БГУИР

Основы программной инженерии

ОТЧЕТ

По индивидуальной практической работе №1

Выполнил: Проверил:

студент Шульдова С.Г.

группы № 281073

Одинец А.С

**СОДЕРЖАНИЕ**

[1. ОПИСАНИЕ ПРИМЕНЕНИЯ 3](#_Toc136453537)

[2. ТЕХНИЧЕСКОЕ ОПИСАНИЕ ПРОЕКТА 4](#_Toc136453538)

[3. РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 7](#_Toc136453539)

[3.1 Руководство по установке приложения 7](#_Toc136453540)

[3.2 Руководство по эксплуатации 9](#_Toc136453541)

[Контрольные вопросы 13](#_Toc136453542)

# ОПИСАНИЕ ПРИМЕНЕНИЯ

Решение квадратного уравнения является важной задачей в математике и науке. Квадратные уравнения имеют широкое применение в различных областях, таких как физика, инженерия, экономика и компьютерные науки. Квадратные уравнения имеют вид ,   
где – это коэффициенты, а – неизвестная переменная.

Разработанная программа, предоставляет удобный инструмент для решения квадратных уравнений. Её использование может быть полезно в различных ситуациях, где требуется нахождение корней таких уравнений. Данное программное решение может быть применено:

1. Программа может быть использована преподавателями и студентами в процессе изучения квадратных уравнений. Она позволяет им проверять свои расчеты и получать надежные результаты.
2. В инженерных расчетах часто возникают задачи, где требуется нахождение корней квадратных уравнений. Программа позволяет инженерам быстро и точно решать такие задачи, что помогает в проектировании и анализе систем.
3. В финансовой и экономической аналитике возникают задачи, где необходимо находить значения переменных в квадратных уравнениях. Программа может быть использована для моделирования и анализа финансовых данных.
4. Алгоритмы и программы в компьютерных науках часто требуют нахождения корней квадратных уравнений. Программа может быть интегрирована в различные компьютерные приложения и алгоритмы для автоматизации и оптимизации вычислений.
5. Программа может быть полезна для людей, которым требуется решать квадратные уравнения в повседневной жизни. Например, при планировании финансовых расчетов, ремонта или других задач, где требуется нахождение корней уравнений.

Разработанная программа использует метод дискриминанта для определения количества и значений корней квадратного уравнения. Она производит вычисления согласно математическим формулам, позволяющим найти корни уравнения. Стоит отметить что разработанный алгоритм программы рассчитан на работу с вещественными корнями квадратных уравнений. Она не предназначена для решения уравнений с комплексными корнями. Кроме того, программа предполагает правильный ввод коэффициентов a, b и c пользователем. Некорректные или неправильно введенные значения могут привести к ошибочным результатам.

Для запуска программы для решения квадратных уравнений необходимо иметь устройство с операционной системой семейства Windows, поддерживающей выполнение программ на языке C#.

# ТЕХНИЧЕСКОЕ ОПИСАНИЕ ПРОЕКТА

Проект реализован на языке программирования C# и использует среду разработки Visual Studio. C# является объектно-ориентированным языком программирования, разработанным для платформы .NET, и широко применяется для создания различных типов приложений, включая Windows-приложения.

В данном проекте использовалась классическая архитектура Windows Forms приложения. Windows Forms является одним из подходов для создания графического пользовательского интерфейса (GUI) в приложениях для операционной системы Windows. Он основан на модели событий и обработчиков событий.

Архитектура Windows Forms включает следующие основные компоненты:

1. Форма (Form): Это главное окно приложения, которое содержит элементы управления и определяет визуальное представление пользовательского интерфейса. В данном проекте класс Form1 отвечает за основную форму приложения.
2. Элементы управления (Controls): Это компоненты, которые добавляются на форму и позволяют пользователю взаимодействовать с приложением. Примеры элементов управления, используемых в данном проекте, включают текстовые поля (TextBox) для ввода коэффициентов и кнопку (Button) для запуска вычислений.
3. Обработчики событий (Event Handlers): Это методы, которые реагируют на события, генерируемые элементами управления или другими компонентами приложения. В данном проекте обработчики событий используются, например, для обработки нажатия клавиш и кликов на кнопку.
4. Диалоговые окна (Dialog Boxes): Это специальные окна, которые используются для вывода сообщений пользователю или получения дополнительной информации от пользователя. В данном проекте диалоговое окно сообщений (MessageBox) используется для вывода результатов вычислений.

