**Отчет по занятию №3.**

**Разработка программы с разветвлениями**

**«Решение квадратного уравнения»**

**Контрольные вопросы по теме: «Строки»**

1. Что представляет собой переменная типа System::String? Краткая характеристика типа. Объявление, инициализация и использование строк типа System::String.

**Ответ:** переменная типа System::String представляет собой строку в .NET Framework, которая представляет последовательность символов Unicode. Объявление строки типа System::String выглядит так: `System::String^ str = "Hello";`. Использование строк типа System::String включает операции сравнения, конкатенации, получение длины и т.д.

1. Что представляет собой переменная типа System::Text::StringBuilder? Чем отличаются строки типов System::String и System::Text::StringBuilder? Объявление, инициализация и использование строк типа System::Text::StringBuilder.

**Ответ:** переменная типа System::Text::StringBuilder представляет изменяемую последовательность символов. Основное отличие от System::String - StringBuilder можно изменять, не создавая новую копию строки каждый раз. Объявление и использование StringBuilder выглядят так: `System::Text::StringBuilder sb; sb.Append("Hello");`.

1. Назначение и краткая характеристика структуры System::Char.

**Ответ:** структура System::Char представляет собой одиночный символ Unicode. Она содержит базовые методы для работы с символами.

1. Основные поля, свойства и методы классов System::String и System::Text::StringBuilder.

**Ответ:** у класса System::String есть свойства, такие как Length (длина строки) и методы для работы со строками, например, Compare, Concat, Substring и т.д. У класса System::Text::StringBuilder есть методы для добавления, удаления, замены символов и т.д.

1. Основные поля и методы структуры System::Char.

**Ответ:** основные методы структуры System::Char включают IsDigit, IsLetter, IsWhiteSpace и т.д.

1. Что представляет собой строка в стиле C? Краткая характеристика этих строк. Объявление, инициализация и использование строк в стиле C.

**Ответ:** строка в стиле C представляет собой массив символов, завершающийся нулевым символом '\0'. Объявление и инициализация строк в стиле C: `char str[] = "Hello";`.

1. Функции стандартной библиотеки для работы со строками и символами.

**Ответ:** некоторые функции стандартной библиотеки для работы со строками и символами включают strcpy, strcat, strlen для строк и isalpha, isdigit для символов.

1. Характеристика строк C++, представляемых библиотечным классом string. Объявление, инициализация и использование строк C++.

**Ответ:** строки в C++ представляются библиотечным классом string, который предоставляет удобные методы для работы со строками. Объявление и использование строки в C++: `std::string str = "Hello";`.

1. Функции библиотечного класса string для работы со строками C++.

**Ответ:** некоторые методы библиотечного класса string включают append, length, find, substr для работы со строками.

1. Как преобразуются строки различных типов в C++?

**Ответ:** преобразование строк различных типов в C++ осуществляется с помощью методов преобразования, например, с помощью to\_string для преобразования чисел в строки и stoi для преобразования строк в числа.

**Контрольные вопросы по теме: «Регулярные выражения»**

1. Что представляют собой регулярные выражения? Поясните смысл использования регулярных выражений для обработки текста? Какие задачи обработки текста решаются с помощью регулярных выражений?

**Ответ:** регулярные выражения - это мощный инструмент для работы с текстом, позволяющий осуществлять поиск и обработку текстовой информации на основе шаблонов. Используя регулярные выражения, можно осуществлять поиск определенных шаблонов в тексте, заменять их на другие строки, разбивать текст на подстроки и многое другое. Регулярные выражения позволяют задавать сложные шаблоны для поиска и обработки текста, что делает их удобным инструментом для работы с текстовой информацией.

1. Основные методы класса System::Text::RegularExpressions::Regex.

**Ответ:** основные методы класса System::Text::RegularExpressions::Regex: - Match: используется для поиска первого вхождения шаблона в строке. - Matches: используется для поиска всех вхождений шаблона в строке. - Replace: используется для замены вхождений шаблона на указанную строку. - Split: используется для разделения строки на подстроки с помощью заданного шаблона.

1. Назовите и охарактеризуйте основные элементы языка регулярных выражений. Приведите примеры.

