

ГОСУДАРСТВЕННОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Экономика и управление»

ТЕОРИЯ И ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

*Методические рекомендации к лабораторным работам
для студентов направления подготовки 27.03.05 «Инноватика»
дневной формы обучения*

Электронная библиотека Белорусско-Российского университета
<http://e.biblio.bru.by/>



Могилев 2018

УДК 004.4
ББК 32.965
Т 338

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Экономика и управление» «23» февраля 2018 г.,
протокол № 6

Составитель ст. преподаватель О. А. Пичугова

Рецензент канд. экон. наук, доц. Н. С. Желток

Методические рекомендации предназначены для студентов направления
подготовки 27.03.05 «Инноватика» дневной формы обучения.

Учебно-методическое издание

ТЕОРИЯ И ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

Ответственный за выпуск	И. В. Ивановская
Технический редактор	С. Н. Красовская
Компьютерная верстка	Е. С. Лустенкова

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 36 экз. Заказ №

Издатель и полиграфическое исполнение:
Государственное учреждение высшего профессионального образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 24.01.2014.
Пр. Мира, 43, 212000, Могилев.

© ГУ ВПО «Белорусско-Российский
университет», 2018



Содержание

1	Порядок выполнения и защиты лабораторных работ.....	4
2	Знакомство с интегрированной средой разработки приложений. Разработка простого консольного приложения.....	5
3	Основные возможности интегрированной среды разработки при работе с консольными приложениями.....	10
4	Лабораторная работа № 1. Программа линейной структуры.....	13
5	Лабораторная работа № 2. Программа с ветвлениями.....	14
6	Лабораторная работа № 3. Программа с циклами.....	16
7	Лабораторная работа № 4. Программа с одномерными массивами.....	17
8	Лабораторная работа № 5. Программа многомерными массивами.....	18
9	Лабораторная работа № 6. Программа с функциями.....	19
10	Лабораторная работа № 7. Отладка программ.....	20
11	Лабораторная работа № 8. Программа с текстовыми файлами.....	21
12	Лабораторная работа № 9. Программа с двоичными файлами.....	23
13	Лабораторная работа № 10. Программа с динамическими массивами...	24
14	Лабораторная работа № 11. Программа со структурами.....	24
15	Лабораторная работа № 12. Программа с элементами управления: окно ввода, командная кнопка и надпись.....	25
16	Лабораторная работа № 13. Программа с элементами управления: список и график.....	26
17	Лабораторная работа № 14. Программа с элементами управления: многостраничный компонент, таблица, зависимый и независимый переключатель, группа, окна диалога.....	27
	Список литературы.....	28



1 Порядок выполнения и защиты лабораторных работ

Перед выполнением лабораторной работы студент должен ознакомиться с соответствующей темой конспекта лекций, а в случае необходимости с рекомендуемой по дисциплине литературой.

В начале занятий он получает задание для выполнения лабораторной работы от преподавателя. После этого студенту необходимо разработать алгоритм решения задачи и набрать текст программы в редакторе кода интегрированной среды разработки приложений C++ Builder. Затем программа должна быть проверена на предмет наличия ошибок синтаксиса (этап компиляции), ошибок в использовании библиотечных и пользовательских функций (этап компоновки), а также логических ошибок или ошибок алгоритма, которые выявляются уже на стадии выполнения. После внесения необходимых исправлений в исходный код, он вновь подвергается тестированию до тех пор, пока не будет получен правильный результат.

Если алгоритм программы имеет несколько вычислительных ветвей, то каждая ветвь программы должна быть протестирована отдельно, а результаты тестирования должны быть зафиксированы в отчете.

По окончании процесса тестирования студент должен оформить отчет по лабораторной работе, который включает в себя задание на лабораторную работу, схему алгоритма, распечатку текста программы и результаты ее выполнения.

Текст программы должен содержать необходимый объем комментариев, указывающих назначение программы и ее отдельных частей, метод решения, смысл основных используемых переменных. Идентификаторы переменных должны быть по возможности близки к аббревиатурам названий этих переменных. Схема алгоритма должна быть изображена в соответствии с ГОСТ 19.701–90.