Архитектура Windows Forms предоставляет удобный способ разработки приложений с графическим интерфейсом, позволяя разделить логику приложения и визуальное представление. В данном проекте код разделен на несколько методов, которые обрабатывают события и выполняют необходимые вычисления, основываясь на взаимодействии с элементами управления на форме.

Проект предоставляет пользовательский интерфейс, состоящий из текстовых полей для ввода коэффициентов a, b и c, кнопки для запуска вычислений и диалогового окна сообщений для вывода результатов. Пользователь может вводить значения коэффициентов, а затем нажать на кнопку для запуска процесса решения квадратного уравнения. Результаты выводятся в диалоговом окне сообщений. На рисунке 2.1 представлен пользовательский интерфейс.

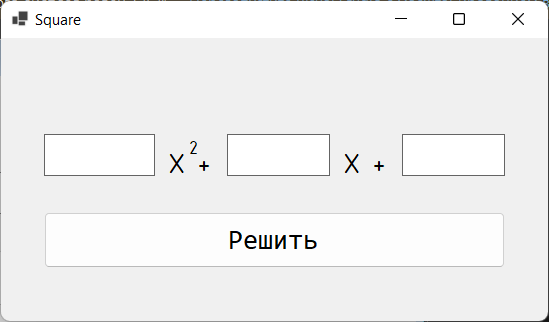


Рисунок 2.1 – пользовательский интерфейс программы

Проект обрабатывает пользовательский ввод, чтобы гарантировать корректность ввода данных. Он проверяет, что вводимые символы являются допустимыми (цифры, знаки арифметических операций и запятая), и предотвращает ввод некорректных символов. Программа также проверяет правильность порядка ввода знаков арифметических операций и запятой, чтобы соответствовать правилам квадратных уравнений.

На рисунке 2.2 представлен блок-схема для проверки допустимости вводимого символа в поля для ввода.

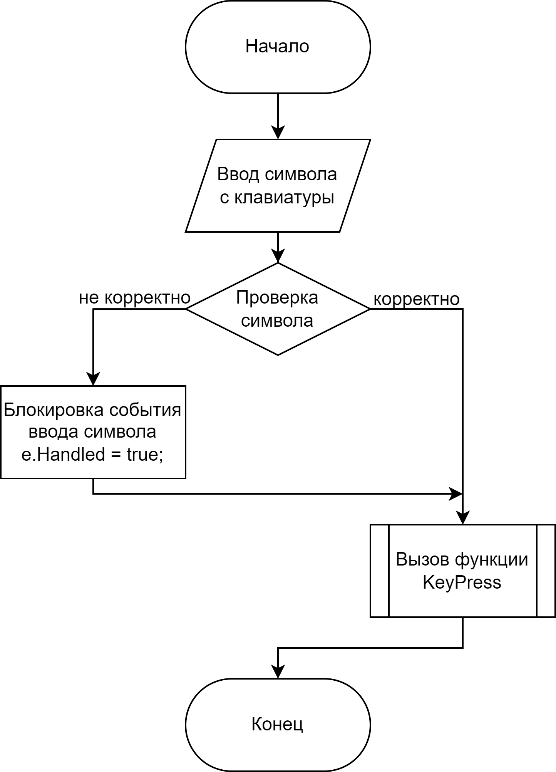


Рисунок 2.2 – блок-схема для проверки введенных символов

Проект предусматривает обработку ошибок, которые могут возникнуть в процессе вычислений. Если пользователь вводит некорректные значения коэффициентов или происходит ошибка во время вычислений, программа выводит соответствующее сообщение об ошибке с описанием проблемы. Это помогает пользователям понять, что пошло не так и какие шаги могут быть предприняты для исправления ситуации.

