте следующий код.
`Form2.Show()`.

34) В **обозревателе решений** выберите **Form1.vb**, а затем в меню **Вид** выберите **Конструктор**.

35) Дважды щелкните элемент управления Button «Ночной город», чтобы открыть обработчик событий **Button2_Click** в редакторе кода.

36) В обработчик событий **Button2_Click** добавьте следующий код.
`End`

37) Нажмите клавишу F5 для запуска приложения. Когда появится форма, щелкните изображение, чтобы появился второй рисунок.

Продемонстрируйте вашу работу преподавателю.

38) ***Дополнительное задание:** Создайте еще одну (несколько) кнопку в вашей программе, при нажатии на которую будет открываться новая форма с фотографиями города.

39) Сохраните проект. В программе Visual Basic выполните команду Файл-Сохранить все. В открывшемся окне укажите имя и путь к вашей папке

Лабораторная работа №7

Тема: Использование графических методов для рисования объектов на форме.

Цель: научиться применять команды рисования основных фигур при создании приложения.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

Основы графики.

Прежде чем начать рисовать, необходимо знать несколько вещей. Экран компьютера состоит из тысяч маленьких точек, известных как *пиксели* — путем задания цвета каждой точки программа определяет, что выводить на экран. Конечно, большая часть этой работы уже сделана кодом, который определяет формы и элементы управления.

Можно представить форму как полотно, на котором можно нарисовать что-нибудь — как и настоящее полотно, форма имеет размер. В то время как настоящее полотно измеряется в дюймах или в сантиметрах, форма измеряется в точках. Система *координат* определяет, где расположена каждая точка, при помощи *координаты X*, изменяющейся слева направо, и *координаты Y*, изменяющейся сверху вниз.

Координаты отсчитываются с левого верхнего угла формы, поэтому если необходимо нарисовать одну точку в десятой точке слева и десятой точке внизу, нужно задать координаты X и Y как **10, 10**.

Точки также используются для определения ширины и высоты рисунка. Чтобы задать квадрат, составляющий 100 точек в ширину и 100 точек в высоту, с верхним левым углом, находящимся на расстоянии 10 точек слева и 10 точек вниз, необходимо записать его координаты как **10, 10, 100, 100**.

Процедура размещения изображения на экране называется *рисованием*. Формы и элементы управления имеют событие **Paint**, происходящее каждый раз, когда они должны быть перерисованы, например, при первом отображении формы или когда форма закрывается другим окном. Любой код, который пишется для отображения графики, обычно размещается в обработчике событий **Paint**.

Рисование линии

Чтобы нарисовать линию через форму, необходимо определить две вещи — ее координаты и цвет. Как было сказано выше, координаты X и Y выражаются в точках. Для линии необходимо задать координаты двух точек — начальной точки, а затем конечной.

Так же, как при рисовании линии на листе бумаги используется ручка, Visual Basic использует объект **Pen** для рисования на форме. Объект **Pen** определяет внешний вид линии — в данном случае, цвет. В следующей процедуре нужно будет нарисовать горизонтальные, вертикальные и диагональные линии на форме.

Рисование простых фигур

Рисование фигуры похоже на рисование линии: необходимо определить координаты и цвет для рисования. Как линия определяется координатами своих начала и конца, так и фигура, например круг или прямоугольник, определяется координатами верхнего левого и правого нижнего углов, шириной и высотой.

Круги и овалы (также называемые *эллипсами*) не имеют верхних левых углов, поэтому используются координаты, описывающие ограничивающие их прямоугольники, — воображаемые прямоугольники той же ширины и высоты, что круг или овал.

Рисование закрашенных фигур

Нарисованные на данный момент фигуры представляют собой только контуры. Для рисования закрашенных фигур необходимо использовать один из методов *заливки*, например **FillRectangle** или **FillEllipse**. В методах заливки используется еще один графический объект для рисования – **Brush**.

Для заливки фигуры другим цветом необходимо определить координаты точки, лежащей внутри фигуры, иначе могут быть закрашены границы. Например, для заполнения квадрата с координатами 0, 0, 150, 150 необходимо указать для заливки точку с координатами 1, 1, 148, 148 (учитывается, что толщина линии составляет одну точку).

Рисование текста

Чтобы нарисовать текст на форме или элементе управления, используйте графический метод **DrawString**. Подобно другим графическим методам, **DrawString** принимает в качестве параметров объект **Brush**, определяющий цвет, и координаты, указывающие, где рисовать текст — в данном случае X и Y координаты левого верхнего угла прямоугольника, ограничивающего текст.

Метод **DrawString** также имеет два дополнительных аргумента — строку, которую требуется нарисовать, и шрифт, который определяет, каким образом будет

выглядеть текст. Чтобы указать шрифт, необходимо сначала создать объект **Font**, а затем использовать этот объект в качестве аргумента метода **DrawString**.

ПРАВИЛА БЕЗОПАСНОСТИ:

1. **Не начинайте работу без разрешения преподавателя.**
2. **Не удаляйте программы и файлы, которые установлены в компьютере.**
3. **Не открывайте чужие файлы и папки.**
4. **При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.**
5. **Сохраняйте свои файлы только в ваши папки.**
6. **После выполнения работы верните первоначальный вид рабочего стола.**
7. **Не пытайтесь самостоятельно устранять неисправности в работе компьютера.**

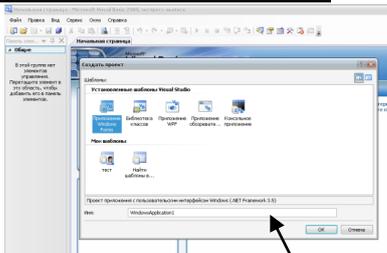
Ход работы.

Задание.

Составить программу, в которой на форме будет создаваться графическое изображение.

Создание проекта для программы

- 1) В меню Windows **Пуск** выберите Visual Basic 2008, экспресс-выпуск.
- 2) В меню **Файл** выберите пункт **Создать проект**.
- 3) В диалоговом окне Создание проекта в области Шаблоны выберите **Приложение Windows**.



- 4) В поле Имя введите **Графические элементы Ваша Фамилия** и нажмите кнопку **ОК**.

Создание интерфейса приложения.

- 5) Настройте общий вид формы: придайте ей имя «Создание графического изображения», измените, цвет фона по вашему усмотрению, установите свойство, которое при запуске программы будет помещать окно в центре экрана.

Text

Создание графического изображения



BackColor	Любой цвет формы на ваше усмотрение
StartPosition	CenterScreen
Icon	В папке Мои рисунки-Иконки - выберите значок по своему усмотрению
Cursor	Любой значок курсора

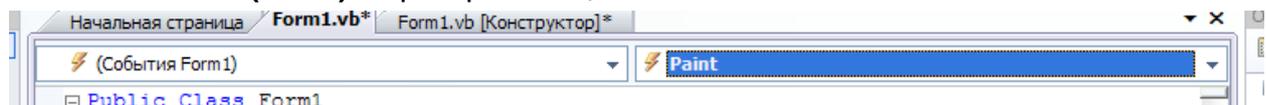
6) Установите на форме кнопку **Выход** (элемент **Button**)

Text	Выход
Font	Шрифт, начертание, размер выбрать по своему усмотрению
ForeColor	Цвет шрифта выбрать по своему усмотрению
BackColor	Цвет кнопки по своему усмотрению

7) Пропишите код для кнопки Выход (End).

Написание кода создания графического изображения.

8) Дважды щелкните форму, чтобы открыть редактор кода, а затем выберите **Рисование (Paint)** из раскрывающегося списка **События**.



9) В обработчике событий **Form1_Paint** добавьте следующий код.

'Рисование куба

'Рисование квадрата1

```
e.Graphics.DrawRectangle(Pens.Blue, 60, 60, 130, 130)
```

```
e.Graphics.FillRectangle(Brushes.Khaki, 61, 61, 129, 129)
```

10) Запустите код на исполнение F5

(В данном коде Blue, Khaki – это цвет линии и цвет закрашивания квадрата соответственно, эти цвета вы можете изменить по своему усмотрению)

11)Вернитесь в редактор кода приложения и введите следующий код.

'Рисование квадрата2

```
e.Graphics.DrawRectangle(Pens.Blue, 20, 100, 130, 130)
```

e.Graphics.FillRectangle(Brushes.Khaki, 21, 101, 129, 129)

(Цвет можете изменить по своему усмотрению)

12) Запустите код на исполнение F5.

13) Вернитесь в редактор кода приложения и введите следующий код.

'Рисование линии1

e.Graphics.DrawLine(Pens.Blue, 60, 60, 20, 100)

'Рисование линии2

e.Graphics.DrawLine(Pens.Blue, 190, 60, 150, 100)

'Рисование линии3

e.Graphics.DrawLine(Pens.Blue, 190, 190, 150, 230)

14) Запустите код на исполнение F5.

15) Вернитесь в редактор кода приложения и введите следующий код.

'Рисование конуса

'Рисование овала

e.Graphics.DrawEllipse(Pens.Red, 250, 200, 100, 50)

'Рисование линий

e.Graphics.DrawLine(Pens.Red, 300, 100, 250, 220)

e.Graphics.DrawLine(Pens.Red, 300, 100, 350, 220)

e.Graphics.DrawLine(Pens.Red, 300, 100, 270, 245)

e.Graphics.DrawLine(Pens.Red, 300, 100, 330, 245)

e.Graphics.DrawLine(Pens.Red, 300, 100, 300, 250)

(Цвет можете изменить по своему усмотрению)

16) Запустите код на исполнение F5.

