**Лабораторная работа**

**Инспекция кода модулей проекта**

**Цель занятия**: закрепление практических навыков работы с системой Visual Studio 2019; научиться проводить инспекцию кода модулей проекта.

**Оборудование, технические и программные средства:** персональный компьютер, среда программирования Visual Studio 2019.

**Продолжительность занятия:**2 часа.

**Задание:**

1. Изучить теоретическую часть.
2. Выполнить задание, следуя указаниям.
3. Ответить на контрольные вопросы (в устной форме).
4. Предъявить преподавателю результаты работы программы и исходные коды.
5. Оформить отчет в соответствии с ходом работы (тема, цель, условие задачи, программный код, результаты тестирования программы, выводы).

**Теоретические сведения:**

Не во всех случаях возможна разработка автоматических или хотя бы четко формализованных ручных тестов для проверки функциональности программной системы. В некоторых случаях выполнение программного кода, подвергаемого тестированию, невозможно в условиях, создаваемых тестовым окружением (например, во встроенных системах, если программный код предназначен для обработки исключительных ситуаций, создаваемых только при установке системы на реальное оборудование). В других случаях верифицируется не программный код, а проектная документация на систему, которую нельзя "выполнить" или создать для нее отдельные тестовые примеры. И в тех и в других случаях обычно прибегают к методу экспертных исследований программного кода или документации на *корректность* или *непротиворечивость*.

Такие экспертные исследования обычно называют инспекциями или просмотрами. Существует два типа инспекций – неформальные и формальные.

Формальная инспекция является четко управляемым процессом, структура которого обычно четко определяется соответствующим стандартом проекта. Таким образом, все формальные инспекции имеют одинаковую структуру и одинаковые выходные документы, которые затем используются при разработке.

Факт начала формальной инспекции четко фиксируется в общей базе данных проекта. Также фиксируются документы, подвергаемые инспекции, списки замечаний, отслеживаются внесенные по замечаниям изменения. Этим формальная инспекция похожа на автоматизированное тестирование – списки замечаний имеют много общего с отчетами о выполнении тестовых примеров.

В ходе формальной инспекции группой специалистов осуществляется независимая проверка соответствия инспектируемых документов исходным документам. Независимость проверки обеспечивается тем, что она осуществляется инспекторами, не участвовавшими в разработке инспектируемого документа. Входами процесса формальной инспекции являются инспектируемые документы и исходные документы, а выходами – материалы инспекции, включающие *список* обнаруженных несоответствий и решение об изменении статуса инспектируемых документов. рис. 1 иллюстрирует *место* формальной инспекции в процессе разработки программных систем.

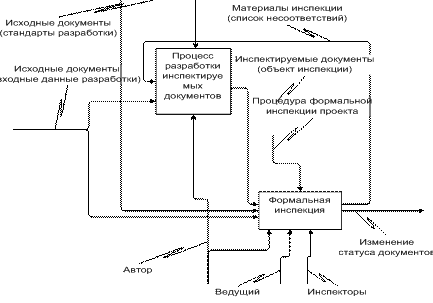


Рис.1. Место формальной инспекции в процессе разработки программных систем

**Выполнение работы:**

**Задание**. Следуя указаниям, создайте программу, которая запрашивает у нового сотрудника имя, фамилию и дату рождения. Вы будете хранить эту информацию в свойствах нового класса с именем Person, и создадите метод класса, который будет вычислять текущий возраст нового сотрудника. Этот проект научит вас создавать собственные классы, экземпляры классов (объекты) а также как использовать эти классы в процедурах событий вашей программы.

Добавление в ваш проект нового класса

Класс, определенный пользователем, позволяет определить в программе ваши собственные объекты, которые имеют свойства, методы и события, точно так же, как объекты, создаваемые на формах Windows с помощью элементов управления из Области элементов. Чтобы добавить в ваш проект новый класс, щелкните в меню Проект (Project) на команде Добавить класс (AddClass), а затем определите этот класс с помощью кода программы и нескольких новых ключевых слов VisualBasic.