Отчет должен быть оформлен на листах формата А4 и подшит в папку. Папка должна быть подписана с указанием дисциплины, группы и фамилии студента.

К защите лабораторной работы допускаются только студенты, выполнившие работу и оформившие отчет. Защита проходит в форме устного и письменного собеседования, когда студент отвечает на вопросы преподавателя, которые приведены в данных методических рекомендациях в списке контрольных вопросов к каждой лабораторной работе, а также дополняет свои ответы письменно примерами программного кода. В случае успешной защиты преподаватель ставит на отчете свою подпись и дату защиты.



2 Знакомство с интегрированной средой разработки приложений. Разработка простого консольного приложения

Лабораторный практикум по дисциплине начнем со знакомства с интегрированной средой разработки (ИСР) C++ Builder. Она содержит все необходимое для проектирования, запуска и тестирования приложений и позволяет облегчить процесс создания программ. ИСР интегрирует в себе редактор кодов, отладчик, инструментальные панели, редактор изображений, инструментарий баз данных.

Запустите ИСР, выполнив щелчок по ярлыку на рабочем столе или через меню **Пуск**. На экране появится основное диалоговое окно среды, представленное в верхней части рисунка 1. Под ним располагаются другие окна, предназначенные для разработки программ в стиле визуального объектно-ориентированного программирования. Закроем их командой **File | Close All (Файл | Закрывать все)** или кнопками закрытия в правом верхнем углу каждого окна, поскольку программы первой части лабораторного практикума будут разрабатываться как консольные приложения.