На рисунке 2.3 представлена блок-схема, иллюстрирующая основные шаги и логику работы алгоритма для решения квадратных уравнений. Блок-схема помогает визуализировать последовательность операций и принятых решений в процессе выполнения программы. Она служит вспомогательным инструментом для понимания работы программы и может быть использована в качестве справочного материала при анализе или сопровождении проекта

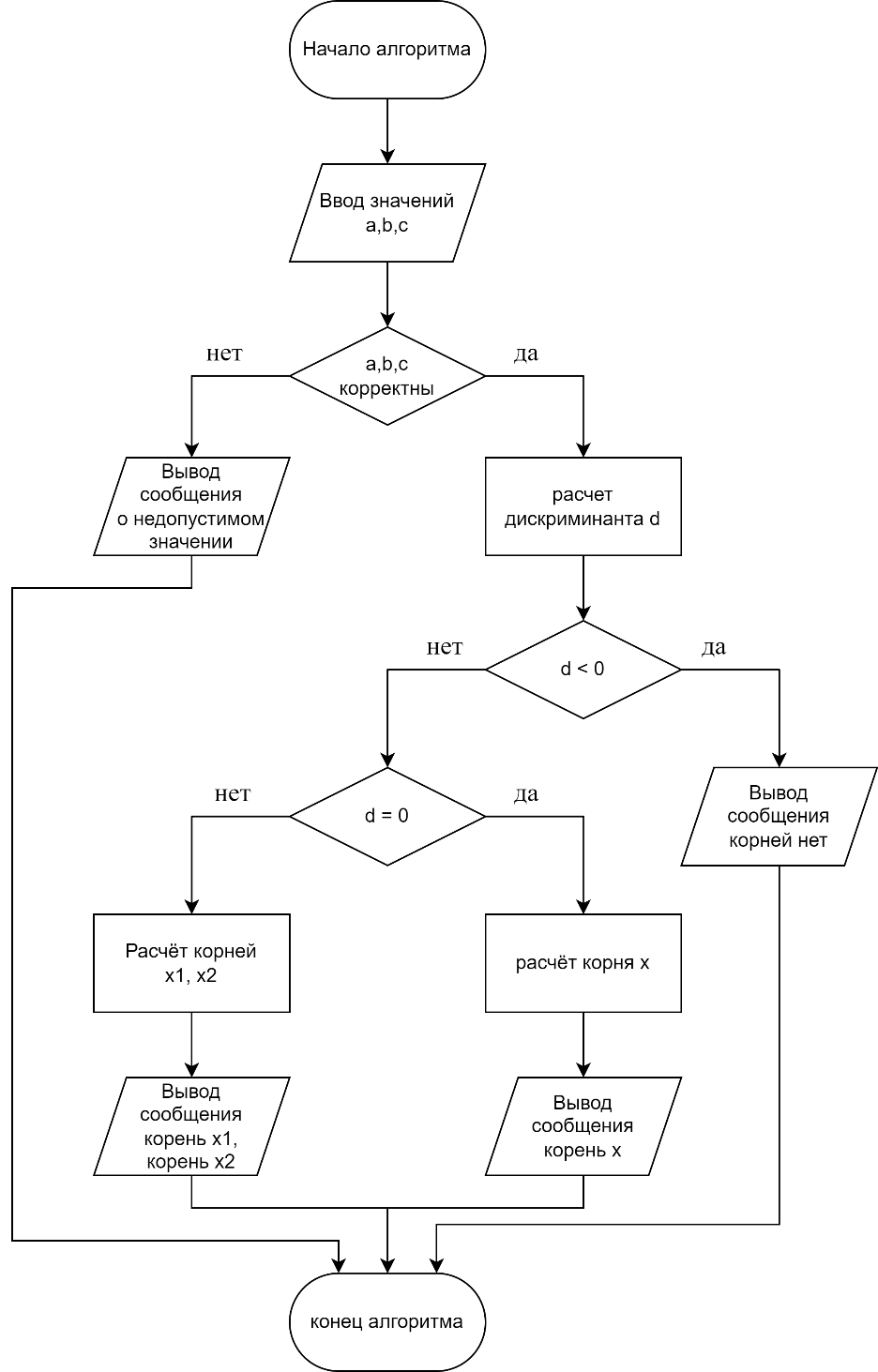


Рисунок 2.3 – блок-схема алгоритма

# РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

## 3.1 Руководство по установке приложения

1. Подготовка к установке:
   1. Убедитесь, что ваш компьютер соответствует минимальным системным требованиям для работы приложения. Это включает операционную систему Windows и наличие платформы .NET Framework (версия, указанная в требованиях приложения).
   2. Убедитесь, что у вас есть установочный файл приложения.
2. Запуск установки:
   1. Найдите установочный файл приложения (обычно с расширением .exe) и дважды щелкните по нему. На рисунке 3.1.1 представлен установочный файл