**Создание проекта PersonClass**

1. Запустите *MicrosoftVisualStudio2010,* затем создайте в своей папке новый проект с именем **MyPersonClass**.
2. Используйте элемент управления *Label* и добавьте в верхней части формы Form1 длинную метку.
3. Используйте элемент управления *TextBox* и нарисуйте под меткой два широких текстового поля.
4. Используйте элемент управления *DateTimePicker* и нарисуйте под текстовыми полями объект выбора даты и времени.
5. Используйте элемент управления *Button* и нарисуйте под объектом выбора даты и времени кнопку.
6. Установите для объектов формы следующие свойства:

|  |  |  |
| --- | --- | --- |
| **Объект** | **Свойство** | **Установка** |
| Label1 | Text | Введите имя, фамилию и дату рождения сотрудника. |
| TextBox1 | Text | Имя |
| TextBox2 | Text | Фамилия |
| Button1 | Text | Отобразить запись |
| Form1 | Text | Класс Person |

1. Ваша форма должна выглядеть примерно так.

Это базовый интерфейс пользователя для формы, которая определяет запись нового сотрудника фирмы. (Эта форма не подключена к базе данных, так что храниться может только одна запись.) Теперь вы должны добавить в проект класс для хранения информации из этой записи.

1. Щелкните на команде *Добавить класс(AddClass)* в меню *Проект (Project)*. VisualStudio откроет диалоговое окно *Добавление нового элемента (AddNewItem)*, показанное ниже.

Диалоговое окно Добавление нового элемента дает возможность задать имя вашего класса. Когда вы присвоите имя, обратите внимание, что вы можете сохранить в новом модуле класса несколько классов и указать имя, которое будет для них общим.

1. Введите в текстовом поле *Имя(Name)* имя *Person.vb*, а затем щелкните *Добавить*. *VisualStudio* откроет в Редакторе кода пустой модуль класса и добавит имя файла *Person.vb* в ваш проект в *Обозревателе решений*, как показано на рисунке.

**Объявление переменных класса**

* Под оператором программы PublicClassPerson введите следующие объявления переменных:

Здесь вы объявляете две переменные, которые будут использованы исключительно в модуле класса для хранения значений двух строковых свойств. Переменные объявлены с помощью ключевого слова Private, так как по соглашению VisualBasic программисты должны держать внутренние переменные класса закрытыми - другими словами, недоступными для просмотра извне самого модуля класса.

**Создание свойств**

1. Под объявлением переменных введите следующий оператор программы и нажмите клавишу (Enter):

Этот оператор создает свойство вашего класса с именем FirstName, которое имеет тип String. Когда вы нажмете (Enter), VisualStudio немедленно создаст структуру кода для остальных элементов объявления свойства. Требуемыми элементами являются: блок Get, который определяет, что программисты увидят, когда будут проверять свойство FirstName, блок Set, который определяет, что произойдет, когда свойство FirstName будет установлено или изменено, и оператор EndProperty, который отмечает конец процедуры свойства.

1. Заполните структуру процедуры свойства так, чтобы она выглядела, как показано ниже.

Ключевое слово Return указывает, что при обращении к свойству FirstName будет возвращена строковая переменная Name1. При установке значения свойства блок Set присваивает переменной Name1 строковое значение. Обратите особое внимание на переменную Value, используемую в процедурах свойств для обозначения значения, которое присваивается свойству класса при его установке. Хотя этот синтаксис может выглядеть странно, просто поверьте мне - именно так создаются свойства в элементах управления, хотя более сложные свойства будут иметь здесь дополнительную программную логику, которая будет проверять значения и производить вычисления.

1. Под оператором EndProperty введите для свойства LastName вашего класса вторую процедуру свойства. Она должна выглядеть так, как показано ниже.

Эта процедура свойства аналогична первой, за исключением того, что она использует вторую строковую переменную (Name2), которую вы объявили в верхней части кода класса. Вы закончили определять два свойства вашего класса. Теперь перейдем к методу с именем Age, который будет определять текущий возраст нового сотрудника на основе даты рождения.