**Ответ:** Основные элементы языка регулярных выражений:

- Литералы: представляют собой символы, которые должны точно соответствовать в тексте. Пример: "abc" будет соответствовать строке "abc".

- Метасимволы: специальные символы, которые представляют группы символов или определенные позиции в тексте.

Примеры: - "." соответствует любому символу, кроме символа новой строки. - "\d" соответствует любой цифре.

- Классы символов: позволяют задать группы символов, которые могут соответствовать одному символу.

Пример: "[a-z]" соответствует любой строчной букве от a до z.

- Квантификаторы: определяют количество повторений предыдущего элемента. Примеры: - "\*" - ноль или более повторений. - "+" - одно или более повторений.

- Группы: позволяют объединять части регулярного выражения и применять к ним операции. Пример: "(abc)+" будет соответствовать одному или более повторениям строки "abc".

Регулярные выражения позволяют решать задачи поиска, замены, разделения текста, валидации данных и многие другие задачи, связанные с обработкой текста.

**Контрольные вопросы по теме: «Ввод-вывод»**

1. Основные свойства и методы класса System::Console, предоставляемые для организации консольного ввода-вывода.

**Ответ:** класс System::Console предоставляет основные методы и свойства для организации консольного ввода-вывода в C++. Некоторые из них включают Read, Write, WriteLine для ввода и вывода данных, а также свойства для управления цветом текста и фона на консоли.

1. Основные возможности, предоставляемые перечислениями System::ConsoleKey и System::ConsoleModifiers и структурой System::ConsoleKeyInfo для организации ввода-вывода.

**Ответ:** перечисления System::ConsoleKey и System::ConsoleModifiers предоставляют возможности для работы с клавишами на клавиатуре. Структура System::ConsoleKeyInfo используется для представления информации о нажатой клавише, включая код клавиши, символ и состояние модификаторов.

1. В чем смысл форматирования строк, в каких случаях оно применяеься? Синтаксис элемента форматирования, его составляющие.

**Ответ:** форматирование строк позволяет управлять выводом текста, выравниванием, числовым форматированием и другими аспектами визуализации данных. Синтаксис элемента форматирования состоит из фигурных скобок {} и опциональных параметров внутри них, например, {0} или {0:C}.

1. Строки стандартных числовых форматов: структура строк формата, спецификаторы стандартных числовых форматов, примеры использования.

**Ответ:** строки стандартных числовых форматов определяют способы представления чисел, такие как десятичные, валютные и процентные форматы. Примеры включают "C" для валют, "D" для десятичного и "P" для процентного формата.

1. Строки настраиваемых числовых форматов: описатели настраиваемых числовых форматов и примеры их использования.

**Ответ:** настраиваемые числовые форматы позволяют задать собственные шаблоны форматирования чисел, используя спецификаторы и разделители. Например, "#,##0.00" для формата денежной суммы с разделителями.

1. Форматирование с помощью управляющих последовательностей: управляющие последовательности, их интерпретация и применение.

**Ответ:** управляющие последовательности позволяют вставлять специальные символы и управляющие команды в строки, такие как символ новой строки (\n) или символ табуляции (\t).

1. Ввод-вывод с использованием элементов управления графического интерфейса. Использование стандартного окна MessageBox для вывода сообщений.

**Ответ:** для вывода сообщений в графическом интерфейсе можно использовать стандартное окно MessageBox, которое предоставляет удобный способ вывода сообщений пользователю.

1. Обработка событий нажатия клавиш для элементов управления: используемые события, свойства классов KeyPressEventArgs и KeyEventArgs, перечисление Keys.

**Ответ:** обработка событий нажатия клавиш для элементов управления включает использование событий KeyPress и KeyDown, доступ к информации о нажатой клавише через свойства KeyPressEventArgs и KeyEventArgs, а также перечисление Keys для представления клавиш.

1. Использование свойств классов System::Globalization::NumberFormatInfo и System::Globalization::CultureInfo для получения сведений о языке и региональных параметрах при форматировании и анализе числовых значений.

**Ответ:** классы System::Globalization::NumberFormatInfo и System::Globalization::CultureInfo позволяют получать информацию о языке и региональных параметрах для корректного форматирования числовых значений, таких как разделители разрядов и десятичный разделитель.