Рисунок 3.1.1 – установочный файл

* 1. Если появится предупреждение безопасности, подтвердите, что хотите запустить файл из надежного источника.

1. Установка:
   1. Появится приветственное окно мастера установки. Нажмите кнопку "Далее" для продолжения. На рисунке 3.1.2 представлено окно мастера установки.

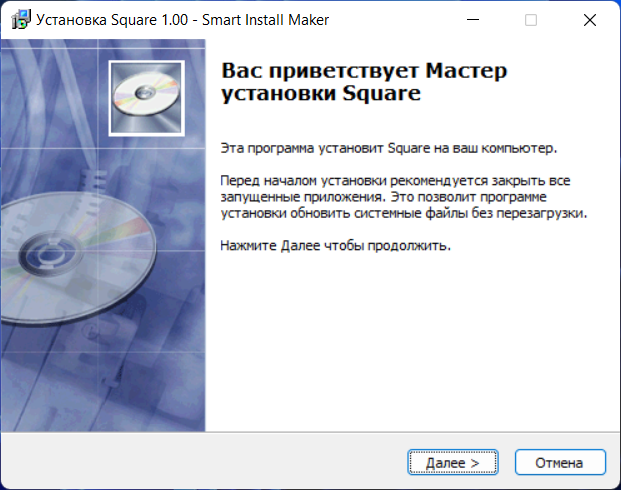


Рисунок 3.1.2 – окно мастера установки

1. Выбор пути установки:

4.1 Укажите путь, по которому вы хотите установить приложение, или оставьте значение по умолчанию. Нажмите "Далее".   
На рисунке 4.2 представлено окно с выбором папки для установки.

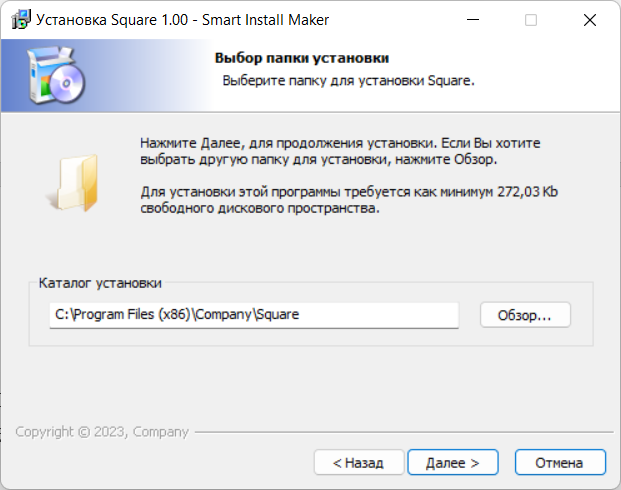


Рисунок 3.1.3 – окно с выбором папки для установки

5. Готовность к установке:

5.1 Появится окно с подтверждением настроек установки. Проверьте, что все параметры указаны верно. Если все правильно, нажмите "Установить" или "Далее" для начала установки. На рисунке 3.1.4 представлено данное окно.

