Интегрированные среды разработки программ

Содержание:

Введение

Для создания элементов ПО, а также отдельных приложений мало обладать знаниями основ определённого языка. Современному программисту потребуется установленная на его персональном компьютере среда программирования. Именно с её помощью работа над будущими программами будет комфортной и приобретёт высокие показатели производительности. Интегрированная среда программирования в классическом виде должна иметь в своём арсенале обычный текстовый редактор, средства для автоматизации сборки и отладчик, а также интерпретатор или компилятор, возможно наличие их обоих в комплекте утилиты.

Поскольку анализ и проектирование изменений ПО занимают больше времени, чем их реализация, интегрированная среда разработки предоставляет автоматизированные инструменты визуализации воздействия изменений.

Непрерывная интеграция является ключевым элементом современых сред разработки программного обеспечения. IDE, поддерживающая многочисленные частые обновления и версии ПО, может облегчить взаимодействие между подразделениями разработки и эксплуатации (методология DevOps). IDE помогает уменьшить время выхода на рынок.

Цель исследования-изучить интегрированные среды разработки программ.

Объект исследования – это совокупность программного обеспечения, предназначенного для **разработки** программного обеспечения. Обычно IDE представляет собой совокупность исполняемых файлов и библиотек. Назначение **интегрированных** средств разработки –ускорение процесса **разработки** программного обеспечения, снижение сложности этого процесса, автоматизация тестирования и отладки программ, выявление множества ошибок при написании исходного кода программы.

Предмет исследования: интегрированные среды разработки программ

Актуальность работы:

IDE ускоряет процесс разработки ПО за счет автоматизации рутинных задач, таких как компиляция и сборка. Она также предоставляет мощные средства отладки, которые помогают выявлять и исправлять ошибки в коде более эффективно. IDE позволяет создавать более качественное и надежное ПО благодаря встроенным средствам анализа кода. Она облегчает работу над изменениями в ПО благодаря интегрированным средствам контроля версий. IDE является ключевым элементом методологии DevOps, что позволяет своевременно обнаруживать и исправлять ошибки в ПО и ускорять его доставку конечным пользователям. В целом, использование IDE актуально и эффективно для ускорения и улучшения процесса разработки программного обеспечения.

Задачи исследования:

- Представить определение интегрированной среды разработки.

- Изучить историю развития IDE.

- Представить классификацию IDE.

- Представить обзор IDE.

Глава 1. Интегрированные среды разработки

1.1. Определение интегрированной среды разработки

Под интегрированной средой разработки обычно комплексное средство, включающее всё необходимое программисту для создания программного обеспечения. Существует некоторый «джентльменский набор» компонентов, которые должны присутствовать в интегрированных средах разработки [6].

Во-первых, это компилятор или интерпретатор, во-вторых - редактор исходного кода программ (обязательно хотя бы с поддержкой подсветки синтаксиса того языка программирования, для которого предназначена среда), а в-третьих - отладчик.

Отладчик - это, более существенная часть интегрированной среды разработки, чем компилятор или интерпретатор, нередко именно отладка программы становится самым сложным и дорогостоящим этапом её создания.

Редактор исходного кода - текстовый редактор для создания и редактирования исходного кода программ. Он может быть отдельным приложением, или встроен в интегрированную среду разработки (IDE) [2].

Редакторы исходного кода имеют некоторые возможности, упрощающие и ускоряющие написание и изменение кода, такие как подсветка синтаксиса, авто дополнение, проверка правильности расстановки скобок, контекстная помощь по коду и многие другие. Такие редакторы предоставляют удобный способ для запуска компилятора, интерпретатора, отладчика или других программ необходимых в процессе разработки программного обеспечения. Несмотря на то, что многие текстовые редакторы могут быть использованы для редактирования исходного кода, если они не имеют расширенных возможностей, автоматизирующих или упрощающих ввод и модификацию

кода, то они не могут называться «редакторами исходного кода», а просто являются «текстовыми редакторами, которые также используют для редактирования исходного кода». Подсветка синтаксиса – выделение синтаксических конструкций текста с использованием различных цветов, шрифтов и начертаний. Обычно применяется в текстовых редакторах для облегчения чтения исходного текста, улучшения визуального восприятия. Часто применяется при публикации исходных кодов в Интернете [4].

Интегрированные среды разработки (IDE) могут содержать инструменты для интеграции с системами управления версиями, такими как Git, SVN или Mercurial, а также различные инструменты для упрощения создания графического интерфейса пользователя.

Многие современные среды программирования также включают браузер классов, инспектор объектов и диаграмму иерархии классов – для использования при объектно-ориентированной разработке программного обеспечения. Хотя, и существуют среды разработки, предназначенные для нескольких языков программирования – такие, как, Embarcadero, RADStudio, QtCreator или Visual Studio, обычно среда разработки предназначается для одного определённого языка программирования – как, например, VisualBasic, Dev-C++.

Частный случай интегрированных сред программирования – среды визуальной разработки, которые включают в себя возможность визуального редактирования интерфейса программы [7].

Среда визуальной разработки - среда разработки программного обеспечения, где наиболее распространенные блоки программного кода представлены в виде графических объектов. Применяются в основном для создания app и разработки графического интерфейса пользователя (GUI).