17) Вернитесь в редактор кода приложения и введите следующий код.

'Рисование текста

Dim aFont As New System.Drawing.Font("Arial", 20, FontStyle.Underline)

e.Graphics.DrawString("Геометрические фигуры", aFont, Brushes.OrangeRed, _
30, 10)

e.Graphics.DrawString("конус!", aFont, Brushes.Turquoise, 250, 260)

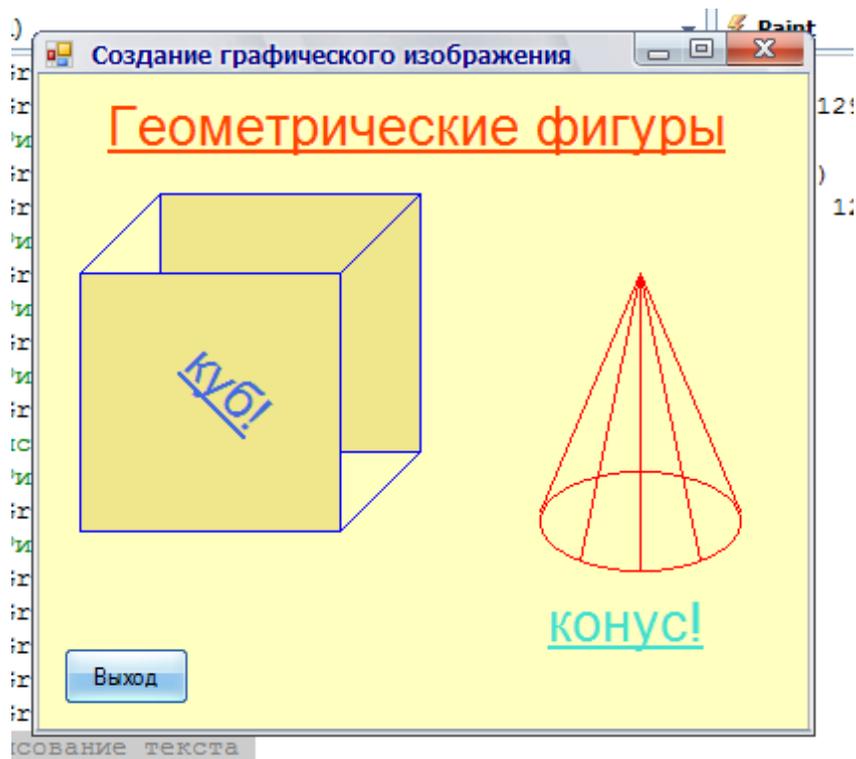
' Рисование текста повернутого под углом 45градусов.

```
e.Graphics.RotateTransform(45)
```

```
e.Graphics.DrawString("куб!", aFont, Brushes.RoyalBlue, 150, 30)
```

18) Запустите код на исполнение F5.

Продемонстрируйте вашу работу преподавателю.



19) ***Дополнительное задание:** Попробуйте самостоятельно нарисовать еще одну геометрическую фигуру цилиндр и подпишите используя графический способ ввода текста.

20) Сохраните проект. В программе Visual Basic выполните команду Файл-Сохранить все. В открывшемся окне укажите имя и путь к вашей папке

Лабораторная работа №8

Тема: Анимация в языке Visual Basic.

Цель: научиться настраивать движение для графических объектов.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

Анимация означает придание неподвижному предмету движения. Еще одно значение слова анимация – мультфильм.

Для того чтобы придать движение объекту необходимо использовать элемент управления **Timer**.



Таймер *Timer1* – объект, вся работа которого заключается в том, чтобы через каждые *Interval* миллисекунд создавать событие (импульс), которое запускает процедуру *Timer1_Tick*.

Свойство таймера **Enabled**:

-**False**- таймер срабатывает при нажатии на кнопку;

- **True**. таймер выдает импульсы сразу после запуска программы.

Свойство **Interval** задает промежуток времени (частоту импульсов) в миллисекундах (мс – тысячных долях секунды).

Несколько таймеров. Таймеров в проекте может быть несколько. Все они работают независимо друг от друга и каждый имеет свою собственную процедуру, которая реагирует только на его импульсы.

ПРАВИЛА БЕЗОПАСНОСТИ:

1. **Не начинайте работу без разрешения преподавателя.**
2. **Не удаляйте программы и файлы, которые установлены в компьютере.**
3. **Не открывайте чужие файлы и папки.**
4. **При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.**
5. **Сохраняйте свои файлы только в ваши папки.**
6. **После выполнения работы верните первоначальный вид рабочего стола.**
7. **Не пытайтесь самостоятельно устранять неисправности в работе компьютера.**

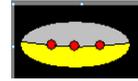
Ход работы.

Задание

Составить программу, движения изображения.

Создание графического элемента в граф. редакторе.

а) Откройте граф. редактор Paint.



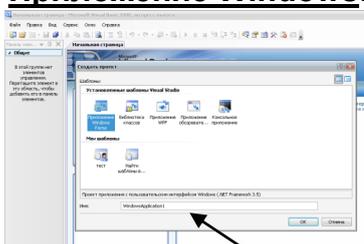
б) Рабочее полотно редактора уменьшите до размеров почтовой марки.

в) На черном фоне нарисуйте маленький летающий объект.

г) Сохраните рисунок в формате *нло.png*.

Создание проекта для программы

- 1) В меню Windows **Пуск** выберите Visual Basic 2008, экспресс-выпуск.
- 2) В меню **Файл** выберите пункт **Создать проект**.
- 3) В диалоговом окне Создание проекта в области Шаблоны выберите **Приложение Windows**.



4) В поле Имя введите **Анимация Ваша Фамилия** и нажмите кнопку **ОК**.

5) **Загрузи картинку *нло.png* в ресурс программы.**

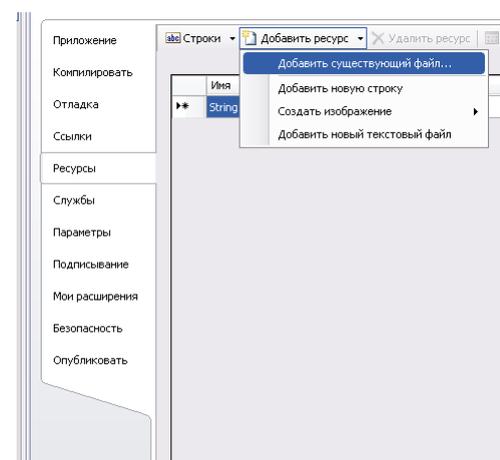
Для этого в **обозревателе решений** дважды щелкните узел **Мой проект (My Project)**, чтобы открыть конструктор проектов.

В конструкторе проектов перейдите на вкладку **Ресурсы** - **Добавить ресурс** - **Добавить существующий файл**.

Откроется диалоговое окно **Добавить существующий файл к ресурсам**. Перейдите в папку, содержащую созданное вами изображение.

Выделите файл изображения) и нажмите кнопку **Открыть**.

Изображение добавляется в проект и будет отображаться в окне **диспетчера ресурсов**.



11) Запустите код на исполнение F5

Создание звездного неба.

12) Вернитесь в конструктор формы.

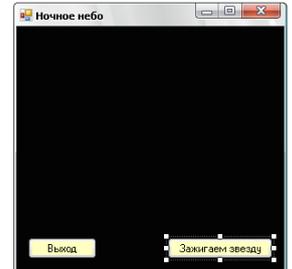
13) Установите на форме кнопку **Зажигаем звезду** (элемент Button)

Text **Зажигаем звезду**

Font Шрифт, начертание, размер выбрать по своему усмотрению

ForeColor Цвет шрифта выбрать по своему усмотрению

BackColor Цвет кнопки по своему усмотрению



14) Дважды щелкните по кнопке, чтобы открыть редактор кода, добавьте следующий код в процедуру Button2_Click:

Private Sub Button2_Click(.....

Граф.DrawEllipse(Pens.White, 300 * Rnd(), 200 * Rnd(), 3, 3)

End Sub

15) Объявите переменную Граф для этого пропишите строку в процедуре **Public Class** Form1

Dim Граф **As** Graphics

16) В процедуре Form1_Load пропишите строку

Private Sub Form1_Load(...

Граф = Me.CreateGraphics

End Sub

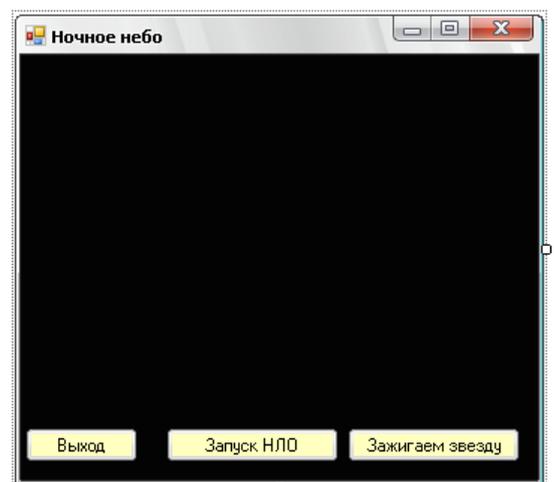
17) Запустите программу на исполнение (при нажатии на кнопку на небе должны появляться звезды).