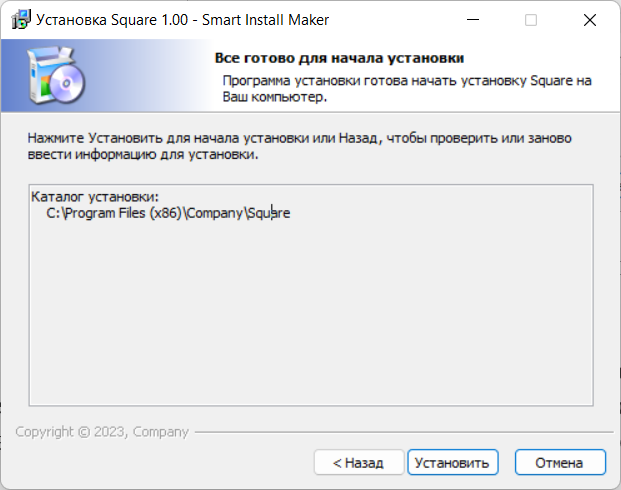


Рисунок 3.1.4 – окно подтверждения установки

6 Процесс установки

6.1 Мастер установки начнет процесс копирования файлов и установки приложения на ваш компьютер. Дождитесь завершения этого процесса.

6.2 По завершении установки появится окно с информацией о завершении процесса. На рисунке 3.1.5 представлено данное окно.

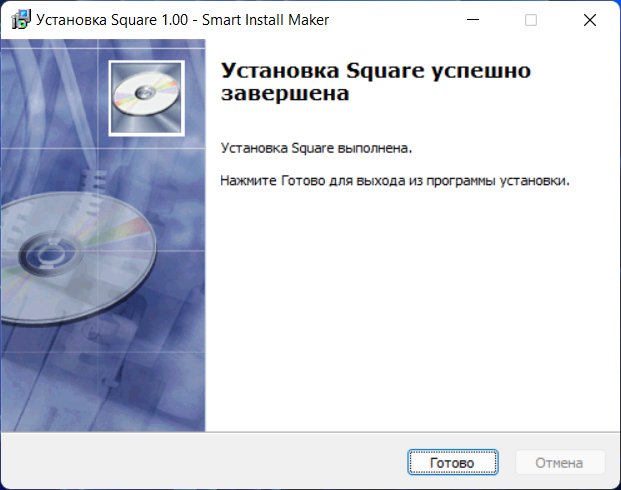


Рисунок 3.1.5 – окно успешной установки

После завершения установки вы можете начать использовать приложение для решения квадратных уравнений.

## 3.2 Руководство по эксплуатации

Приложение для решения квадратных уравнений предназначено для автоматизации процесса нахождения корней квадратных уравнений вида  
. Квадратные уравнения широко применяются в математике, физике, экономике и других областях, поэтому данное приложение может быть полезным инструментом для решения таких задач.

Основная цель приложения – упростить и ускорить процесс решения квадратных уравнений для пользователей. Вместо ручного вычисления и применения формул, пользователи могут вводить значения коэффициентов   
a, b, c в соответствующие текстовые поля приложения. После нажатия кнопки "Решить", приложение производит вычисления и выводит результаты на экран. На рисунке 3.2.1 представлено изображение программы.

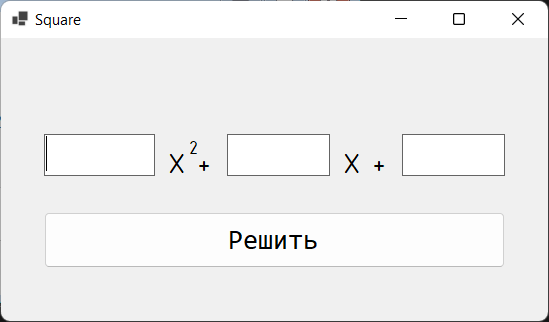


Рисунок 3.2.1 – окно программы

Приложение обладает следующими основными функциями:

1. Ввод значений коэффициентов a, b и c квадратного уравнения.
2. Проверка правильности введенных данных и обработка ошибок при некорректном вводе.
3. Автоматическое вычисление дискриминанта и нахождение корней квадратного уравнения.
4. Вывод результатов на экран, включая значения дискриминанта и корней уравнения.
5. Обработка различных сценариев, включая случаи, когда дискриминант меньше нуля, равен нулю или больше нуля.

