|  |  |  |
| --- | --- | --- |
| **Дата** | **Содержание выполняемой работы** | **Подпись**  **руководителя** |
| 21.04.2020 | **Практическая работа №14**  Тема: Сборки, библиотеки, атрибуты, директивы.  Наименование работы: Сборки. Создание библиотеки. Использование библиотеки. Атрибуты. Пространство имен. Директивы препроцессора.  Цель: Сформировать навыки создания собственных библиотек. Сформировать навыки создания программ с применением собственных библиотек.  Задание 1.  Изучил предложенную теорию. Кратко законспектировал.  Сборки, библиотеки, атрибуты, директивы.  Сборка – совокупность взаимосвязанных типов.  Промежуточный язык не содержит инструкций, зависящих от операционной системы и типа компьютера, что обеспечивает две основные возможности:   * выполнение приложения на любом типе компьютера, для которого существует среда выполнения CLR; * повторное использование кода, написанного на любом .NET-совместимом языке.   Метаданные типов — это сведения о типах, используемых в сборке. Компилятор создает метаданные автоматически. В них содержится информация о каждом типе, имеющемся в программе, и о каждом его элементе. Например, для каждого класса описываются все его поля, методы, свойства, события, базовые классы и интерфейсы.  Манифест — это набор метаданных о самой сборке, включая информацию обо всех файлах, входящих в состав сборки, версии сборки, а также сведения обо всех внешних сборках, на которые она ссылается. Манифест создается компилятором автоматически, программист может дополнять его собственными атрибутами.  Создание собственной библиотеки.  Для того, чтобы создать собственную библиотеку необходимо выбрать файл проекта - шаблон ClassLibrary. Библиотека хранится в виде файла с расширением \*.DLL. Для использования такой библиотеки необходимо создать собственный проект и включить в него с помощью команды Project->Add Reference файл собственной библиотеки с расширением dll.  Использование собственной библиотеки.  Любая библиотека — это сервер, предоставляющий свои ресурсы клиентам. Для того, чтобы компилятор мог ее обнаружить, необходимо после создания проекта подключить ссылку на библиотеку с помощью команды Project ► Add Reference. Для поиска каталога, содержащего библиотеку, следует использовать кнопку Browse. После подключения библиотеки можно пользоваться ее открытыми элементами таким же образом, как если бы они были описаны в том же модуле.  Рефлексия.  Рефлексия – это получение информации о всех типах в программе во время ее выполнения. Например, можно получить список всех классов, интерфейсов и т.д. в сборке используя класс System.Type и типы пространства имѐн System.Reflection.  Атрибуты.  Атрибуты – это дополнительные сведения об элементах программы, включающие в себя такие метаданные сборки, которые можно извлекать во время выполнения программы. Атрибуты могут быть стандартными и пользовательскими. Все атрибуты описываются в квадратных скобках перед тем членом класса, к которому он относится.  Пространство имён.  Пространство имён – это хранилище типов, определяющее их область видимости. Пространство имѐн служит для:   * логического группирования элементов программы, расположенных в различных файлах; * группирования имѐн, предоставляемых сборкой в пользование другим модулям.   Директивы препроцессора.  В языке C# препроцессор практически отсутствует, однако некоторые директивы перешли в данный язык из языка C++. Препроцессор - это предварительный этап компиляции, который формирует окончательный вариант текста программы. Директива препроцессора – указание компилятору исключить или включить в процесс компиляции отдельные фрагменты кода, который должен выполняться при определѐнных условиях. В основном в языке C# используются директивы условной компиляции.  Задание 2.  Создал программы по образцу. Листинг кода переписал в дневник-отчет. Вставил скриншот выполнения программы.  Задание 2.1  Создание библиотеки для математических вычислений. Листинг кода задания 2.1 переписал в Приложение 13. Результат выполнения программы представлен на рисунке 29.    Рисунок 29 – результат выполнения программы  Задание 2.2  Создание библиотеки для работы с пользовательским классом. Листинг кода задания 2.2 переписал в Приложение 13. Результат выполнения представлен на рисунке 30.    Рисунок 30 – результат выполнения программы  **Контрольные вопросы**   1. Укажите, для чего предназначена сборка.   Сборка – совокупность взаимосвязанных типов.   1. Укажите состав сборки.   Сборка включает в себя:   * Манифест; * Метаданные; * Код на языке IL; * Ресурсы.  1. Укажите, как создать и использовать библиотеку в Visual Studio.NET.   Для того, чтобы создать собственную библиотеку необходимо выбрать файл проекта - шаблон ClassLibrary.  Любая библиотека — это сервер, предоставляющий свои ресурсы клиентам. Для того, чтобы компилятор мог ее обнаружить, необходимо после создания проекта подключить ссылку на библиотеку с помощью команды Project ► Add Reference.   1. Укажите, для чего предназначена рефлексия.   Рефлексия – это получение информации о всех типах в программе во время ее выполнения.   1. Опишите пространство имен.   Пространство имён – это хранилище типов, определяющее их область видимости. Пространство имѐн служит для:   * логического группирования элементов программы, расположенных в различных файлах; * группирования имѐн, предоставляемых сборкой в пользование другим модулям.  1. Перечислите директивы препроцессора.   #define, #undef, #if, #elif, #error, #warning, #region, #endregion, #pragma. |  |