Преимущества:

- быстрота разработки;

- лёгкость освоения;

- стандартизация внешнего вида программ.

Недостатки:

- привязка к конкретной среде разработки связана с проблематичностью перехода на другую среду разработки;

- затруднённое использование нестандартных компонентов;

- наличие недокументированных особенностей компонентов.

Некоторые визуальные среды разработки имеют собственный формат хранения проекта, и при переходе на другую среду может возникнуть непереносимость свойств проекта и некоторых частей проекта, таких как собственные библиотеки используемой среды разработки [9].

1.2. История развития IDE

Первые IDE были созданы для работы через консоль или терминал. Ранние системы не могли поддерживать того, что программы были подготовлены, используя блок-схемы, вводя текст с перфорированных карт (или перфолента, и т.д.) прежде, чем представить их компилятору. Dartmouth BASIC был первым языком, который был создан с IDE (и был также первым, который был разработан для использования в консоли или терминале). Эта IDE (часть DartmouthTimeSharingSystem) была командная (т.е. управлялась при помощи команд), и поэтому очень отличалась от управляемых с помощью меню, графических IDE, распространенных сегодня. Однако это позволило редактировать, управлять файлами, компилировать, отлаживать и выполнять способом, непротиворечивым современным IDE [3].

Одной из первых IDE возможностью подключения плагинов была Softbench. В 1995 Computerwoche прокомментировал, что использование IDE не было хорошо воспринято разработчиками, обосновывая это тем, что они будут ограничивать их в творческом потенциале.

Сразу же после создания Java, уже в 1996 г., появились интегрированные среды разработки программ для Java, и их число все время возрастает. Некоторые из них являются просто интегрированными оболочками над JDK, вызывающими из одного окна текстовый редактор, компилятор и интерпретатор. Эти интегрированные среды требуют предварительной установки JDK. Другие содержат JDK в себе или имеют собственный компилятор, например, JavaWorkshop фирмы SUN Microsystems, JBuilder фирмы Inprise, VisualAgeforJava фирмы IBM и множество других программных продуктов. Их можно устанавливать, не имея под руками JDK. Надо заметить, что перечисленные продукты написаны полностью на Java.

Большинство интегрированных сред являются средствами визуального программирования и позволяют быстро создавать пользовательский интерфейс, т е. относятся к классу средств RAD (RapidApplicationDevelopment).

Выбор какого-либо средства разработки диктуется, во-первых, возможностями вашего компьютера, ведь визуальные среды требуют больших ресурсов, во-вторых, личным вкусом, в-третьих, уже после некоторой практики, достоинствами компилятора, встроенного в программный продукт [1].

К технологии Java подключились и разработчики CASE-средств. Например, популярный во всем мире продукт RationalRose.

1.3. Классификация IDE

Конечно, современные интегрированные среды разработки предлагают программистам гораздо больше возможностей, чем входят в описанный выше необходимый минимум. Например, многие современные IDE являются визуальными - они позволяют создавать интерфейс программы с помощью мышки, точно в таком виде, в каком он предстанет потом пользователю. IDE, не являющиеся визуальными, требуют от программиста писать специальный код, ответственный за создание пользовательского интерфейса программы [5].

Различные типы IDE предоставляют разные функции.

Инструменты интеграции обеспечивают инструменты языка программирования и скриптов

Интеллектуальный редактор кода предоставляет помощников по кодингу, таких как предупреждение ошибок, которые помогут автоматически завершить.

SmartCodeNavigation обеспечивает удобную навигацию кода в больших проектах с большим количеством файлов и папок с под проектами

Компилятор обеспечивает компиляцию там, где нужен данный язык пр-я.

Отладчик обеспечивает отладку и проверяет скомпилированные 2е файлы

Рефакторинг будет выполнять рефакторинг во время разработки кода и предоставлять предложения

Поддержка языков программирования и скриптов сделает среду IDE полно задачной, поддерживая несколько связанных языков программирования и языков скриптов, которые могут находиться в одном проекте.

Интерактивная консоль предоставляет интерактивную оболочку или консоль для выдачи команд, связанных с проектом, и получения результатов в терминальном режиме.

Контроль версий обеспечивает версионность кода и облегчает отслеживание изменений

Плагины предоставляют множество полезных функций в качестве дополнения к существующей IDE

Интегрированные среды созданы для работы с конкретными платформами приложений и устранения барьеров, связанных с жизненным циклом разработки ПО. Они используются в командах разработчиков для создания нового ПО и сервисов. Среды разработки программ предназначены для программирования кода и имеют встроенные функции, которые знают, как работает платформа и как использовать ее функции посредством

автоматической компиляции кода, его отладки или интеллектуального завершения.

С ростом технологий и увеличением требований пользователей функциональность программных сред изменяется, а коллекция инструментов для разработчиков значительно расширяется с 1990-х годов. Существуют четыре категории сред для программирования: ориентированные на язык и структурно-ориентированные. Ориентированные на язык среды разрабатываются вокруг конкретного языка и предоставляют ограниченную поддержку для программирования в целом. Структурно-ориентированные среды включают методы, позволяющие напрямую манипулировать структурами, и не зависят от языка, что привело к появлению генераторов для сред.

Существуют два типа сред разработки.

Первый тип - среды инструментария, которые предоставляют независимые от языка инструменты для программирования, включая контроль версий и управление конфигурацией.