Далее продемонстрированы результаты выполнения программы при задании различных значению.

На рисунке 3.2.2 представлено окно результата выполнения программы при значении дискриминанта больше нуля.

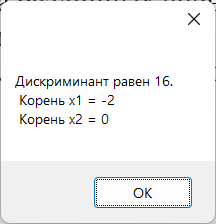


Рисунок 3.2.2 – результат работы программы при значении дискриминанта больше нуля

На рисунке 3.2.3 представлен результат программы при нулевом значении дискриминанта.

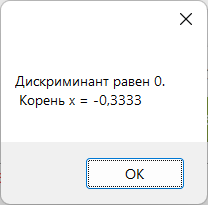


Рисунок 3.2.3 – результат программы при значении нулевом значении дискриминанта

На рисунке 3.2.4 представлен результат работы программы при отрицательном значении дискриминанта.

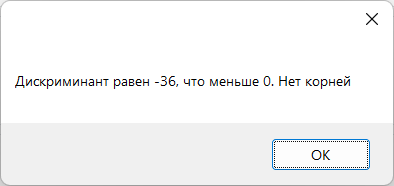


Рисунок 3.2.4 – результат работы программы при отрицательном значении дискриминанта

Приложение для решения квадратных уравнений включает в себя обработку исключительных ситуаций, которые могут возникнуть во время работы программы. В случае возникновения ошибок, пользователю будет предоставлена информация о проблеме, и приложение будет предпринимать соответствующие действия для предотвращения сбоев или неправильных результатов.

Например, при попытке ввода некорректных символов, таких как буквы или специальные символы, в поля для коэффициентов a, b и c, приложение будет отображать сообщение об ошибке и предотвратит сохранение некорректных данных. Это помогает пользователю избежать неправильного ввода и обеспечивает корректное функционирование приложения.

Запрет на ввод более одного раза знаков "+", "-", ",": Приложение также включает проверки для предотвращения некорректного использования знаков "+", "-", и ".". В соответствии с правилами математики, ввод более одного раза этих знаков может привести к неправильным результатам или некорректной интерпретации данных.

Поэтому, если пользователь пытается ввести более одного раза знак "+" или "-", или если он пытается ввести символ "," (запятая) более одного раза в поле для коэффициентов, приложение не будет разрешать такой ввод и предотвратит сохранение некорректных данных. Пользователю будет показано сообщение об ошибке, указывающее на правила ввода и просить исправить введенные значения.

Такие проверки помогают гарантировать корректность введенных данных и предотвращают возможные ошибки в решении квадратных уравнений.

На рисунке 3.2.5 представлен скриншот программы при попытке ввода некорректных значений.

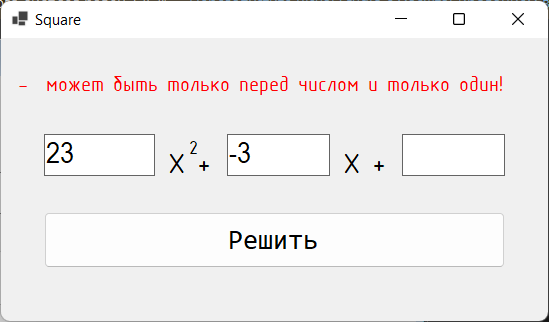


Рисунок 3.2.5 – некорректный ввод значений

В заключении можно сказать что, приложение обладает простым и интуитивно понятным интерфейсом, что делает его доступным для широкого круга пользователей, включая начинающих в области математики. Оно предоставляет быстрый и точный способ получения решений квадратных уравнений без необходимости выполнять сложные вычисления вручную.