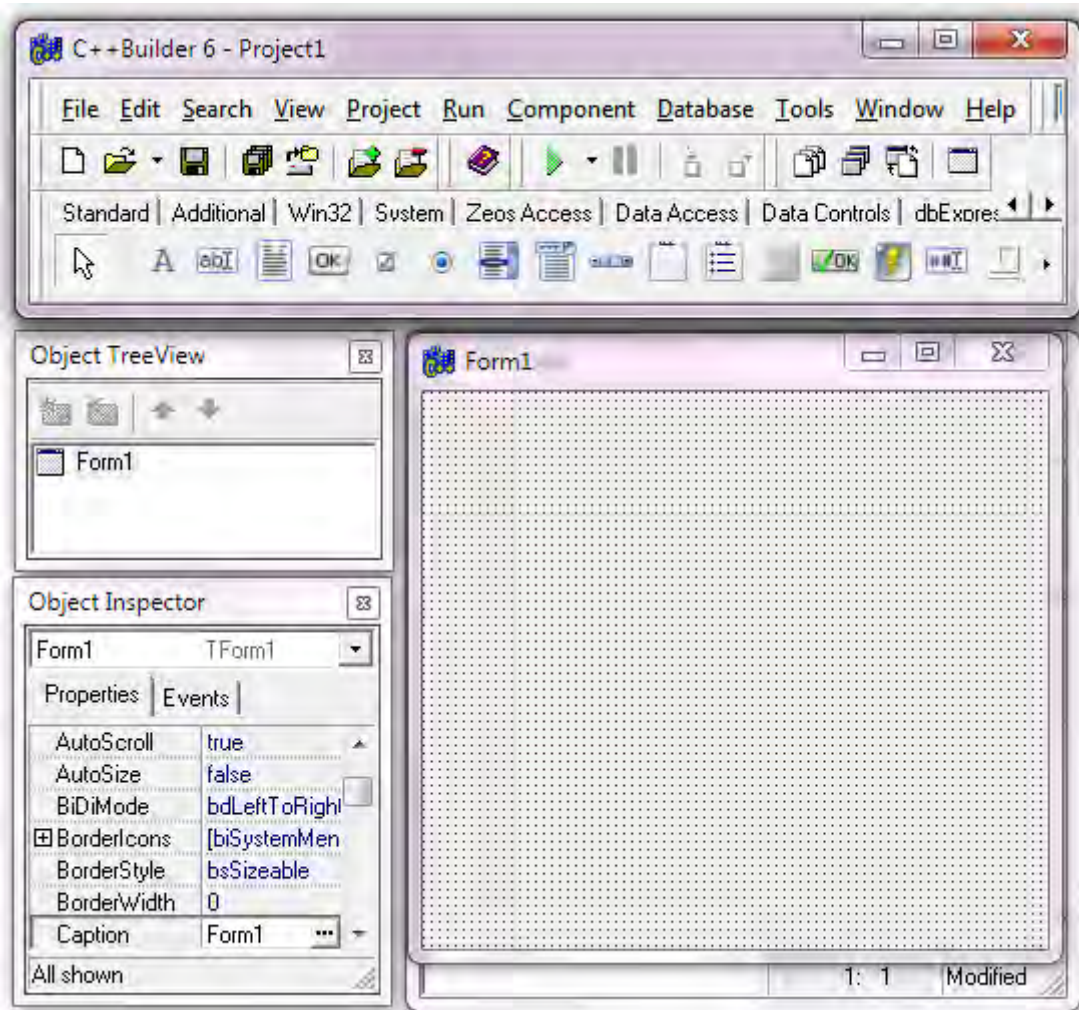


Рисунок 1 – Основное диалоговое окно среды разработки

Выполните команду меню **File | New | Other...** (**Файл | Новый | Другие...**). На экране откроется диалоговое окно **New Items (Новые Элементы)**, представленное на рисунке 2, в котором нужно выбрать пиктограмму **Console Wizard (Мастер Консоли)** и подтвердить выбор кнопкой **OK**.



Рисунок 2 – Диалоговое окно **New Items (Новые Элементы)**

В появившемся диалоговом окне **Мастера Консоли** (рисунок 3) подтвердите выбор языка **C++** и консольного приложения кнопкой **OK**.

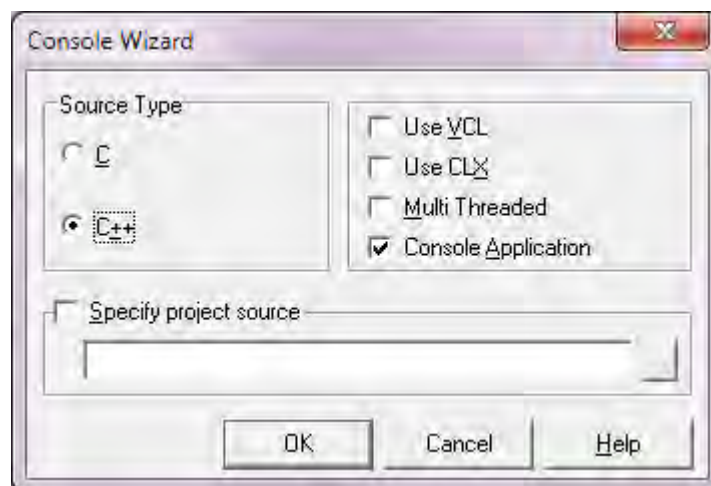
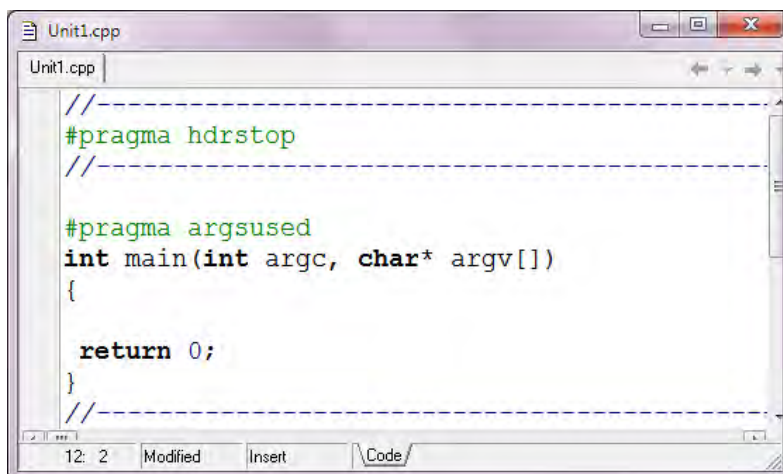