Второй тип - это методы среды, обеспечивающие множество подпрограмм для управления командой и проектом, а также инструменты для конкретных спецификаций и методов проектирования.

Глава 2. Обзор IDE, Редакторы кода IDE:Code:Blocks, Dev-C++, Visual.

Code: :Blocks

Code: :Blocks - это бесплатная C и C++ IDE с открытым исходным кодом, приложение расширяемое, полностью настраивается и может работать на нескольких платформах.

Это программное обеспечение разработано для Windows, Linux, FreeBSD и Mac OS X. Оно создано с использованием плагинов и может быть расширено путём установки или создания дополнительных компонентов.

Особенности Code Blocks:

Простой интерфейс с вкладками открытых файлов;

Совместимость с Linux, Mac и Windows;

Написана на C++;

Не требует интерпретируемых или проприетарных языков програмирован.

Множество встроенных и настраиваемых плагинов;

Поддерживает несколько компиляторов, включая GCC, MSVC ++, и др.

Отладчик с поддержкой контрольных точек;

Текстовый редактор с подсветкой синтаксиса и функцией авто заполнения;

Настраиваемые внешние инструменты;

Простые средства управления задачами, идеально подходящие для совместной работы.

Программа имеет WYSIWYG-редактор для библиотеки wxWidgets для разработки графического интерфейса пользователя (GUI).

Поддержка нескольких компиляторов:MinGW/GCC C/C+,Visual C++,и др.

Настраиваемый и расширяемый пользовательский интерфейс с поддержкой вкладок, с подсветкой синтаксиса, сворачиванием блоков кода, авто-завершением кода, интеллектуальными отступами и браузером классов.

Имеется быстрая система сборки и используются рабочие пространства для объединения нескольких проектов.

Code::Blocks имеет расширенную функциональность для отладки и поддерживает GNU GDB и MS CDB.

Кроме того, у программы есть полная поддержка точек остановки (breakpoints), отображение стека вызовов, дизассемблер, настраиваемый дамп памяти и отображение информации профайлера (GNU Profiler).

IDE совместима с MSVC и Dev-C++, может импортировать проекты и рабочие пространства.

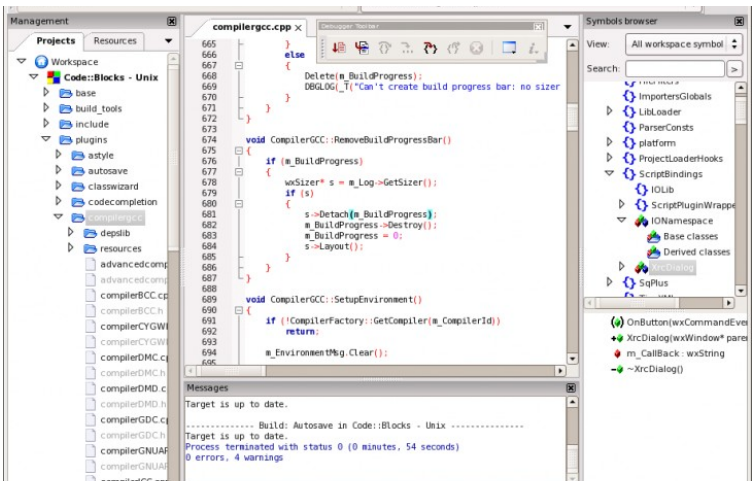


Рис. 1 Окно Code::Blocks

Недостатки: относительно компактная среда разработки Си, поэтому она не подходит для крупных проектов. Это отличный инструмент для новичков, но продвинутые программисты могут быть разочарованы ее ограничениями.

Dev-C++

Dev-C++ - интегрированная среда разработки на C и C++, полнофункциональная C++ IDE.

Возможности DevCpp:

Удобный редактор с подсветкой синтаксиса, нумерацией строк, авто отступами и т.д.

Возможность авто завершения кода для удобства работы и повышения производительности.

Заготовки кода и шаблоны для вставки.

Использование закладок в редакторе для быстрого перемещения по коду.

Экспорт исходных файлов или целого проекта в HTML или RTF для публикации исходных кодов на своём веб сайте.

Встроенный менеджер проектов.

Импорт проектов из MS Visual C++.

Возможность настройки ассоциации файлов по расширению - c, cpp, h.

В навигаторе классов два варианта обзора - просмотр функций, классов и их членов как для всего проекта, так и для текущего редактируемого файла.

Гибкая настройка рабочей среды, редактора и компилятора, большое количество различных опций.

Используется Mingw GCC компилятор, может работать с любым компилятором GCC.

Возможность отладки проекта - встроенный дебаггер GDB.

Возможна работа с CVS (скачивается отдельно).

Существует пор табельная версия программы, не требующая установки.

Мультиязычный пользовательский интерфейс с поддержкой русского и украинского языков.