18) Вернитесь в конструктор формы.

19) Установите на форме элемент **PictureBox**

Image Выберите нужный рисунок

SizeMode StretchImage



20) Установите на форме кнопку **Запуск НЛО** (элемент Button)

Text **Запуск НЛО**

Font Шрифт, начертание, размер выбрать самостоятельно

ForeColor Цвет шрифта выбрать по своему усмотрению

Настройка Таймера (движения).

21) Установите на форме элемент **Timer1**  и задайте свойства:

Enabled False

GenerateMember True

Interval 50

22) Дважды щелкните по элементу Timer1.

В редакторе кода в процедуре **Private Sub** Timer1_Tick пропишите код:

```
PictureBox1.Left = PictureBox1.Left + 1
```

```
PictureBox1.Left = x
```

```
x = x + Шаг
```

```
If x > Width Then Шаг = -2
```

23) Объявите переменные Шаг, x, для этого пропишите строки к уже имеющейся в процедуре **Public Class** Form1

```
Dim Шаг As Integer = 2
```

```
Dim x As Integer = 0
```

24) Дважды щелкните по кнопке **Запуск НЛО**.

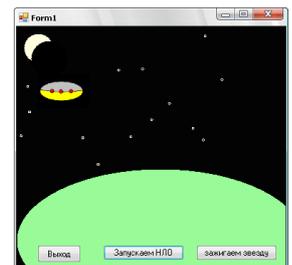
В редакторе кода в процедуре **Private Sub** Button3_Click введите код

```
Timer1.Enabled = True
```

25) Запустите программу на исполнение. Продемонстрируйте работу преподавателю.

26) ***Дополнительное задание.** Попробуйте нарисовать на форме еще один движущийся элемент.

27) Сохраните проект. В программе Visual Basic выполните команду Файл-Сохранить все. В открывшемся окне укажите имя и путь к вашей папке



Лабораторная работа №9

Тема: Использование флажков и переключателей.

Цель: научиться использовать флажки и переключатели для представления и извлечения выбора пользователя.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть:

При создании пользовательского интерфейса для программы часто необходим способ для предоставления вариантов выбора. Например, предположим, что было написано приложение по приему заказов в пиццерии — может потребоваться разрешение для пользователей выбрать любую из множества начинок. Элемент управления **CheckBox** обеспечивает визуальное представление, что облегчает представление вариантов выбора для этой возможности.

Элемент управления **CheckBox** состоит из надписи и поля, в котором пользователь может сделать выбор. Когда пользователь щелкает на поле, в нем отображается знак отметки. Если щелкнуть на поле еще раз, этот знак будет удален. Состояние флажка можно получить с помощью свойства **CheckBox.Checked**. Если поле отображает метку, это свойство возвращает **True**. Если знак отметки не отображается, свойство возвращает **False**.

Использование переключателей для предоставления возможности выбора единственного варианта.

Если нужно разрешить пользователю выбрать только один из нескольких параметров, то в этом случае можно использовать элемент управления **RadioButton**.

В отличие от флажков, переключатели (также называемые дополнительными кнопками) всегда работают в составе группы. Выбор одного переключателя немедленно удаляет содержимое всех остальных переключателей в группе. Определяя группу переключателей, разработчик формы предлагает пользователю набор вариантов, из которых может быть выбран только один.

Можно использовать группы элементов управления **RadioButton**, чтобы пользователи могли выбирать между исключаящими друг друга параметрами.

Например, можно позволить пользователям выбирать для пиццы обычный соус или острый, но не оба одновременно. Как и в случае с элементом управления **CheckBox**, можно получить информацию о статусе элемента управления **RadioButton** из свойства **RadioButton.Checked**.

ПРАВИЛА БЕЗОПАСНОСТИ:

- 1. Не начинайте работу без разрешения преподавателя.**
- 2. Не удаляйте программы и файлы, которые установлены в компьютере.**
- 3. Не открывайте чужие файлы и папки.**
- 4. При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.**
- 5. Сохраняйте свои файлы только в ваши папки.**
- 6. После выполнения работы верните первоначальный вид рабочего стола.**
- 7. Не пытайтесь самостоятельно устранять неисправности в работе компьютера.**

Ход работы.

Задание

Составить программу с элементами выбора.

Использование флажков

- 1) В меню **Файл** выберите **Создать проект**.
- 2) В диалоговом окне **Создание проекта** в области **Шаблоны** выберите

Приложение Windows.

- 3) В поле **Имя** введите **Переключатели** и нажмите кнопку **ОК**.

Откроется новый проект Windows Forms.

- 4) Из **Панели элементов** перетащите **Button** и три элемента управления **CheckBox** на форму.

- 5) В окне **Свойства** измените свойство **Text** для **CheckBox1**, **CheckBox2** и **CheckBox3** для чтения **Автобус**, **Поезд** и **Самолет** соответственно.

- 6) В окне **Свойства** измените свойство **Text** элемента **Button1** на текст **Выбрать**.

- 7) В форме дважды щелкните кнопку; откроется окно обработчика событий **Button1_Click** в редакторе кода.

- 8) В обработчик событий **Button1_Click** добавьте следующий код:

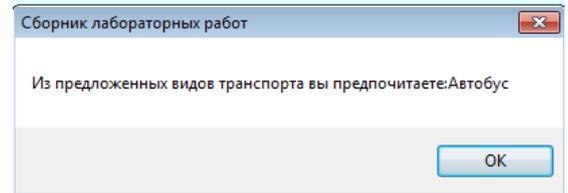
```
Dim toppings As String = ""
If CheckBox1.Checked = True Then
    toppings &= "Автобус "
End If
If CheckBox2.Checked = True Then
    toppings &= "Поезд "
End If
If CheckBox3.Checked = True Then
```

```

toppings &= "Самолет"
End If
If toppings <> "" Then
    MsgBox("Из предложенных видов транспорта вы предпочитаете:" & toppings)
End If

```

9) Нажмите клавишу F5 для запуска приложения. При появлении формы выберите начинку и нажмите кнопку. Отображается окно сообщения, показывающее выбранный вами транспорт.



Использование переключателей

1. Из **Панели элементов** перетащите на форму еще один элемент Button и три элемента управления RadioButton.

2. В окне **Свойства** установите свойство **Text** для элемента **RadioButton1** в **Вагон купе**.

3. Установите свойство **Checked** для **RadioButton1** как **True**.

4. В окне **Свойства** установите свойство **Text** для элемента **RadioButton2** равным **Плацкарта**.

5. В окне **Свойства** установите свойство **Text** для элемента **RadioButton3** равным **Общий вагон**.

6. В окне **Свойства** установите свойство **Text** для элемента **Button2** равным **Заказать билет**.

7. В форме дважды щелкните кнопку, чтобы открыть окно обработчика событий **Button2_Click** в редакторе кода.

8. В обработчик событий **Button2_Click** добавьте следующий код:

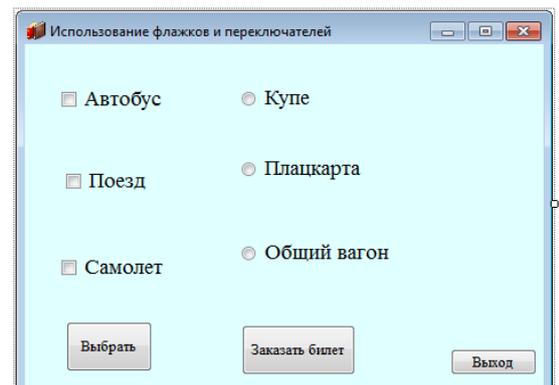
```

If RadioButton1.Checked = True Then
    MsgBox("Вы выбрали вагон купе")
ElseIf RadioButton2.Checked = True Then
    MsgBox("Вы выбрали место в плацкартном вагоне")
Else
    MsgBox("Вы выбрали место в общем вагоне")
End If

```

9. Нажмите клавишу F5 для запуска приложения. Выберите один из переключателей и нажмите кнопку **Заказать билет**. Появится окно сообщения, отображающее ваш выбор.

10. Попробуйте выбрать два переключателя в одно и то же время. Обратите внимание, что это невозможно. Сразу же после нажатия кнопки отметка в другой удаляется автоматически.



11. Продемонстрируйте вашу работу преподавателю!

12. Сохраните ваш проект.

Лабораторная работа №10

Тема: Создание меню для программы.

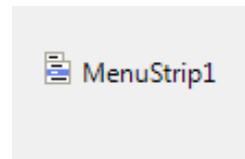
Цель: научиться создавать пункты меню в проекте..

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Теоретическая часть.