# Контрольные вопросы

1) Дайте определение терминам: приложение, программный продукт, утилита.

Приложение: Приложение – это компьютерная программа или набор программ, разработанных для выполнения конкретных задач и предназначенных для использования конечными пользователями. Оно обеспечивает определенные функциональные возможности и может работать на различных платформах, таких как операционные системы Windows, macOS, Android или iOS. Приложения могут быть разработаны для различных целей, включая игры, утилиты, образовательные программы, программы для бизнеса и другие.

Программный продукт – это комплекс программных компонентов, разработанных для решения конкретной задачи или удовлетворения определенных потребностей пользователей или организаций. Программный продукт может включать в себя несколько приложений, модулей или библиотек, объединенных общей целью или функциональностью. Он обычно создается и продается или распространяется как единое целое, предлагая решение для конкретной проблемы или области деятельности.

Утилита – это программный инструмент, предназначенный для выполнения специфических функций или задач на компьютере. Утилиты обычно выполняются в командной строке или имеют графический интерфейс пользователя (GUI). Они могут быть предназначены для управления файлами и папками, настройки системных параметров, восстановления данных, проверки безопасности, оптимизации производительности и многого другого. Утилиты часто используются для автоматизации рутинных задач и повышения эффективности работы с компьютером.

2) «Жизненный цикл программного продукта» (ЖЦ ПП) – дайте определение и перечислите основные этапы.

Жизненный цикл программного продукта (ЖЦ ПП) - это последовательность этапов и процессов, через которые проходит программный продукт, начиная с его создания и заканчивая выводом из эксплуатации. ЖЦ ПП описывает все основные этапы, активности и роли, которые включены в процесс разработки, поддержки и управления программным продуктом.

Основные этапы ЖЦ ПП включают:

1. Анализ и планирование: в этом этапе определяются требования к программному продукту, проводится анализ рынка, конкурентных продуктов и потенциальных пользователей. Планируется бюджет, расписание и ресурсы для разработки.
2. Проектирование: на этом этапе определяется архитектура программного продукта, создается дизайн пользовательского интерфейса, определяются основные модули и компоненты системы. Проектирование также включает определение структуры данных и алгоритмов, которые будут использоваться в продукте.
3. Разработка: на этом этапе происходит фактическое программирование и создание кода программного продукта. Разработчики реализуют функциональность, проводят тестирование и отладку, чтобы обеспечить работоспособность и соответствие требованиям.
4. Тестирование и верификация: в этом этапе проводятся различные виды тестирования, включая модульное тестирование, интеграционное тестирование и системное тестирование. Целью тестирования является обнаружение ошибок и несоответствий требованиям для их исправления.
5. Внедрение и установка: на этом этапе программный продукт готов к установке и использованию. Он может быть распространен пользователем или развернут на целевой системе. Внедрение также включает обучение пользователей и поддержку при переходе к новому продукту.
6. эксплуатация и поддержка: после внедрения продукта начинается его эксплуатация. В этом этапе осуществляется поддержка продукта, включая устранение ошибок, обновления, добавление новой функциональности и техническую поддержку для пользователей.

3) Дайте определение терминам: интерфейс, пользовательский интерфейс, графический интерфейс.

Интерфейс – это средство взаимодействия между двумя или более системами, которое позволяет им обмениваться информацией, командами или данными. В контексте программного обеспечения, интерфейс определяет способ взаимодействия между пользователем и компьютерной системой.

Пользовательский интерфейс – это конкретный вид интерфейса, предназначенный для взаимодействия пользователя с компьютерной системой или программным приложением. Он обеспечивает средства ввода и вывода информации, а также позволяет пользователю выполнять операции, настраивать параметры и получать обратную связь от системы.

Графический интерфейс – это один из видов пользовательского интерфейса, который использует графические элементы, такие как иконки, кнопки, окна и меню, для визуального представления информации и обеспечения взаимодействия с пользователем. Графический интерфейс делает использование компьютерной системы более интуитивным, позволяя пользователям выполнять операции с помощью мыши, клавиатуры и других визуальных элементов, вместо ввода команд или кода.