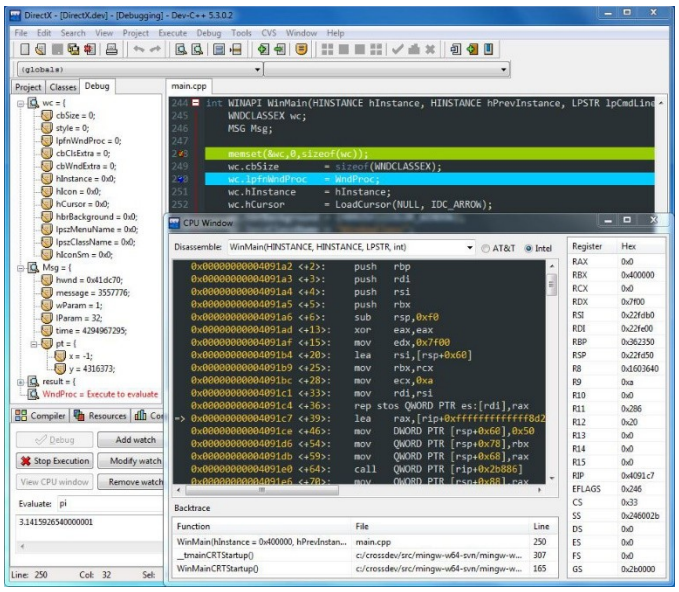


Рис. 2 ОкноDev-C++

VisualStudio

VisualStudio —продукты Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с

поддержкой технологии WindowsForms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, WindowsMobile, Windows CE, .NET Framework, Xbox, , Android, IOS, .NET Поддерживает следующие языки: VisualBasic, C++, C#, F#.

Что она умеет?

IntelliSense. Технология авто дополнения Microsoft. Дописывает название функции при вводе начальных букв. Кроме прямого назначения, IntelliSense используется для доступа к документации и для устранения неоднозначности в именах переменных, функций и методов, используя рефлексию.

PerfomanceAnalizer. Инструмент, отображающий затраты ресурсов при работе приложения/сервиса в виде статистики и графиков.

TestManager. Встроенный менеджер тестов. После создания теста можно с помощью специального окна запускать и настраивать тесты.

Extension/UpdatesManager. Менеджер плагинов, адаптеров, провайдеров. Позволяет легко найти, установить, обновить любое дополнение.

Nuget. Система управления пакетами для платформ разработки Microsoft, в первую очередь библиотек .NET Framework. Управляется .NET Foundation. Удобная установка библиотек в любой .Net проект.

FileManager. Для добавления нового файла в проект существует встроенный менеджер файлов. Удобное создание любых файлов на основе шаблонов. Реализовано большое количество стандартных шаблонов (Пример: класс). Также можно добавлять свои. При установке новой технологии - добавляются соответствующие шаблоны.

Customization. Возможность изменить внешний вид VisualStudio под себя. Изменения цветов, темы, шрифтов, отступов и т.д. Расположение окон в удобном вам виде.

Setting. Настройка всего вышеперечисленного функционала. Настройка быстрых клавиш, уведомлений, быстрый запуск, стартового окна, вкладок, разметки языков и много другого

Благодаря огромному количеству настроек, поддерживаемых технологий, быстродействию и удобству VisualStudio считается одной из лучших сред разработки. Из минусов можно выделить огромный вес пакетов технологий.

3. Знакомство с интегрированной средой разработки Visual Studio

Окно запуска

. Окно запуска помогает быстрее добраться до кода. Здесь есть действия, позволяющие клонировать или извлечь код, открыть существующий проект или решение, создать новый проект или просто открыть папку с файлами кода.

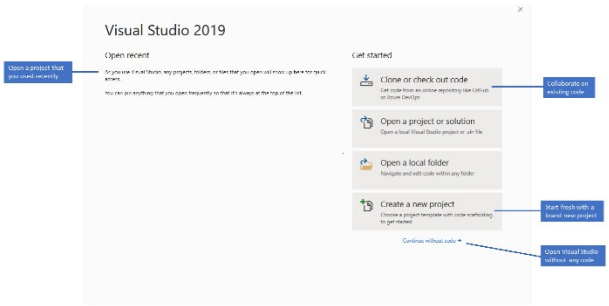


Рис. 3 Окно запуска

Если используется VisualStudio впервые, список последних проектов будет пустым.

Создание проекта

Чтобы продолжить изучение функций VisualStudio, давайте создадим новый проект.

На начальном экране выберем Создать проект.

Открывается диалоговое окно с заголовком Создание проекта. В нем можно выполнить поиск, фильтрацию и выбор шаблона проекта. Здесь также отображается список недавно использованных шаблонов проекта.

Введем в поиск вверху строки консоли, чтобы оставить в списке только те типы проектов, в имени которых есть слово "консоль". Дополнительно уточните результаты, выбрав C# (или другой нужный язык) из средства выбора Язык.

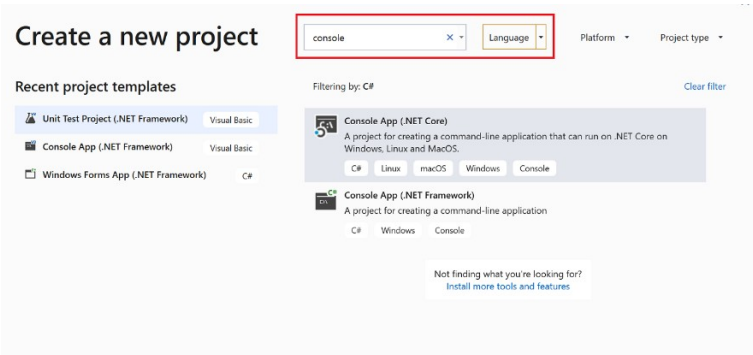


Рис. 4 Новый проект

Если выбрали язык C#, VisualBasic или F#, выберите шаблон Консольное приложение (.NET Core) и щелкните Далее. (Для другого языка просто выберите любой шаблон. Рассматриваемый нами пользовательский интерфейс выглядит одинаково для всех языков программирования.)