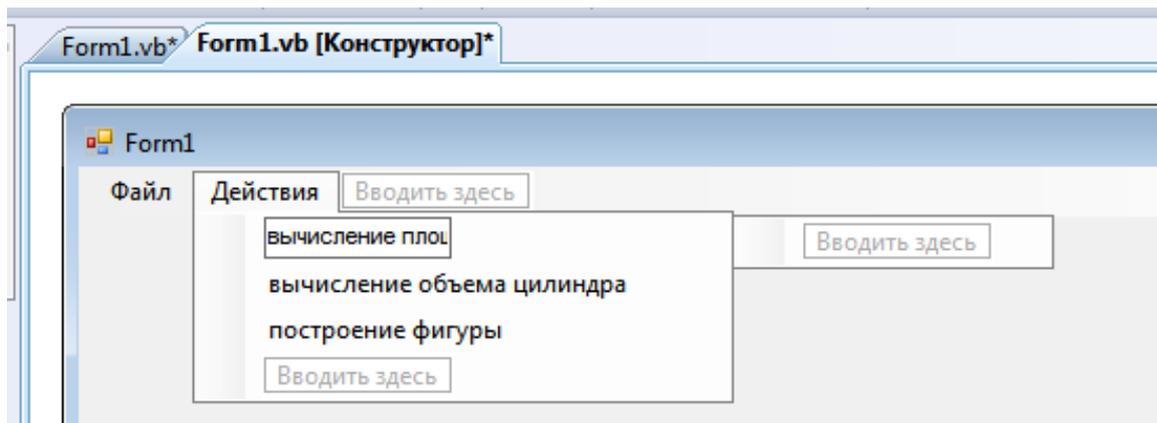
Любое приложение Windows имеет свои пункты меню.

Для создания пунктов меню в проекте используется элемент **Menu Strip**



Чтобы создать меню, нужно найти элемент **Menu Strip** и разместить его на форме. Однако элемент оказывается не на форме, а под ней. Эта область, где расположен значок, называется **дизайнером компонентов (Component Designer)**. VB помещает на дизайнер компонентов те элементы, которым во время работы проекта ни к чему быть видимыми.

Однако, одновременно на том же рисунке появляется проект нашего будущего меню: в левой верхней части формы, там, где обычно и размещается Главное меню, появилось белое поле со словами «Вводить здесь». Вот сюда то и вводятся основные команды пунктов меню.



ПРАВИЛА БЕЗОПАСНОСТИ:

1. Не начинайте работу без разрешения преподавателя.
2. Не удаляйте программы и файлы, которые установлены в компьютере.
3. При выполнении лабораторной работы строго соблюдайте инструкции преподавателя.
4. Сохраняйте свои файлы только в ваши папки.
5. После выполнения работы верните первоначальный вид рабочего стола.

Задание

Составить программу вычисления полной площади и объема цилиндра используя пункты меню.

- 1) В меню **Файл** выберите **Создать проект**.
- 2) В диалоговом окне **Создание проекта** в области **Шаблоны** выберите **Приложение Windows**.
- 3) В поле **Имя** введите **Создание меню** и нажмите кнопку **ОК**.
Откроется новый проект Windows Forms.
- 4) Настройте общий вид формы: придайте ей имя «Вычисление площади и объема цилиндра», измените цвет фона белый, установите свойство, которое при запуске программы будет помещать окно в центре экрана.

Text	Вычисление площади и объема цилиндра
BackColor	Белый
StartPosition	CenterScreen
- 5) Установите на форме две пары элементов **Label1 – Textbox1; Label2 – Textbox2**. Элементам **Label1 ; Label2 –** присвойте текст **Введите радиус цилиндра; Введите высоту цилиндра** (Свойство Text)

Из **Панели элементов** перетащите **Menu Strip** на форму.

6) Введите пункт меню **Файл** подпункты **Очистить форму; Выход**

7) Введите второй пункт меню **Действия** подпункты **Вычисление площади цилиндра; вычисление объема цилиндра; построение фигуры.**

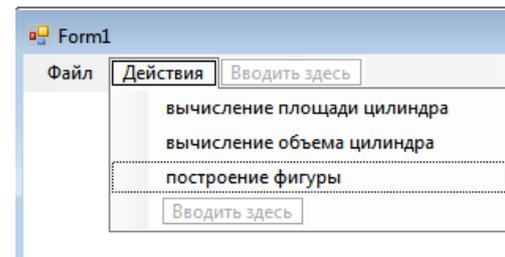
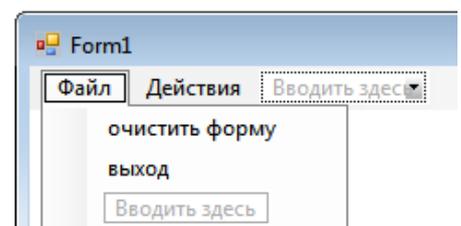
8) Настройте действия для каждого пункта меню.

9) Дважды щелкните по форме, в редакторе кода пропишите процедуру объявления переменных.

```
Public Class Form1
    Dim Граф As Graphics
    Dim r, h As Single
    Dim L, Socн, V, Сбок, Сполн As Double
```

```
Private Sub Form1_Load(...)
    Граф = Me.CreateGraphics
```

```
End Sub
```



10) Дважды щелкните по пункту меню **Очистить** в открывшемся редакторе кода введите процедуру

```
Private Sub ВыходToolStripMenuItem_Click(...)
    'Стираем с формы:
    Граф.Clear(Color.White)
End Sub
```

11) Дважды щелкните по пункту меню **Выход** в открывшемся редакторе кода введите процедуру

```
Private Sub ВыходToolStripMenuItem1_Click(...)
    End
End Sub
```

12) Дважды щелкните по пункту меню **Вычисление объема цилиндра** в открывшемся редакторе кода введите процедуру

```
Private Sub ВычислениеОбъемаЦилиндраToolStripMenuItem_Click(...)
    r = TextBox1.Text
    h = TextBox2.Text
    L = 2 * Math.PI * r
    Socн = Math.PI * r ^ 2
    V = Socн * h
    Граф.DrawString("Объем цилиндра = " & Format(V, "0.000"),
    Me.Font, Brushes.Black, 250, 60)
End Sub
```

13) Дважды щелкните по пункту меню **Вычисление площади цилиндра** в открывшемся редакторе кода введите процедуру

```
Private Sub ВычислениеПлощадиToolStripMenuItem_Click(...)
    Sбок = L * h
    Сполн = Sбок + 2 * Socн
    Граф.DrawString("Полная площадь поверхности цилиндра = " &
    Format(Сполн, "0.000"),
    Me.Font, Brushes.Black, 250, 40)
End Sub
```

14) Дважды щелкните по пункту меню **Построение фигуры** в открывшемся редакторе кода введите процедуру

```
Private Sub ПостроениеФигурыToolStripMenuItem_Click(...)
    'Рисуем цилиндр:
    r = TextBox1.Text
    h = TextBox2.Text
    Граф.DrawEllipse(Pens.Black, 200, 100, 2 * r, r)
    Граф.DrawEllipse(Pens.Black, 200, 100 + h, 2 * r, r)
    Граф.DrawLine(Pens.Black, 200, 100 + r / 2, 200, 100 + r / 2 + h)
    Граф.DrawLine(Pens.Black, 200 + 2 * r, 100 + r / 2, 200 + 2 * r,
    100 + r / 2 + h)
End Sub
```

15) Запустите программу на исполнение.

5. Комплект лабораторных работ по теме «Моделирование и формализация.»

Лабораторная работа №1

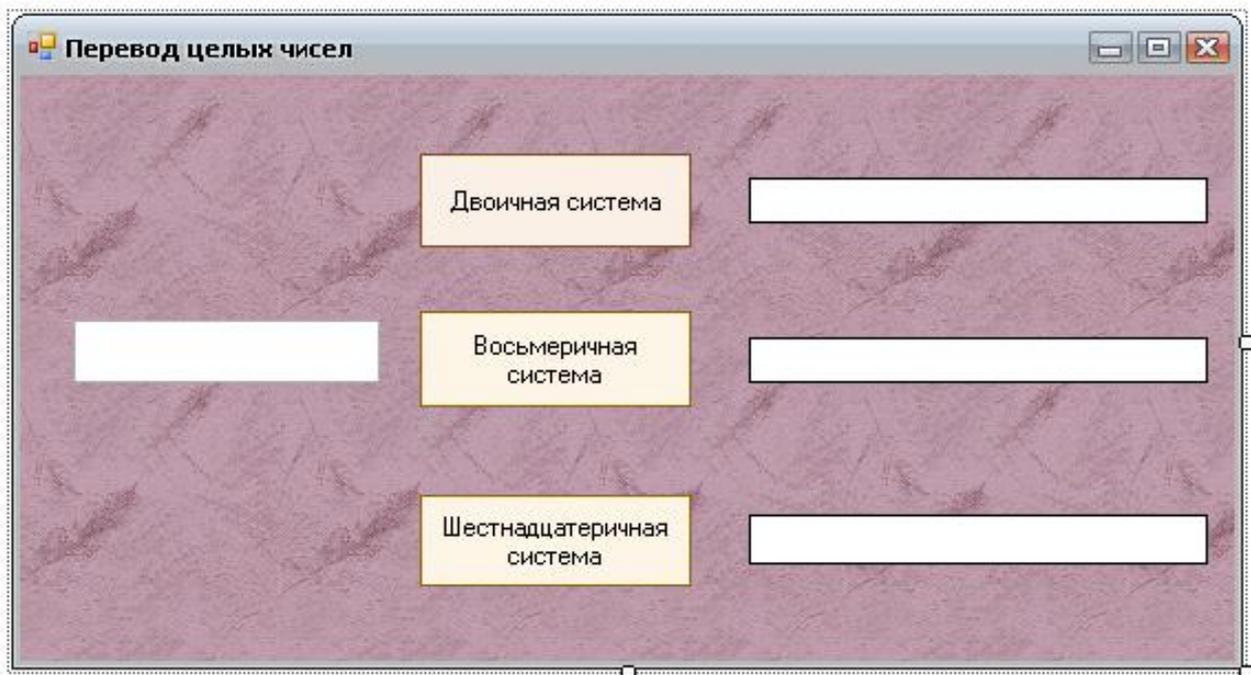
Тема: Проект «Перевод целых чисел»

Цель: научиться создавать проект, реализующий перевод целых чисел в двоичную, восьмеричную и десятичную системы счисления.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Задание:

Составить модель конвектора позволяющий переводить числа из одной системы счисления в другую.