Рисунок 3 – Диалоговое окно **Console Wizard (Мастер консоли)**

На экране появится окно редактора кода с функцией **main()**, представленное на рисунке 4.



```
Unit1.cpp
Unit1.cpp
//-----
#pragma hdrstop
//-----

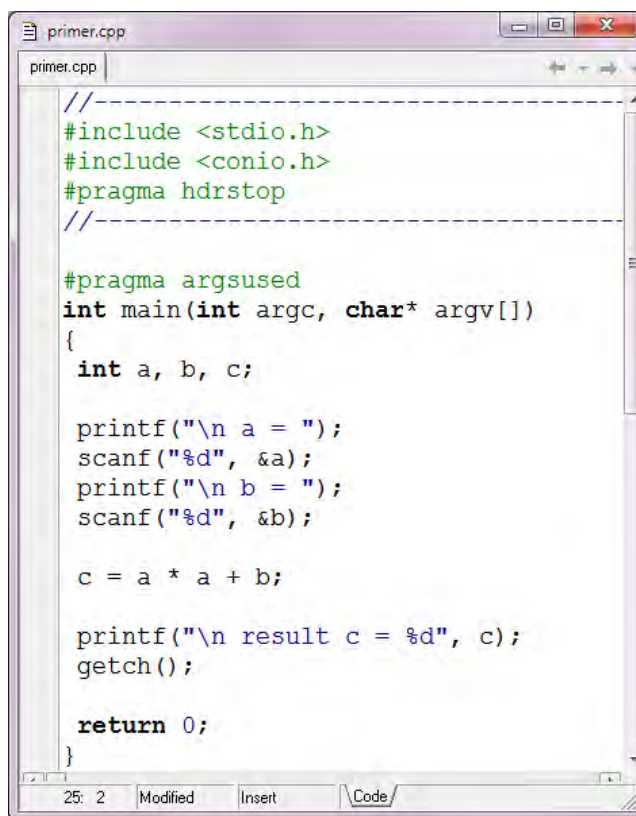
#pragma argsused
int main(int argc, char* argv[])
{

    return 0;
}
//-----
12: 2 Modified Insert \Code/
```

Рисунок 4 – Окно редактора кода

Все, что требуется для организации простого *пользовательского интерфейса* программы языка Си++, например, ввод чисел в программу и их отображение на экране, выделено в специальные *библиотеки*, которые имеются для большинства операционных систем и аппаратных платформ, благодаря чему одна и та же программа на Си++ может работать на разных типах платформ – надо только сменить библиотеку и заново выполнить компиляцию.

Разработаем наше первое простое приложение линейной структуры, которое выводит на экран результат вычисления выражения $c = a^2 + b$, где значения переменных a и b пользователь вводит с клавиатуры. Окно редактора кода с текстом программы представлено на рисунке 5.



```
primer.cpp
primer.cpp
//-----
#include <stdio.h>
#include <conio.h>
#pragma hdrstop
//-----

#pragma argsused
int main(int argc, char* argv[])
{
    int a, b, c;

    printf("\n a = ");
    scanf("%d", &a);
    printf("\n b = ");
    scanf("%d", &b);

    c = a * a + b;

    printf("\n result c = %d", c);
    getch();

    return 0;
}
25: 2 Modified Insert \Code/
```


Рисунок 5 – Окно редактора кода с текстом программы


В самое начало программного кода добавляем директивы компилятора. Это директивы подключения **#include** заголовочных файлов стандартной библиотеки ввода/вывода **<stdio.h>** и библиотеки консольного ввода/вывода **<conio.h>**, в которых размещены объявления библиотечных функций.