**Создание метода**

* Под процедурой свойства LastName введите следующее определение функции:

Чтобы создать метод класса, который выполняет некое действие, добавьте в ваш класс процедуру Sub. Хотя многие методы не требуют для выполнения своей работы аргументов, метод Age, определенный мной, требует для своих вычислений аргумент Birthday типа Date. Это метод использует для вычитания даты рождения нового сотрудника из текущей системной даты метод Subtract, и возвращает значение, выраженное в днях, деленных на 365.25 - примерную длину одного года в днях. Функция Int преобразует это значение в целое, и это число с помощью оператора Return возвращается в вызывающую процедуру - как и в случае с обычной функцией.

Определение класса закончено! Вернитесь к форме *Form1* и используйте новый класс в процедуре события.

**Совет**. Хотя в данном примере это и не делалось, в реальном проекте полезно добавить в модуль класса логику для проверки типов данных. Это делается для того, чтобы неправильное использование свойств или методов, не приводило к возникновению ошибок времени исполнения, из-за которых выполнение программы может прерваться.

**Создание объекта с помощью нового класса**

1. Щелкните в Обозревателе решений на значке Form1.vb, а затем на кнопкеОткрыть в конструкторе. Появится интерфейс пользователя Form1.
2. Чтобы открыть в Редакторе кода процедуру события Button1\_Click, сделайте двойной щелчок мышью на кнопке **Отобразить запись**.
3. Введите следующие операторы программы:

Эта процедура сохраняет в объекте с именем Employee, который имеет тип Person, значения, введенные пользователем. Ключевое слово New указывает, что вы хотите немедленно создать новый экземпляр объекта Employee. Теперь нужно объявить переменную с помощью класса, созданного вами самими! Затем процедура объявляет переменную с именем DOB типа Date. Она будет хранить дату, введенную пользователем, и устанавливает свойства FirstName и LastName объекта Employeeравными имени и фамилии, введенным в два объекта текстовых полей формы. Значение, возвращаемое объектом выбора даты и времени, сохраняется в переменной DOB, а последний оператор программы отображает окно сообщения, содержащее свойства FirstName и LastName, а также возраст нового сотрудника, определенный методом Age, который при передаче в него переменной DOB возвращает целое значение. Как только вы определили класс в модуле класса, его легко можно использовать в процедуре события.

1. Чтобы запустить программу, щелкните на кнопке Начать отладку (F5).В среде разработки появится интерфейс пользователя, готовый к приему ваших данных.
2. Введите в текстовое поле FirstName ваше имя, а в текстовое поле LastName - фамилию.
3. Щелкните на раскрывающемся списке объекта выбора даты и времени, и прокрутите его до вашей даты рождения.

**Совет**. Вы можете быстро прокрутить список, щелкнув в открытом диалоговом окне объекта выбора даты на поле года. Появятся небольшие стрелки прокрутки, и вы сможете переходить сразу на год вперед или назад. Также можно быстро перейти на нужный вам месяц, щелкнув на поле месяца, а затем на месяце в появившемся меню.

Ваша форма будет выглядеть примерно так.

1. Щелкните на кнопке **Отобразить запись**. Ваша программа сохраняет значения имени и фамилии в свойствах и использует метод Age для вычисления текущего возраста нового сотрудника. Появится диалоговое окно с результатом.
2. Чтобы закрыть это окно сообщения, щелкните на **OK**, а затем поэкспериментируйте с несколькими различными значениями дат, щелкая на**Отобразить запись** каждый раз, когда вы меняете значение поля даты рождения.

**Требования к отчету:** Текст должен быть написан шрифтом Times New Roman, 12. Интервал между строками и абзацами – 1,5. Отступ слева 1,5. Ориентация текста – по ширине страницы. Скриншоты необходимо подписать. Название практической работы, цель работы, ход работы, вывод, ответы на контрольные вопросы/

**Контрольные вопросы:**

1. Что такое инспекция кода?
2. Какие виды инспекции кода вы знаете?
3. Определите понятия класс, экземпляр класса, объект.
4. На какие этапы можно разбить процесс создания экземпляров класса.
5. Как создать базовый файл класса, какой код он содержит.