На странице Настроить новый проект сохраним имя проекта по умолчанию и щелкнем Создать.

Будет создан проект. В окне редактора откроется файл с именем *Program.cs*.

В редакторе отображается содержимое файлов. Кроме того, здесь можно выполнять основную часть работы с кодом в Visual Studio.

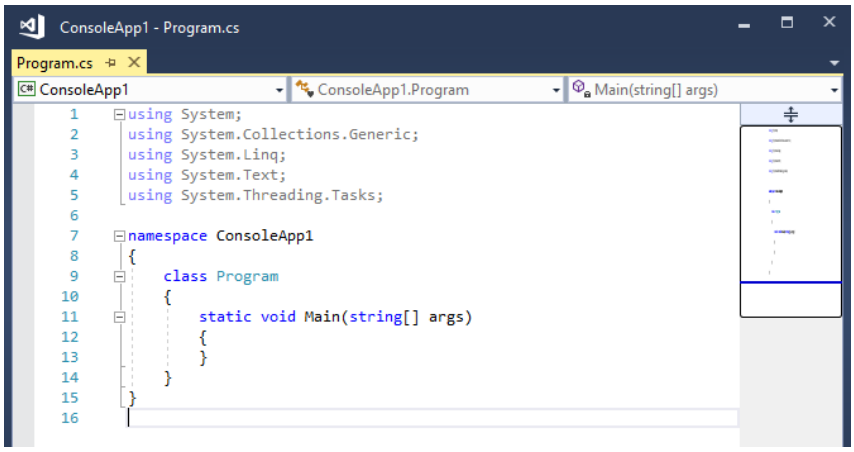


Рис. 5 Окно редактора

Обозреватель решений

В **обозревателе решений**, который обычно находится в правой части VisualStudio, отображается графическое представление иерархии файлов и папок в проекте, решении или папке с кодом. В **обозревателе решений** можно просматривать эту иерархию и переходить к нужным файлам.

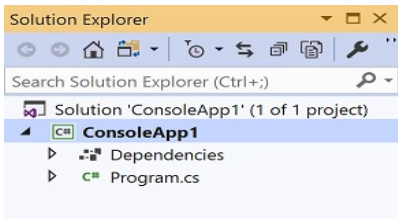


Рис. 6 Окно Обозревателя решений

Меню

В меню в верхней части VisualStudio сгруппированы команды по категориям. Например, в меню Проект содержатся команды, связанные с проектом, над которым вы работаете. В меню Инструменты можно настроить VisualStudio, выбрав Параметры. Также можно включить в установку нужные компоненты, выбрав Получить средства и компоненты.

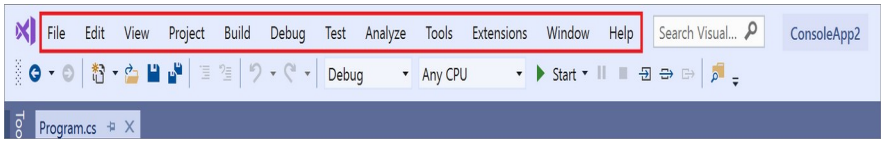


Рис. 7 Меню

Список ошибок

Откроем окно Список ошибок, выбрав в меню Вид пункт Список ошибок.

Список ошибок содержит ошибки, предупреждения и сообщения о текущем состоянии кода. Если в файле или любой другой части проекта будут обнаружены ошибки (например, отсутствуют скобки или точка с запятой), они будут перечислены здесь.

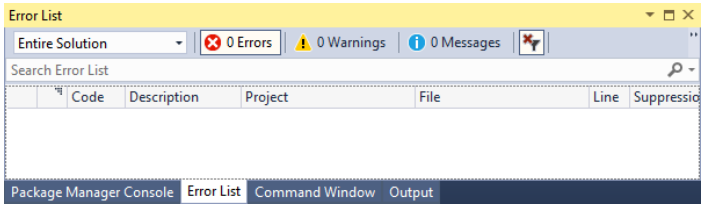


Рис. 8 Список ошибок

Окно вывода

В окне вывода отображаются выходные сообщения от процесса сборки проекта и поставщика системы управления версиями.

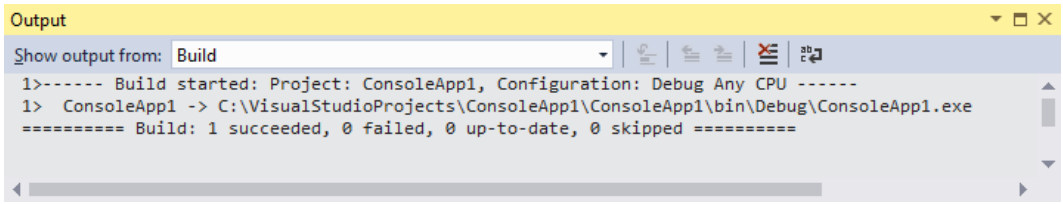


Рис. 9Окно вывода

Поле поиска

Поле поиска — это быстрый и простой способ перейти к любым инструментам в VisualStudio. Сюда можно вводить текст, связанный с тем, что вы планируете делать, чтобы получить список соответствующих возможностей. Предположим, нам нужно детализировать выходные данные о сборке, чтобы отобразить дополнительные сведения о функциях нашей сборки. Вот как это можно сделать:

Чтобы активировать поле поиска нажмем клавиши Ctrl+Q.