Директива компилятора **#pragma hdrstop** предназначена для ускорения процесса повторной компиляции программы, состоящей из нескольких модулей. Директивы подключения стандартных заголовочных файлов располагаются до директивы **#pragma hdrstop**, а директивы подключения пользовательских заголовочных файлов – после нее. Она останавливает добавление стандартных заголовочных файлов в специальный перечень заголовочных файлов препроцессора. При повторной компиляции они просто берутся из этого файла, что и позволяет ускорить процесс компиляции.

Первый оператор внутри функции **main()** объявляет необходимые переменные. Библиотечная функция **printf()** выводит на экран подсказку для ввода значения переменной **a**, а следующая за ней функция **scanf()** вводит в переменную **a** то значение, которое пользователь набрал на клавиатуре. Затем те действия выполняются с переменной **b**. Следующий оператор вычисляет значение переменной **c**, после чего функция **printf()** выводит его на экран.

Функция **getch()** предназначена для ввода одиночного символа, и пока она ожидает ввода, результаты работы программы будут видны на экране. Она добавлена для того, чтобы пользователь увидел результат до того, как окно работающего консольного приложения будет закрыто. Оператор **return 0;** сообщает операционной системе о нормальном завершении работы программы.

Командой меню **File | Save As...** (**Файл | Сохранить как...**) или кнопкой  на панели управления сохраним нашу программу в нужной папке, дав ей имя **primer**. Расширение **cpp** среда разработки добавит автоматически.

Теперь программа должна пройти процессы компиляции и компоновки, для того чтобы получить исполняемый файл с расширением ***.exe**. Можно выполнить их по отдельности командами меню **Project (Проект)**. Но можно поступить проще, выбрав команду главного меню **Run | Run (Выполнение | Выполнить)** или нажав клавишу **F9**, или кнопку  панели управления (рисунок 6). Результаты работы программы представлены на рисунке 7.

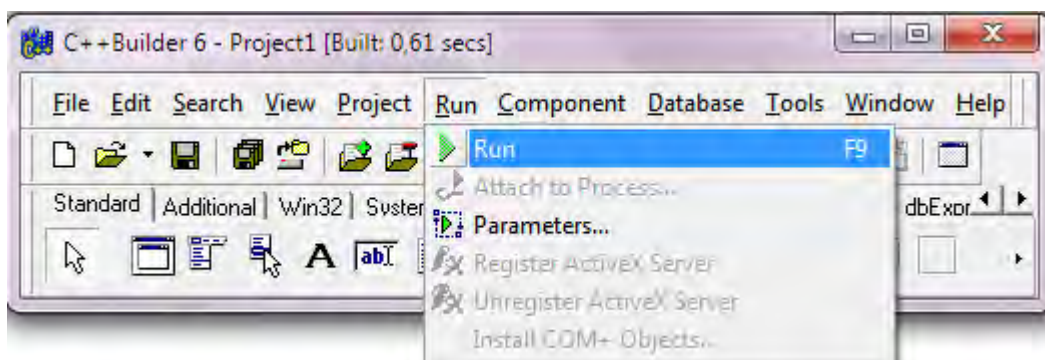


Рисунок 6 – Запуск программы на выполнение командой меню **Run**

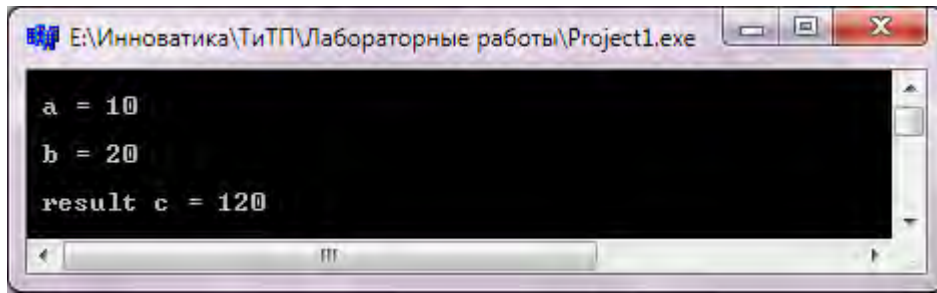


Рисунок 7 – Окно работающего консольного приложения

Мы видим, что для введенных пользователем значений исходных данных нашей программы **10** и **20** значение переменной $c = 120$. Нажмите любую клавишу на клавиатуре, чтобы закрыть это окно и вернуться в редактор кода.