Ход работы:

1. Разместить на форме следующие инструменты:
 - текстовое поле *TextBox1* для ввода целого числа в десятичной системе счисления;
 - три надписи *Label1*, *Label2*, *Label3* для вывода целых чисел в двоичной, восьмеричной и шестнадцатеричной системах;

- три кнопки *Button1*, *Button2*, *Button3*, реализующих перевод в соответствующую систему счисления.
- 2. Настроить общий вид вашей формы с помощью диалогового окна *Свойства*.
- 3. Создать обработчики событий для кнопок, реализующих перевод в соответствующую систему счисления.

- Объявить переменные:

```
Dim N As Integer ' вводимое десятичное число
Dim R As Integer ' остаток от деления исходного делимого числа
Dim Bin As String ' двоичное число в строковой форме
Dim Oct As String ' восьмеричное число в строковой форме
Dim Hex As String ' шестнадцатеричное число в строковой форме
Dim h As String ' разряд шестнадцатеричного числа в строковой форме
```

- Обработчик событий для кнопки1:

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    ' вводим целое десятичное число
    N = Val(TextBox1.Text)
    Label1.Text = ""
    Bin = ""
    ' Проверяем условие: пока исходное целое число или остаток от
    деления больше 0, то проводим вычисления
    Do While N > 0
        ' Вычисляем остаток от деления на два (основание системы
    счисления)
        R = N Mod 2
        N = N \ 2
        ' Записываем полученный остаток от деления слева от двоичного
    числа
        Bin = CStr(R) + Bin
    Loop
    Label1.Text = Bin ' ответ появляется в надписи
End Sub
```

- Обработчик событий для кнопки2:

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    N = Val(TextBox1.Text)
    Label2.Text = ""
    Oct = ""
    Do While N > 0
        R = N Mod 8
        N = N \ 8
        Oct = CStr(R) + Oct
    Loop
    Label2.Text = Oct
End Sub
```

- Обработчик событий для кнопки3:

```
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
```

```
N = Val(TextBox1.Text)
Label3.Text = ""
Hex = ""
Do While N > 0
    R = N Mod 16
    N = N \ 16
    h = R
    If h = 10 Then
        h = "A"
    ElseIf h = 11 Then
        h = "B"
    ElseIf h = 12 Then
        h = "C"
    ElseIf h = 13 Then
        h = "D"
    ElseIf h = 14 Then
        h = "E"
    ElseIf h = 15 Then
        h = "F"
    End If
    Hex = h + Hex
Loop
Label3.Text = Hex
End Sub
End Class
```

4. Запустить программу на исполнение.
5. **Продемонстрировать преподавателю.**
6. Сохранить проект.

Лабораторная работа №2

Тема: Проект «Графический редактор»

Цель: Научиться создавать модель простейшего графического редактора, который позволит рисовать линии, прямоугольники, закрашенные прямоугольники, окружности и закрашенные окружности.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Задание:
Составить модель графического редактора.

Задачи:

- Для рисования элементов использовать меню.
- Обеспечить возможность выбора инструмента рисования: пера или кисти.
- Обеспечить выбор цвета из полной палитры цветов.
- Обеспечить возможность открытия и сохранения графических файлов.

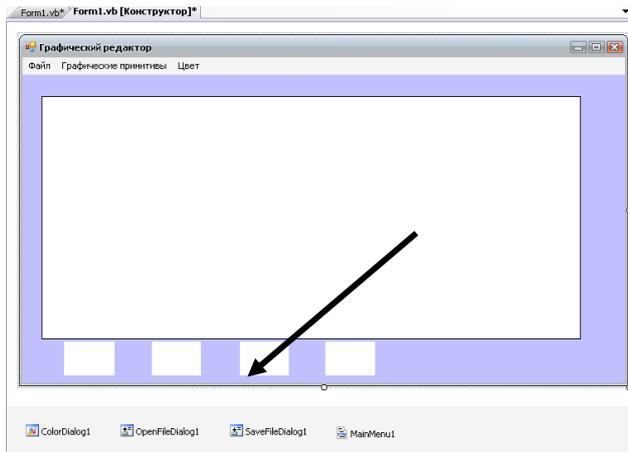
Ход работы.

1. Разместить на форме следующие инструменты:
 - Графическое поле *PictureBox1*, которое будет использоваться как область рисования;
 - Четыре надписи *Label* для вывода имен координат;
 - Меню *MainMenu1* для создания меню проекта.

Файл	Графические примитивы	Цвет
Открыть	Линия	Перо
Сохранить	Прямоугольник	Кисть
	Закрашенный прямоугольник	
	Окружность	
	Закрашенная окружность	
	Очистить	

- Диалог *ColorDialog1*, который позволяет выбрать цвет с использованием диалогового окна Цвет;
- Диалог *OpenFileDialog1*, который позволяет открыть файл;

- Диалог *SaveFileDialog1*, который позволяет сохранить файл.



2. Настроить внешний вид вашей формы с помощью окна *Свойства*.

Обратите внимание, что элементы *MainMenu1*, *ColorDialog1*, *OpenFileDialog1*, *SaveFileDialog1* видны только в процессе создания проекта и отображаются ниже вашей формы. На форме они не видны!

3. Написать обработчик событий для каждого инструмента.

- 1) Объявить переменные:

Dim X1, X2, Y1, Y2 As Integer

Dim Pen1 As New Pen(Color.FromArgb(255, 0, 255))

Dim Brush1 As New SolidBrush(Color.FromArgb(255, 0, 255))

Dim Image1 As New Bitmap(500, 300)

Dim Graph1 As Graphics = Graphics.FromImage(Image1)

- 2) Создать программный код для инструмента *PictureBox1*, который обеспечивает запоминание и вывод координат щелчков правой и левой кнопок мыши. Левая кнопка дает координаты *X1, Y1*; а правая – координаты *X2, Y2*.

Private Sub PictureBox1_Click(ByVal sender As Object, ByVal e As

System.Windows.Forms.MouseEventArgs) Handles PictureBox1.MouseDown

PictureBox1.Image = Image1

Select Case e.Button

Case MouseButton.Left

X1 = e.X

Y1 = e.Y

Label1.Text = X1

Label2.Text = Y1

Case MouseButton.Right

X2 = e.X

Y2 = e.Y

Label3.Text = X2

Label4.Text = Y2

End Select

End Sub

3) Создадим обработчики событий для пунктов меню:

- Файл-Открыть

Private Sub MenuItem2_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem2.Click

OpenFileDialog1.ShowDialog()

End Sub

- Файл-Сохранить

Private Sub MenuItem3_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem3.Click

SaveFileDialog1.ShowDialog()

End Sub

- Линия

Private Sub MenuItem5_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem5.Click

Graph1 = Graphics.FromImage(Image1)

PictureBox1.Image = Image1

Graph1.DrawLine(Pen1, X1, Y1, X2, Y2)

End Sub

- Прямоугольник

Private Sub MenuItem6_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem6.Click

Graph1 = Graphics.FromImage(Image1)

PictureBox1.Image = Image1

Graph1.DrawRectangle(Pen1, X1, Y1, X2, Y2)

End Sub

- Закрашенный прямоугольник

Private Sub MenuItem7_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem7.Click

Graph1 = Graphics.FromImage(Image1)

PictureBox1.Image = Image1

Graph1.FillRectangle(Brush1, X1, Y1, X2, Y2)

End Sub

- Эллипс

Private Sub MenuItem8_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem8.Click

Graph1 = Graphics.FromImage(Image1)

PictureBox1.Image = Image1

Graph1.DrawEllipse(Pen1, X1, Y1, X2, Y2)

End Sub

- Закрашенный эллипс

Private Sub MenuItem9_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem9.Click

Graph1.FillEllipse(Brush1, X1, Y1, X2, Y2)

End Sub

- ОЧИСТИТЬ

Private Sub MenuItem10_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem10.Click

Graph1 = Graphics.FromImage(Image1)

Graph1.Clear(Color.White)

End Sub

- Перо

Private Sub MenuItem12_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem12.Click

ColorDialog1.ShowDialog()

Pen1.Color = ColorDialog1.Color

End Sub

- Кисть

Private Sub MenuItem13_Click(**ByVal** sender **As** System.Object, **ByVal** e **As** System.EventArgs) **Handles** MenuItem13.Click