Введем детализация в поле поиска. В отображаемом списке результатов выберите ChangeMSBuildverbosity (Изменить уровень детализации MSBuild).

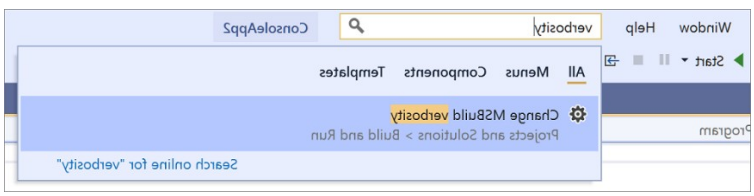


Рис. 10 Поле поиска

На странице Сборка и запуск откроется диалоговое окно Параметры.

В разделе Степень подробности сообщений при сборке проекта MSBuild выберем значение Обычная и нажмите кнопку ОК.

На этот раз в окне Вывод отображаются более подробные сведения из журнала, связанные с процессом сборки. В нашем случае — о том, какие файлы были скопированы в определенное расположение.

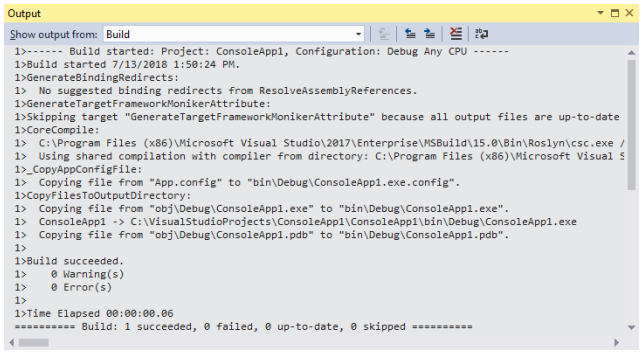


Рис. 11 Подробные сведения из журнала, связанные с процессом сборки.

Практическая работа VisualStudio

Введение

В процессе выполнения всех заданий, вами будет создан полноценный веб-сайт сервисного центра ООО «Матрешка».

Структура веб-страниц представлена на рисунке 1.1.

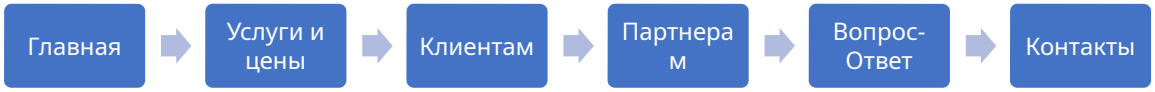


Рисунок 1.1 - Структура итогового веб-сайта

Цель: ознакомиться с Visual Studio Code, а так же создать страницу с дизайном для всего сайта.

Задачи:

–Исследовать……рабочую…область.

– Научиться создавать новую страницу, используя содержимое HTML5, CSS.

–Научиться……сохранять…....документ.

– Научиться изменять название страницы и текстовые заголовки.

– Научиться добавлять изображения на передний план и в качестве фоновых.

Подготовка к созданию проекта изучение редактора VisualStudio Code.

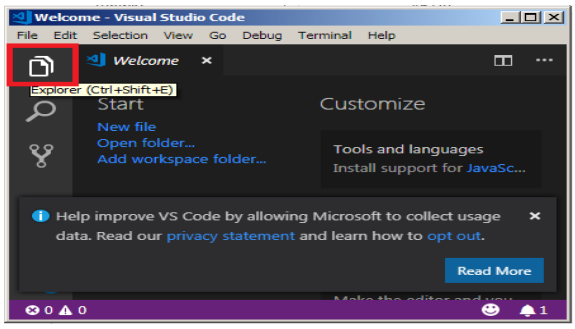
Visual Studio предлагает общедоступные инструменты и гибкость, необходимые для создания и развертывания современных веб-приложений.

–HTML5,CSS3,LESS/SASS,JavaScript.ASP.NET,Node.js,Python,JavaScript Платформа и среда выполнения с открытым кодом.

– Развертывание в Windows, Azure, Масили Linux.

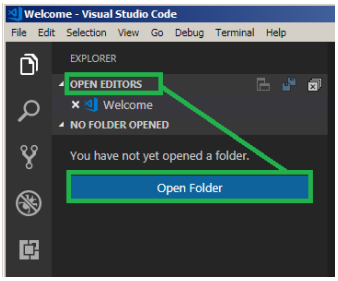
– Бесплатно для небольших групп и разработчиков открытого кода.

– Запустите **Visual Studio Code**. В боковой панели навигации кликните по иконке страниц.(рисунок 1.2).