Для завершения работы с нашим первым приложением выполните команду меню **File | Exit (Файл | Выход)**. При этом на экране появится окно, представленное на рисунке 8 с предложением сохранить изменения в проекте. Особой необходимости в этом для такого простого консольного приложения нет, поэтому нажмите кнопку **No (Нет)**.

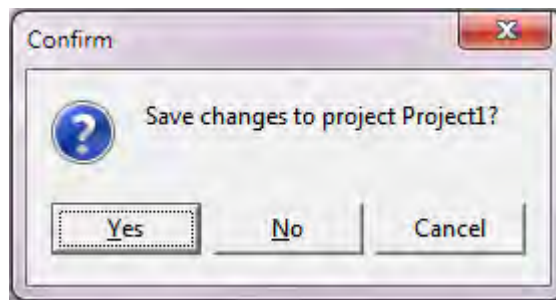


Рисунок 8 – Диалоговое окно сохранения изменений проекта

При повторном открытии файла с этой программой в ИСР командой **File | Open (Файл | Открыть...)** появится диалоговое окно, представленное на рисунке 9, где пользователю предлагается создать проект, чтобы открываемый файл можно было компилировать и выполнять. В этом окне нужно обязательно нажать кнопку **Yes (Да)**.

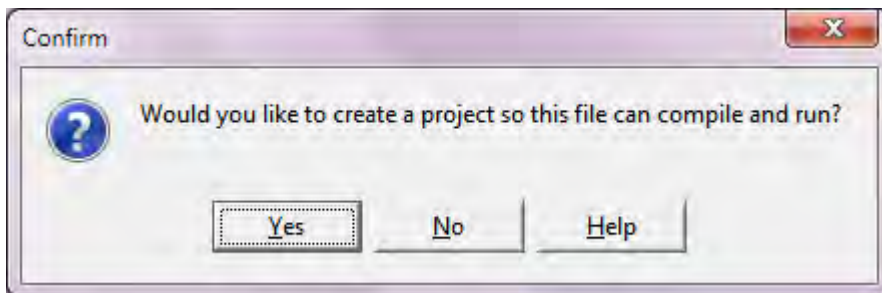


Рисунок 9 – Диалоговое окно, которое появляется при открытии файла

3 Основные возможности интегрированной среды разработки при работе с консольными приложениями

Для удобства работы программиста под строкой *главного меню* ИСР расположена *панель управления*, содержащая кнопки быстрого вызова, которые дублируют наиболее часто используемые команды главного меню. Эту панель можно настраивать под свои потребности, добавляя и удаляя кнопки. Под панелью находится *палитра компонентов* с множеством вкладок (рисунок 10).

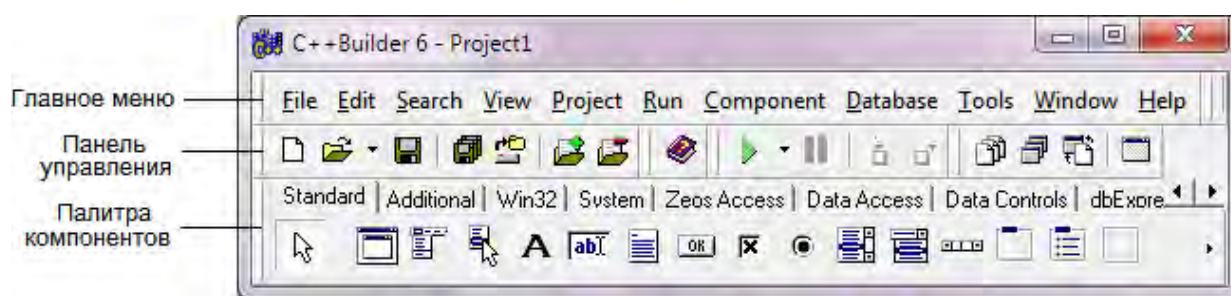

















Рисунок 10 – Элементы основного диалогового окна среды разработки

На панели управления, приведенной на рисунке 10, слева направо расположены кнопки:

- для добавления новых элементов  ;
- для открытия файла  с расположенным рядом списком недавно открытых файлов;
- для сохранения модуля  ;
- для сохранения всех файлов проекта  ;
- для открытия проекта  ;
- для добавления файлов в проект  ;
- для удаления файлов из проекта  ;
- для вызова справки  ;
- для запуска программы на выполнение  ;
- для остановки программы  ;
- две кнопки трассировки кода  с заходом и без захода внутрь функции;
- для переключения между несколькими программными модулями  , переключения между несколькими формами  , переключения между формой и модулем  и для добавления новой формы  .

Остальные кнопки панели управления и расположенная ниже палитра компонентов предназначены для визуальной разработки графического интерфейса приложений Windows.

Кратко опишем команды главного меню, которые могут быть полезны при разработке консольных приложений.

На рисунке 11 представлены команды меню **File (Файл)**.



Рисунок 11 – Команды меню **File (Файл)**

Команды этого меню осуществляют работу с файлами и предоставляют доступ к операциям создания (**New**) новых форм (окон) и приложений, открытия (**Open**), сохранения (**Save**) и закрытия (**Close**) файлов и проектов, печати текстов программ (**Print**) и добавления заголовков модулей (**Include**), т. е. файлов с расширением **.h**. Перед некоторыми командами этого и последующих меню находятся значки-иконки. Эти же значки изображены на кнопках быстрого запуска, что позволяет без труда понять их назначение. Справа от некоторых команд меню написано соответствие команд горячим клавишам, т. е. клавишам, с помощью которых вызывается данная команда. Команды, отмеченные справа треугольными стрелками, говорят о том, что данная команда имеет расширенные возможности, которые становятся доступными после наведения на эту стрелку курсора мышки.

На рисунке 12 представлены команды меню **Edit (Редактирование)**.

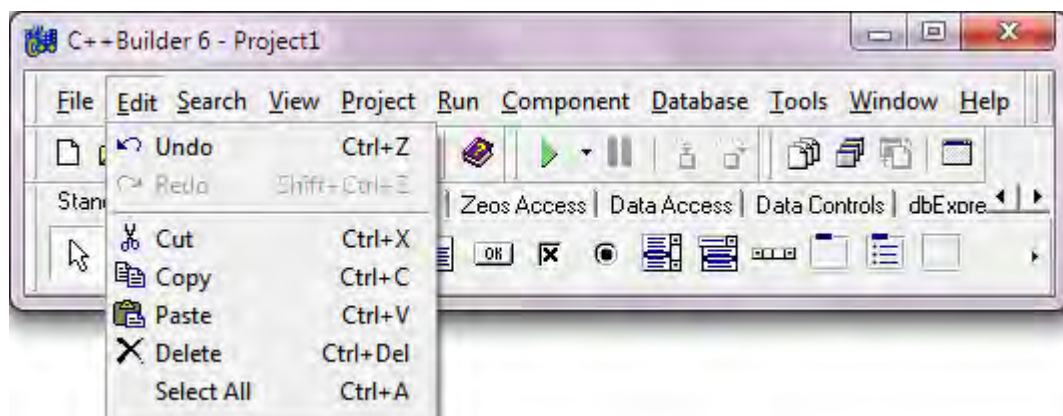


Рисунок 12 – Команды меню **Edit (Редактирование)**

В этом меню собраны команды редактирования, такие как отмена (**Undo**) и повторение (**Redo**) операций, вырезание (**Cut**), копирование (**Copy**), вставка (**Paste**) и удаление (**Delete**), команды выделения всего текста (**Select All**) и перехода к строке программного кода с заданным номером.