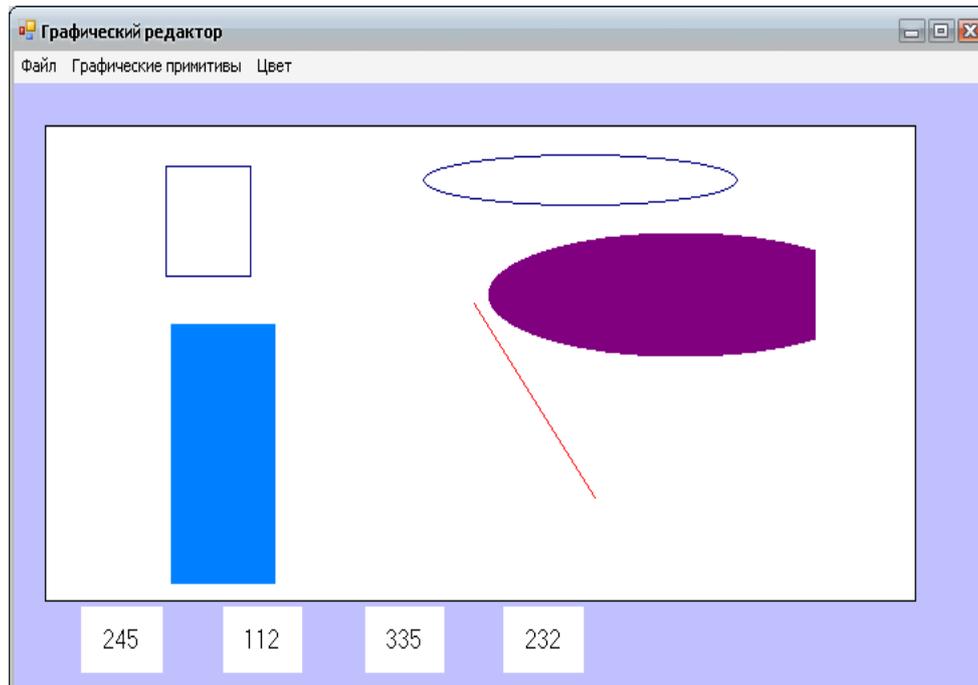
ColorDialog1.ShowDialog()

Brush1.Color = ColorDialog1.Color

End Sub

4. Запустить проект на исполнение.

Щелчками левой и правой кнопок мыши по полю задать координаты двух точек. Из списка выбрать цвет пера и кисти. С использованием меню графические примитивы нарисовать фигуры.



5. Продемонстрировать вашу работу преподавателю.

6. Сохраните проект.

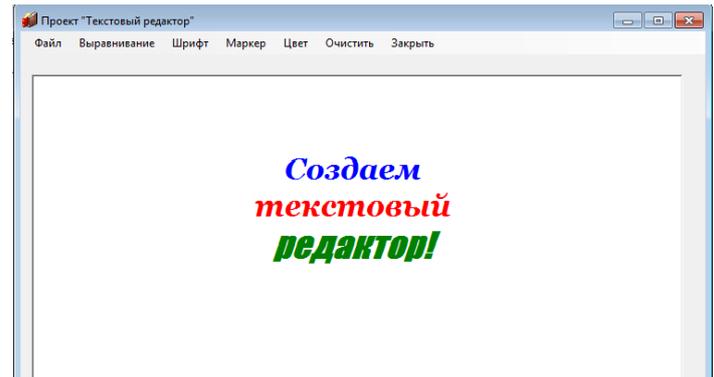
Лабораторная работа №3

Тема: Проект «Текстовый редактор»

Цель: Научиться *создавать* на основе поля *RichTextBox* текстовый редактор

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Задание:
Создать модель текстового редактора



Ход работы:

1. Ввод и редактирование текста производится в окне

Большое прямоугольное поле на форме – это и есть *RichTextBox*. Вы вводите в него вручную любой текст и форматируете его, то есть изменяете его внешний вид. Чтобы отформатировать фрагмент текста, вы его, как принято в текстовых редакторах, сначала выделяете, а затем нажимаете одну из шести кнопок в окне редактора, а именно:

Создайте проект. Поместите на форму нужные кнопки, метки, счетчик *NumericUpDown*. Поместите на форму элемент управления *RichTextBox* и для краткости переименуйте его в *RTB*. Поместите в проект диалоговые окна открытия, сохранения файла, настройки шрифта.

Вот программа:

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    RichTextBox1.HideSelection = False    "Чтобы выделение не снималось с
фрагмента при уходе фокуса с поля"
End Sub
```

```
Private Sub ОткрытьToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ОткрытьToolStripMenuItem.Click
    OpenFileDialog1.Filter = "RTF-файлы|*.RTF|Текстовые файлы|.txt"
    If OpenFileDialog1.ShowDialog() = DialogResult.Cancel Then Exit Sub
    Dim Файл As String = OpenFileDialog1.FileName
    RichTextBox1.LoadFile(Файл)
End Sub
```

```
Private Sub СохранитьToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles СохранитьToolStripMenuItem.Click
    SaveFileDialog1.Filter = "RTF-файлы|*.RTF|Текстовые файлы|.txt"
    If SaveFileDialog1.ShowDialog() = DialogResult.Cancel Then Exit Sub
```

```

Dim Файл As String = SaveFileDialog1.FileName
RichTextBox1.SaveFile(Файл)
End Sub

```

```

Private Sub ШрифтToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ШрифтToolStripMenuItem.Click
    FontDialog1.ShowColor = True 'Предоставить возможность выбирать цвет
шрифта
    FontDialog1.ShowDialog() 'Показать диалоговое окно настройки шрифта
    'Присвоить выделенному фрагменту шрифт, выбранный в диалоговом окне:
    RichTextBox1.SelectionFont = FontDialog1.Font
    'Присвоить выделенному фрагменту цвет шрифта, выбранный в диалоговом
окне:
    RichTextBox1.SelectionColor = FontDialog1.Color
End Sub

```

```

Private Sub ПоЛевомуКрайуToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ПоЛевомуКрайуToolStripMenuItem.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Left
End Sub

```

```

Private Sub ПоЦентруToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ПоЦентруToolStripMenuItem.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Center
End Sub

```

```

Private Sub ПоПравомуКрайуToolStripMenuItem_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ПоПравомуКрайуToolStripMenuItem.Click
    RichTextBox1.SelectionAlignment = HorizontalAlignment.Right
End Sub

```

```

Private Sub ПоставитьToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles ПоставитьToolStripMenuItem.Click
    RichTextBox1.SelectionBullet = True
End Sub

```

```

Private Sub СнятьToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles СнятьToolStripMenuItem.Click
    RichTextBox1.SelectionBullet = False
End Sub

```

```

Private Sub ЗакрытьToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ЗакрытьToolStripMenuItem.Click
    Me.Close()
End Sub

```

2. Запустить программу на исполнение.
3. **Продемонстрировать преподавателю.**
4. Сохранить проект.

Лабораторная работа №4

Тема: Проект «Движение тела»

Цель: научиться создавать компьютерную модель на основе формальной модели, описывающей движение тела, брошенного под углом к горизонту.

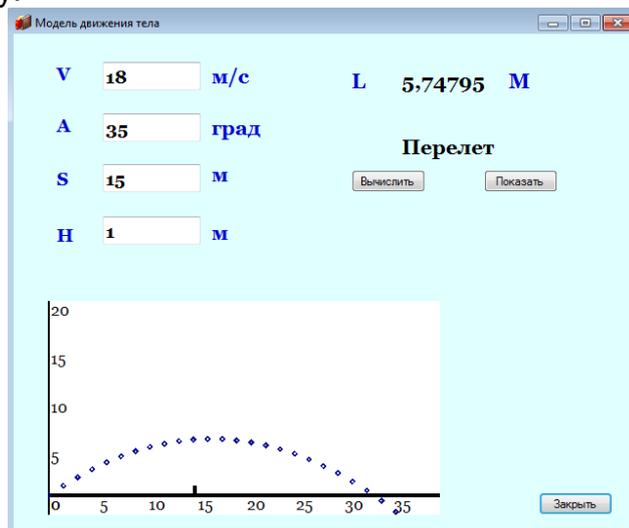
Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.

Задание:

Создать модель движения тела брошенного под углом к горизонту.

Задачи:

1. Создать графический интерфейс проекта «Бросание мячика в стенку».
2. Создать программный код обработчика события, определяющий попадание мячика в стенку.



Ход работы.

1. Разместить на форме:
 - Четыре элемента *TextBox* для ввода значений начальной скорости, угла бросания мячика, расстояния до стенки, высоты стенки.
 - Двенадцать надписей *Label* для вывода имен переменных, единиц измерения, высоты мячика на заданном расстоянии, текстового сообщения о результатах броска.
 - Две кнопки *Button* для запуска процедуры вычисления результатов и для демонстрации движения мячика.
2. Настроить внешний вид формы по- своему усмотрению.
3. Создать программный код, который определяет попадание мячика в стенку.
 - G – ускорение свободного падения
 - Pi – число Пи
 - V0 – начальная скорость
 - A – угол бросания
 - S – расстояние до стенки

```

H – высота стенки
L – высота мячика
Const G As Single = 9.18
Const Pi As Single = 3.14
Dim V0, A, S, L, H As Single
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
' Ввод начальных значений с использованием функции преобразования
строки в вещественное число
V0 = Val(TextBox1.Text)
A = Val(TextBox2.Text)
S = Val(TextBox3.Text)
H = Val(TextBox4.Text)
' Попадание в стенку
L = S * Math.Tan(A * Pi / 180) - (G * S ^ 2) / (2 * V0 ^ 2 * Math.Cos(A * Pi /
180) ^ 2)
Label10.Text = L
'Проверка условия с использованием оператора Select Case
Select Case L
Case Is < 0
Label12.Text = "Недолет"
Case Is > H
Label12.Text = "Перелет"
Case Else
Label12.Text = "Попадание"
End Select
End Sub

```

4. Поместить на форму графическое поле. Установить размер 400; 200.
5. Написать обработчик события, осуществляющий преобразование компьютерной системы координат графического поля в математическую. Построение осуществить в цикле со счетчиком с использованием метода рисования точки, в котором координатами точки являются координаты мячика.