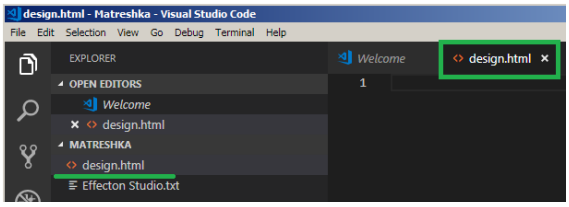
1.2 - Боковая панель

– В появившемся меню, кликните на кнопку **Open editors** и в открывшемся окне выберите созданную заранее папку (напимар Matreshka). Кликните кнопку **Open Folder**.



1.3 - Боковая панель. Выбор папки

– Для создания первой страницы сайта **design.html** наведите мышку на область названия папки(проекта) **Matreshka** и кликните значок  . В появившейся форме введите **design.html.**

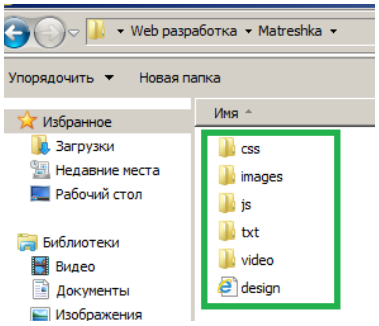


1.4 - Создан файл design.html

Наполнение папки Matreshka

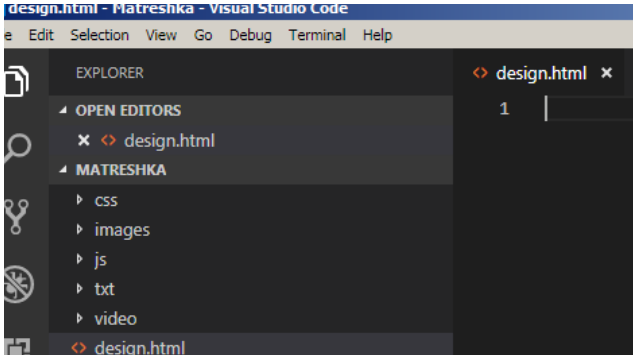
Для создания проекта, нам понадобятся дополнительные файлы: CSS, JS, мультимедиа.

– Для этого скачайте в вашу папку, папки и файлы из папки **work_files.**



1.5 - Папка Matreshka

– Откройте редактор. Папки появились в навигационном меню проекта. Они будут задействованы позже.

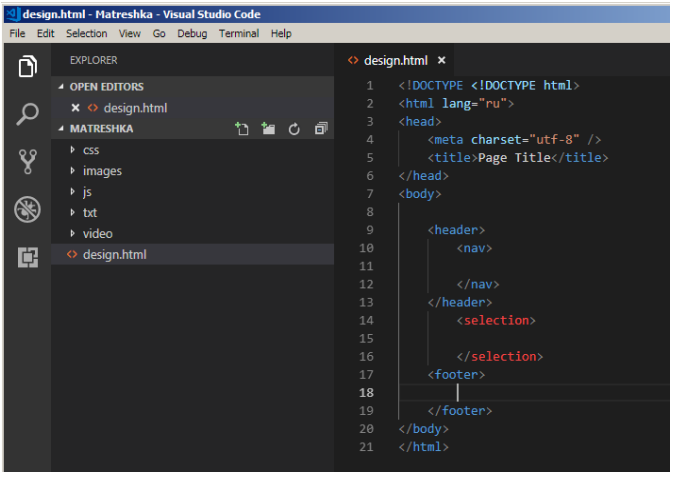


1.6 - Название страницы

Создание дизайна сайта.

Для создания дизайна сайта нами будет подключен файл CSS, в котором указаны свойства для элементов HTML5. Изучить CSS вы можете самостоятельно.

– Заполните файл **design.html** согласно стандартной структуре HTML5 документа, как на рисунке 1.7:



1.7 - Стандартная структура

– Между тегами **title** введите название страницы **ООО «Матрешка»** (рисунок 1.8).

```
<head>
    <meta charset="utf-8" />
    <title>ООО "Матрешка"</title>
</head>
```

1.8 - Название страницы

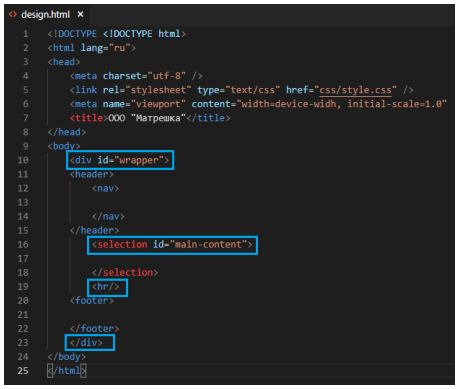
– Следующим шагом будут **meta и link** данные. Между тегами **head** добавьте следующие строки (рисунок 1.9):

```
<head>
    <meta charset="utf-8" />
    <link rel="stylesheet" type="text/css" href="css/style.css" />
    <meta name="viewport" content="width=device-widh, initial-scale=1.0"
    <title>ООО "Матрешка"</title>
</head>
```

1.9 - Добавлены meta и link данные

Данным кодом вы к файлу **design.html** подключили css файл со свойствами для элементов HTML5, а также **meta name="viewport"** для того, чтобы сообщить браузеру, что ваша страница предназначена для правильного поведения области просмотра как на персональных компьютерах, так и на мобильных устройствах.

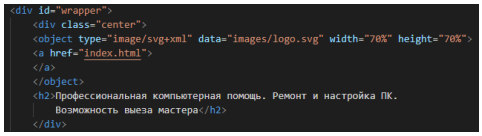
– Для отображения страницы как на примере, нужно прописать id для тэгов и разделительную полосу при помощи тега hr.