1. Защита от дурака и способы ее реализации. Примеры использования.

**Ответ:** защита от дурака включает проверку ввода данных на корректность и обработку ошибок ввода, чтобы избежать некорректной работы программы. Примеры включают проверку диапазона значений или типов данных.

1. Ввод-вывод данных в языке С++: объекты для ввода и вывода, их использование.

**Ответ:** для ввода-вывода данных в C++ можно использовать объекты классов для потоков ввода (istream) и вывода (ostream), такие как cin и cout, а также методы для работы с файлами, например, ifstream и ofstream.

1. Ввод-вывод данных в языке C: функции для ввода и вывода, спецификаторы формата и их параметры.

**Ответ:** в языке C функции для ввода и вывода данных включают scanf и printf, которые используют спецификаторы формата для определения типа данных и их вывода или ввода. Например, %d для целых чисел и %f для чисел с плавающей запятой.

**Контрольные вопросы по теме: «Операторы управления и циклы»**

1. Что называется выражением?

**Ответ:** выражение - это комбинация операторов, операндов и других выражений, которая может быть вычислена до получения конечного значения.

1. Для чего используются круглые скобки в выражениях?

**Ответ:** круглые скобки в выражениях используются для указания порядка выполнения операций и для явного выделения части выражения.

1. Перечислите арифметические операторы и выполняемые ими функции в порядке убывания приоритетов.

**Ответ:** арифметические операторы в порядке убывания приоритетов: \*, /, %, +, -.

1. Какие операторы используются в операциях отношения?

**Ответ:** операторы отношения: == (равно), != (не равно), < (меньше), > (больше), <= (меньше или равно), >= (больше или равно).

1. Какие операторы используются в логических выражениях?

**Ответ:** логические операторы: ! (логическое НЕ), && (логическое И), || (логическое ИЛИ).

1. Какие операторы могут использоваться в поразрядных операциях над числовыми выражениями?

**Ответ:** в поразрядных операциях над числовыми выражениями используются операторы: & (поразрядное И), | (поразрядное ИЛИ), ^ (поразрядное исключающее ИЛИ), ~ (поразрядное НЕ).

1. В каких выражениях (операциях) используются операторы: ==, !=, , <=, >= ?

**Ответ:** операторы ==, !=, <, >, <=, >= используются в выражениях сравнения.

1. В каких выражениях (операциях) используются операторы: !, &&, || ?

**Ответ:** операторы !, &&, || используются в логических выражениях.

1. В каких выражениях (операциях) используются операторы: &, |, ^, <> ?

**Ответ:** операторы &, |, ^, <<, >> используются в поразрядных операциях.

1. Результатом каких выражений (операций) могут служить значения true и false?

**Ответ:** значения true и false могут быть результатом логических выражений.

1. Какие действия выполняет оператор присваивания?

**Ответ:** оператор присваивания (=) присваивает значение правого операнда левому операнду.

1. Какой(ие) символ(ы) используется(ются) для записи полного оператора присваивания?

**Ответ:** для записи полного оператора присваивания используется символ "=".

1. Допускаются ли сокращенные операторы присваивания?

**Ответ:** да, допускаются сокращенные операторы присваивания, такие как "+=", "-=", "\*=", "/=" и другие.

1. Что представляет собой блок операторов?

**Ответ:** блок операторов представляет собой группу операторов, заключенных в фигурные скобки "{ }", которые выполняются последовательно.

1. Каким(и) символом(и) в тексте программы выделяется однострочный комментарий?

**Ответ:** однострочный комментарий выделяется символом "//" в начале строки.

1. Каким(и) символом(и) в тексте программы выделяется многострочный комментарий?

**Ответ:** многострочный комментарий выделяется символами "/\*" в начале и "\*/" в конце блока комментария.

1. Как разбить один длинный оператор на несколько строк, т.е. разместить его на нескольких строках?

**Ответ:** длинный оператор можно разделить на несколько строк, используя символ обратного слэша "\" в конце строки для продолжения на следующей строке.

1. Как разместить два (или более) коротких оператора в одной строке?

**Ответ:** два (или более) коротких оператора можно разместить в одной строке, разделяя их точкой с запятой ";" или запятой ",".

**Результат работы программы:**