```

Dim Graph1 As Graphics
Dim Pen1 As New Pen(Color.Black, 4)
Dim Pen2 As New Pen(Color.DarkBlue, 4)
Dim drawBrush As New SolidBrush(Color.Black)
Dim drawFont As New Font("Georgia", 12)

```

```

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
Dim X, Y, T As Single
Graph1 = Me.PictureBox1.CreateGraphics()
Graph1.Clear(Color.White)
'Печать шкал математической системы координат в компьютерной
системе координат
For X = 0 To 400 Step 50
Graph1.DrawString(X / 10, drawFont, drawBrush, X, 200)
Next X

```

```

For Y = 20 To 220 Step 50
    Graph1.DrawString((Y - 20) / 10, drawFont, drawBrush, 0, 220 - Y)
Next Y
'Преобразование компьютерной системы координат в математическую
Graph1.ScaleTransform(1, -1)
Graph1.TranslateTransform(0, -200)
'Рисование осей математической системы координат
'Ось X
Graph1.DrawLine(Pen1, 0, 0, 400, 0)
'Ось Y
Graph1.DrawLine(Pen1, 0, -20, 0, 200)
'Стенка
Graph1.DrawLine(Pen1, S * 10, 0, S * 10, H * 10)
'Построение траектории движения мячика
For T = 0 To 10 Step 0.1
    Y = V0 * Math.Sin(A * Pi / 180) * T - G * T * T / 2
    X = V0 * Math.Cos(A * Pi / 180) * T
    Graph1.DrawEllipse(Pen2, X * 10, Y * 10, 1, 1)
Next T
End Sub
End Class

```

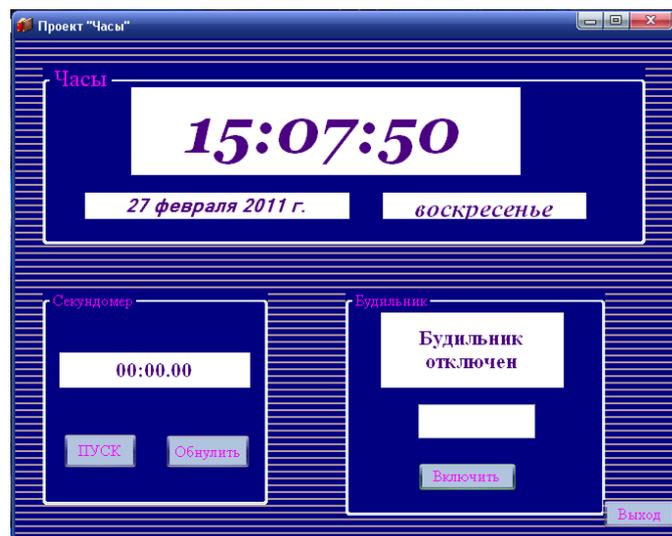
6. Запустить проект и ввести значения. Щелкнуть по кнопкам *Вычислить* и *Показать*. Появятся результаты вычислений и траектория движения мячика.
7. Подобрать значения начальной скорости и угла бросания, обеспечивающие его попадание в стенку.
8. **Продемонстрировать преподавателю.**
9. Сохранить проект.

Лабораторная работа №5

Тема: Проект «Часы»

Цель: научиться создавать компьютерную модель будильника и секундомера.

Программное обеспечение: ОС Windows 7; языка программирования Visual Basic 2008.



Задачи:

На верхнем циферблате должно отображаться текущее время суток (системное время Windows) в часах, минутах и секундах.

Под ним – дата и день недели.

Время будильника пользователь устанавливает, вручную редактируя (изменяя с клавиатуры) содержимое циферблата будильника.

При нажатии на кнопку «Выключить будильник» надпись на кнопке меняется на «Включить будильник», а надпись над циферблатом меняется на «Будильник отключен».

При срабатывании будильника раздается короткий системный звук.

Секундомер измеряет время с точностью до 0.01 сек. Если нажать на «ПУСК», то начнется отсчет времени, а надпись на кнопке сменится на «ПАУЗА». Если опять нажать на эту кнопку, надпись снова изменится на «Пуск», а цифры на секундомере замрут. И т.д.

При нажатии на кнопку «Обнулить» секундомер останавливается и сбрасывается в ноль. На циферблате – 00:00.00.

Основные части проекта:

1. Часы
2. Будильник

3. Секундомер

I. Часы

1. Создайте новый проект. Поместим их на форму три рамки *GroupBox* и дайте имена:

Часы

Будильник

Секундомер

Далее работаем с первой рамкой «Часы»

2. Разместите элемент управления *Label* и дайте ему имя *Циферблат_часов*. Настройки цвета, шрифта сделайте по своему усмотрению.
3. Разместите в рамке элемент *Label* для даты. Дайте ему имя *Циферблат_даты*.
4. Разместите в рамке элемент *Label* для даты. Дайте ему имя *Циферблат_дня_недели*.
5. Поместите на форму таймер и дайте ему имя *Таймер_часов*. Задайте ему интервал = 100. Это значит, что процедура будет выполняться около 10 раз в секунду и не пропустит момента смены системного времени.
6. Настройка даты. В данном проекте необходимо, чтобы дата запускалась автоматически и менялась в полночь. Кроме того, в полночь будет меняться еще и день недели. Поэтому, создадим процедуру *Смена_даты_и_дня_недели*, которая будет запускаться при загрузке проекта.

II. Будильник

1. Поместите в рамку с именем *Будильник* следующие объекты, необходимые для будильника:

- *Label* с именем *Метка_будильника*.
- Текстовое поле *TextBox* с именем *Циферблат_будильника*.
- Кнопку *Button* с именем *Кнопка_включения_выключения_будильника*.
- Кнопку *Button* с именем *Кнопка_выключения_звонка*.

2. Рассмотрим работу будильника. Во время работы в каждый момент времени он может находиться в одном из двух состояний: «Установлен» или «Не установлен». Компьютер должен в любой момент времени знать, в каком именно состоянии находится будильник, чтобы принять решение – звонить или не звонить. Для этого создадим процедуру *Будильник_установлен* и присвоим ей тип *Boolean*.

III. Делаем секундомер

1. Разместите в проекте второй таймер, дав ему имя *Таймер_секундомера*.

2. Поместите в рамку с именем *Секундомер* элементы управления для секундомера и дайте им имена:

Элемент Label - Циферблат_секундомера .

Элемент Button - Кнопка_пуска_паузы_секундомера.

Элемент Button - Кнопка_обнуления_секундомера.

Таймер нужен секундомеру, чтобы вовремя менялись цифры на циферблате, а *собственный* таймер предпочтительно иметь для того, чтобы не зависеть от часов. Поскольку нам нужно отслеживать сотые доли секунды, установим ему интервал меньше 10.

3. **Рассмотрим режимы работы секундомера.** Он может находиться в трех состояниях: считает, стоит в паузе, стоит в нуле. Придумаем переменную и дадим ей имя *Режим_секундомера*. Объявить ее типом *Boolean* явно недостаточно, ведь в типе *Boolean* всего два возможных значения, а нам нужно три. Создаем перечисление:

```
Enum Режим
```

```
    считает
```

```
    пауза
```

```
    в_нуле
```

```
End Enum
```

```
Dim Режим_секундомера As Режим
```

Для каждого режима нам придется организовать свою переменную для отсчета секунд на секундомере:

```
Dim Секунды_на_секундомере As Double
```

```
Dim Секунды_при_запуске_секундомера As Double
```

```
Dim Секунды_на_паузе_секундомера As Double
```

Обратите внимание, что эти переменные, хоть и имеют смысл времени, объявлены, как дробные числовые.

4. **Для отсчета времени на секундомере** мы будем использовать функцию *Timer* класса *DateAndTime*, потому что она выдает секунды с дробной частью. Поскольку она выдает число секунд, прошедших с полуночи, а нам нужно число секунд, прошедших с момента запуска секундомера, да еще с учетом того, что во время паузы на секундомере уже стояли какие-то показания, придется ее переделать.

Засечем в момент пуска секундомера значение функции *Timer* оператором

```
Секунды_при_запуске_секундомера = DateAndTime.Timer
```

Тогда в каждый момент времени после запуска секундомера выражение

```
DateAndTime.Timer - Секунды_при_запуске_секундомера
```

как раз и будет равняться числу секунд, прошедших с момента запуска секундомера. А если нам удастся правильно засечь *Секунды_на_паузе_секундомера* (имеющие смысл секунд, прошедших с момента пуска до паузы), то дело решит оператор

```
Секунды_на_секундомере = DateAndTime.Timer -  
Секунды_при_запуске_секундомера + Секунды_на_паузе_секундомера
```

5. **Внешний вид показаний секундомера.** Переменная *Секунды_на_секундомере* – это дробное число типа *Double*. Но на секундомере эти

самые цифры, должны выглядеть по-другому – 00:01:05.28. Для этого нужно преобразовать число секунд в стандартный формат времени. Оъявим переменную и константу:

```
Dim Время_на_секундомере As Date
Const Полночь As Date = #12:00:00 AM#
```

Создадим следующие операторы:

```
Время_на_секундомере = Полночь.AddSeconds(Секунды_на_секундомере)
Циферблат_секундомера.Text = Format(Время_на_секундомере, "mm:ss.ff")
```

6. Настроим кнопки секундомера. Для этого нам тоже понадобится три оператора

```
Sub Секундомер_обнулить()
Sub Секундомер_остановить()
Sub Секундомер_запустить()
```

Полный текст программы

```
Public Class Form10
    Enum Режим
        считает
        пауза
        в_нуле
    End Enum
    Dim Режим_секундомера As Режим
    Dim Будильник_установлен As Boolean
    Dim Секунды_на_секундомере As Double
    Dim Секунды_при_запуске_секундомера As Double
    Dim Секунды_на_паузе_секундомера As Double
    Dim Время_на_часах As String

    'НАЧАЛЬНАЯ УСТАНОВКА МЕХАНИЗМА
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Смена_даты_и_дня_недели()
        Выключить_будильник()
        Смена_даты_и_дня_недели()
        Секундомер_обнулить()

    End Sub

    'ПРОЦЕДУРЫ РАБОТЫ ЧАСОВ И БУДИЛЬНИКА

    Private Sub Таймер_часов_Tick(ByVal sender As Object, ByVal e As EventArgs) Handles
Таймер_часов.Tick
        Время_на_часах = Format(Now, "HH:mm:ss")
        Циферблат_часов.Text = Время_на_часах
```

```

    If Время_на_часах = "00:00:00" Then Смена_даты_и_дня_недели()
    If Время_на_часах = Циферблат_будильника.Text Then
Microsoft.VisualBasic.Interaction.Beep()

```

```
End Sub
```

```

Sub Смена_даты_и_дня_недели()
    Циферблат_даты.Text = Format(Now, "Long Date")
    Циферблат_дня_недели.Text = Format(Now, "dddd")
End Sub

```

```

Sub Включить_будильник()
    Будильник_установлен = True
    Метка_будильника.Text = "Будильник установлен на"
    Кнопка_включения_Выключения_будильника.Text = "Выключить будильник"
End Sub

```

```

Sub Выключить_будильник()
    Будильник_установлен = False
    Метка_будильника.Text = "Будильник отключен"
    Кнопка_включения_Выключения_будильника.Text = "Включить будильник"
End Sub

```

```

Private Sub Кнопка_включения_выключения_будильника_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Кнопка_включения_Выключения_будильника.Click
    If Будильник_установлен Then Выключить_будильник() Else
Включить_будильник()
End Sub

```

'ПРОЦЕДУРЫ РАБОТЫ СЕКУНДОМЕРА

```

Private Sub Таймер_секундомера_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Таймер_секундомера.Tick
    Dim Время_на_секундомере As Date
    Const Полночь As Date = #12:00:00 AM#
    Секунды_на_секундомере = DateAndTime.Timer -
Секунды_при_запуске_секундомера + Секунды_на_паузе_секундомера
    Время_на_секундомере = Полночь.AddSeconds(Секунды_на_секундомере)
    Циферблат_секундомера.Text = Format(Время_на_секундомере, "mm:ss.ff")
End Sub

```

```

Private Sub Кнопка_пуска_паузы_секундомера_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Кнопка_пуска_паузы_секундомера.Click
    If Режим_секундомера <> Режим.считает Then Секундомер_запустить() Else
Секундомер_остановить()
End Sub

Sub Секундомер_запустить()
    Секунды_при_запуске_секундомера = DateTime.Timer
    Режим_секундомера = Режим.считает
    Таймер_секундомера.Enabled = True
    Кнопка_пуска_паузы_секундомера.Text = "ПАУЗА"
End Sub

Sub Секундомер_остановить()
    Секунды_на_паузе_секундомера = Секунды_на_секундомере
    Режим_секундомера = Режим.пауза
    Таймер_секундомера.Enabled = False
    Кнопка_пуска_паузы_секундомера.Text = "ПУСК"
End Sub

Private Sub Кнопка_обнуления_секундомера_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles Кнопка_обнуления_секундомера.Click
    Секундомер_обнулить()
End Sub

Sub Секундомер_обнулить()
    Секунды_на_паузе_секундомера = 0
    Циферблат_секундомера.Text = "00:00.00"
    Режим_секундомера = Режим.в_нуле
    Таймер_секундомера.Enabled = False
    Кнопка_пуска_паузы_секундомера.Text = "ПУСК"
End Sub
End Class

```

Лабораторная работа №6

Тема: Математическая модель решения квадратных уравнений

Цель: научиться создавать проект, реализующий решение квадратных уравнений типа: $ax^2 + bx + c = 0$.

Программное обеспечение: ОС Windows 7; язык программирования Visual Basic 2008.

Коеффициент a= 1
 Коеффициент b= -5
 Коеффициент c= 6
 Дискриминант = 1
 Корень 1 = 2
 Корень 2 = 3

введите коэффициенты a, b, c завершить работу

Задача: Дано квадратное уравнение общего вида: $ax^2 + bx + c = 0$. Ввести в память компьютера числовые коэффициенты: **a**, **b**, **c**. Найти дискриминант $d=b^2 - 4ac$. Если знак дискриминанта положительный, найти все действительные корни. Если знак дискриминанта отрицательный, сообщить о том, что действительных корней нет.

Задания:

1. Разместите на форме 6 надписей *Label* для обозначения коэффициентов, дискриминанта и корней и 7 надписей для вывода результатов.
2. Разместите на форме две кнопки *Button*. Первая кнопка запускает алгоритм решения, а вторая завершает работу проекта.
3. Написать процедуру, реагирующую на щелчок мыши и реализующую алгоритм решения квадратного уравнения:
 - 1) ввод коэффициентов исходного уравнения a, b, c;
 - 2) расчет дискриминанта $d=b^2 - 4ac$;
 - 3) анализ знака дискриминанта;
 - 4) вычисление корней уравнения и вывод их на форму, если знак дискриминанта $d>0$ (положительный);
 - 5) вывод сообщения: "Решений нет", если знак дискриминанта $d<0$ (отрицательный).

Текст программы:

```

Public Class Form1
    ' Объявляем переменные
    Dim a As Single
    Dim b As Single
    Dim c As Single
    Dim d As Single
    Dim x1 As Single
    Dim x2 As Single
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    ' Ввод начальных значений с помощью окна InputBox
    a = InputBox("введите коэффициент a")
    b = InputBox("введите коэффициент b")
    c = InputBox("введите коэффициент c")
    Label1.Text = (a)
    Label2.Text = (b)
    Label3.Text = (c)
    ' Расчет и вывод значений дискриминанта
    d = b * b - 4 * a * c
    Label8.Text = (d)
    ' Проверка знака дискриминанта
    If d < 0 Then
        Label13.Text = ("Корней нет")
        Label10.Text = ("")
        Label12.Text = ("")
    Else
        If d >= 0 Then
    ' Рачет и вывод значений корней уравнения, если дискриминант больше или равен
    нулю
            x1 = (-b - Math.Sqrt(d)) / (2 * a)
            x2 = (-b + Math.Sqrt(d)) / (2 * a)
            Label10.Text = (x1)
            Label12.Text = (x2)
            Label13.Text = ("")
        End If
    End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        End ' Закрытие программы
    End Sub
End Class

```

Заключение.

Современный этап развития общества характеризуется внедрением информационных технологий во все сферы человеческой деятельности. Новые информационные технологии оказывают существенное влияние и на сферу образования. Происходящие фундаментальные изменения в системе образования вызваны новым пониманием целей, образовательных ценностей, а также необходимостью перехода к непрерывному образованию, разработкой и использованием новых технологий обучения, связанных с оптимальным построением и реализацией учебного процесса с учетом гарантированного достижения дидактических целей.

В данной работе мы попытались решить одну из дидактических задач образовательного учреждения – формирование мышления учащегося, развитие его интеллекта. Поскольку важной составляющей интеллектуального развития человека является алгоритмическое мышление, то обучение решению стандартных алгоритмических задач является первичной целью школьного образования на разных ступенях изучения информатики.

В результате нашей работы были разработаны лабораторные работы, включающие интересные методы построения и использования алгоритмов при решении стандартных задач из раздела «Основы алгоритмизации и программирование», «Моделирование и формализация».

Список литературы.

1. Зиборов В.В. Visual Basic 2010 на примерах. И.: ВHV-СПб , 2010; 336с.
2. Клевцов А.Л., Глушаков С.В. » Visual Basic. NET 2008 . Эффективное руководство. И.:АСТ 2008, 560с.
3. Культин Н. Visual Basic. Освой на примерах. И.:БХВ-Петербург, 2004; 342с.
4. Климов А. Занимательное программирование на Visual Basic .NET. И.:БХВ-Петербург, 2005; 528с.
5. Лукин С.Н. Понятно о Visual Basic.NET. Самоучитель. И.: Диалог-МИФИ, 2005; 736с.
6. <http://vbbook.ru>- Интернет учебник Visual Basic
7. <http://msdn.microsoft.com/ru-ru/library/> Интерактивный учебник по Visual Basic
8. http://progbook.ru/visual_basic/ - Сайт учебной литературы по программированию.