4) Какова основная цель в составлении «Технического задания» (ТЗ)?

Основная цель составления "Технического задания" (ТЗ) заключается в установлении четких требований и спецификаций для разработки или реализации проекта, программного продукта или системы. В составленном ТЗ должны быть описаны функциональные и нефункциональные требования, а также прочие детали, необходимые для успешной реализации проекта.

5) Для решения каких основных задач предназначаются «Системы контроля версий» (СКВ)?

1. Защищает исходный код от потери. Данные хранятся на удалённом сервере, даже если разработчики удалят файлы с локального компьютера, они останутся в репозитории.
2. Обеспечивает командную работу. Программисту не надо использовать инструменты для командной работы и платить за них. Каждый может работать на своём компьютере и обновлять файлы по мере необходимости.
3. Помогает отменить изменения. В любой момент можно вернуться к контрольной точке, сравнить исходный код с текущим и обновить главную ветку после ревью.
4. Распределённая работа. Необязательно работать с проектом «наживую». Плагин может функционировать на сайте, а программисты будут спокойно создавать новую версию.

6) Перечислите известные вам СКВ?

Популярными системами управления версиями являются Git, Mercurial, SVN и Preforce.

7) Для решения каких задач предназначаются Системы управления дефектами (СУД)?

1. Документирование найденных багов в работе разрабатываемого продукта, а также обращений от пользователей, предложения для улучшений;
2. Прослеживание за процессом устранения и выполнения таких ошибок, обращений и улучшений.

8) Перечислите известные вам СУД?

Allure TestOps, EasyQA, EveryQA

9) Что такое IDE? Расшифруйте и дайте определение.

Интегрированная среда разработки (IDE) – это программное приложение, которое помогает программистам эффективно разрабатывать программный код. Оно повышает производительность разработчиков, объединяя такие возможности, как редактирование, создание, тестирование и упаковка программного обеспечения в простом для использования приложении.

10) Опишите используемую Вами IDE. Особенности и преимущества.

Мною была использована IDE Visual Studio Community.

Visual Studio Community – это интегрированная среда разработки (IDE) от компании Microsoft, предназначенная для создания различных типов приложений, включая веб-приложения, мобильные приложения, настольные приложения и многое другое. Особенности и преимущества Visual Studio Community:

1. Мощный инструментарий: Visual Studio Community предоставляет широкий набор инструментов и возможностей для разработки. Он включает редактор кода с подсветкой синтаксиса, автодополнением и функцией отладки, интегрированные средства для создания пользовательского интерфейса, инструменты тестирования и многое другое.
2. Множество поддерживаемых языков: IDE поддерживает множество популярных языков программирования, включая C#, Visual Basic.NET, F#, C++, JavaScript, Python и другие. Это позволяет разработчикам работать с предпочитаемым языком и создавать приложения на различных платформах.
3. Разработка для разных платформ: Visual Studio Community позволяет разрабатывать приложения для различных платформ, включая Windows, macOS, Android, iOS и веб-платформы. Это обеспечивает гибкость и возможность создания кросс-платформенных решений.
4. Интеграция с другими инструментами: Visual Studio Community интегрируется с другими популярными инструментами и платформами, такими как Azure, Git, Docker и многими другими. Это облегчает разработку, развертывание и управление проектами.
5. Обширное сообщество и ресурсы: Visual Studio Community имеет обширное сообщество разработчиков, где можно получить поддержку, задать вопросы и найти решения проблем. Также доступны обучающие материалы, документация и примеры кода, которые помогают разработчикам быстро освоить инструмент и повысить свою продуктивность.
6. Бесплатное использование: Visual Studio Community доступен для бесплатного использования индивидуальными разработчиками, стартапами и небольшими командами. Это делает его доступным и привлекательным для разработчиков, которые только начинают свой путь в программировании или работают над небольшими проектами.