1.10 – Добавьте выделенные области кода на свою страницу

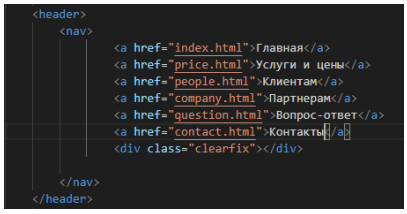
– Добавим логотип на страницу. Логотип у вас используется в формате SVG, если же у пользователя поддержка SVG не предусмотрена браузером выводиться будет альтернативное изображения в формате PNG.

Вся область логотипа и подпись снизу будет обернута **DIV** с именем **center** для того, что бы все объекты внутри располагались по центру. Клик по картинке будет переводить вас на главную страницу сайта (рисунок 1.11).



1.11 - Код добавления логотипа на страницу

– Следующим шагом будет создание навигации на сайте. Стиль навигации уже прописан в CSS файле. Вам осталось добавить ссылки между тегоми **nav**. В конце добавим **div clearfix. Clearfix** - метод отмены действия **float** без изменения структуры HTML-документа (рисунок 1.12).

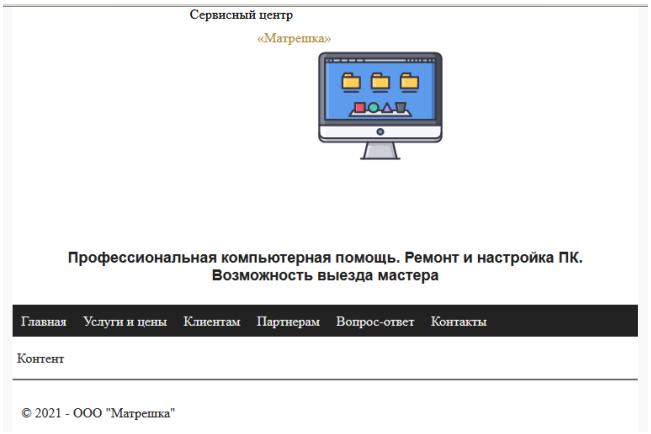


1.12 - Код ссылок для вставки навигации

– Добавьте копирайт в подвал сайта между тегами **footer** (рисунок 1.13).

1.13 – Добавление подвала

Создана первая веб-страница на HTML5 (рисунок 1.16): добавлен текст и цвета, добавлены изображения и подключена CSS. Вы выполнили предварительный просмотр страницы в браузере (рисунок 1.14).



1.14 - Предварительный просмотр веб-страницы

Заключение

Список IDE намного больше представленного выше, выбирая подходящую среду разработки для себя стоит руководствоваться личными потребностями. В первую очередь нужно определиться, какие функции в приоритете, что планируется делать и какие инструменты необходимы в первую очередь. Язык программирования, который используется в работе, тоже играет роль при выборе.

Существует мнение, что только платный продукт способен максимально удовлетворить запросы, решить поставленные задачи и выполнить работу. Другие считают наоборот, что бесплатная интегрированная среда вполне может справится с поставленными задачами и платить нет смысла.

В процессе выполнения курсовой работы были изучены след вопросы:

1. Определение интегрированной среды разработки
2. Историю развития IDE
3. Классификацию IDE

4. IDE Code::Blocks
5. IDE Dev-C++
6. VisualStudio

Команды разработчиков со всего мира трудятся ежедневно, совершенствуя продукт и чем больше они делают шагов для улучшения, тем больше появляется споров какая же среда лучше. Но вывод один — руководствоваться следует индивидуальными предпочтениями. Делая выбор, лучше учитывать больше факторов и только пробой и постепенным перебором средств разработки получится найти оптимальный вариант.

Список литературы

1. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. – **435**с.
2. Анашкина, Н.В. Технологии и методы программирования / Н.В. Анашкина, Н.Н. Петухова, В.Ю. Смольянинов. - М.: Academia, 2017. - **312**с.
3. Гагарина, Л.Г. Технология разработки программного обеспечения / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснадул. - М.: Форум, Инфра-М, **2016**. – **423**с.
4. Гвоздева, В.А. Введение в специальность программиста / В.А. Гвоздева. - М.: Форум, Инфра-М, **2017**. – **265**с.
5. Громов Ю.Ю., Иванова О.Г., Алексеев В.В. и др. Интеллектуальные информационные системы и технологии: учебное пособие – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2017. – 244 с.
6. Емельянова, Н.З. Проектирование информационных систем / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. - М.: Форум, **2018**. – **312**с.

7. Иванова, Г.С. Объектно-ориентированное программирование / Г.С. Иванова, Т.Н. Ничушкина. - М.: МГТУ им. Н. Э. Баумана, 2019. – **576**с.

8. Мещеряков, С.В. Эффективные технологии создания информационных систем / С.В. Мещеряков, В.М. Иванов. - М.: Политехника, **2018**. – **243**с.

9. Рудаков, А.В. Технология разработки программных продуктов. Учебник / А.В. Рудаков. - М.: Академия, 2017. – **318**с.

10. Веб-сайт «NetBeans» [электронный ресурс]. Режим доступа https://netbeans.org/