На рисунке 13 представлены команды меню **Search (Поиск)**.

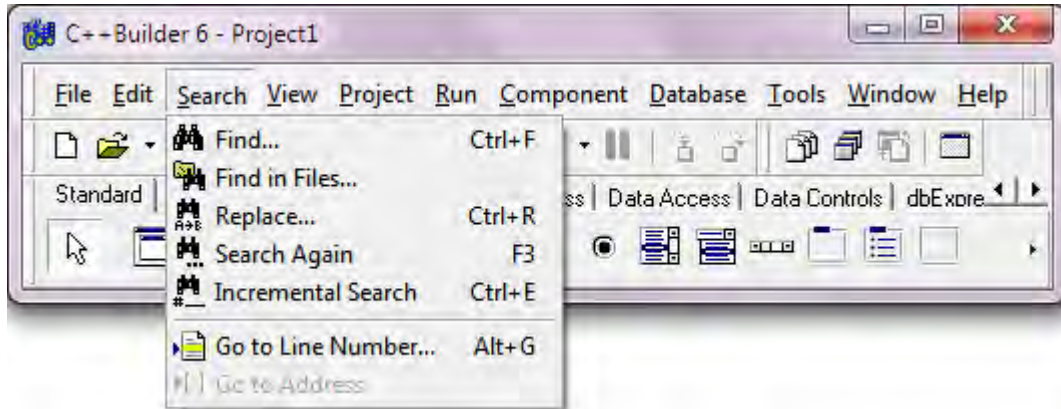


Рисунок 13 – Команды меню **Search (Поиск)**

Команды данного меню позволяют осуществить поиск текста в файле (**Find**), продолжить поиск после запуска программы, произвести автоматическую замену (**Replace**), быстро перейти к строке кода, задав ее номер (**Go to Line Number**).

Команды меню **View (Просмотр)** по большей своей части предназначены для разработки приложений с графическим интерфейсом.

Команда **Debug Windows (Окна отладки)** подробно разобрана в методических рекомендациях по отладке приложений.

Из меню **Project (Проект)** можно воспользоваться командами компиляции модуля (**Compile Unit**), компоновки и сборки проекта (**Make Project** и **Build Project**) (рисунок 14).

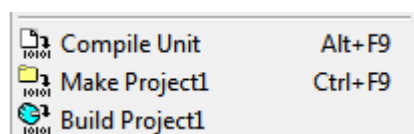


Рисунок 14 – Отдельные команды меню **Project (Проект)**

Команда **Run (Выполнить)** одноименного меню уже упоминалась ранее и была показана на рисунке 6. Остальные команды этого меню более подробно рассмотрены в методических рекомендациях по отладке приложений [2].

Команды меню **Window (Окно)** позволяют переключаться между различными программными модулями.

В меню **Help (Помощь)** включены команды, с помощью которых можно получить справку как о языке C++, так и о работе со средой разработки и со всеми ее компонентами.

4 Лабораторная работа № 1. Программа линейной структуры

Цель работы: ознакомиться с типами данных, основными операциями языка и функциями ввода-вывода, научиться использовать переменные и константы, а также функции ввода-вывода в консольных приложениях.

Задания на лабораторную работу

Перед выполнением лабораторной работы изучить тему 1 «Типы данных и операции языка. Форматированный ввод-вывод» из файла «Конспект лекций по ТиТП.docx».

Вариант 1

В магазин поступила партия костюмов в количестве N_1 единиц стоимостью S_1 р. за единицу и партия пальто в количестве N_2 единиц стоимостью S_2 р. за единицу. Торговая наценка на костюм $T_1 = 20\%$, торговая наценка на пальто $T_2 = 25\%$.

Вычислить общую выручку V и общую прибыль P магазина от реализации товаров. Наценки T_1 и T_2 объявить как константы.

Вариант 2

В магазин поступила партия товара в количестве N единиц стоимостью S р. за единицу. Торговая наценка T_1 на единицу товара составляла в текущем месяце 20% . В течение месяца было реализовано M единиц товара ($M < N$). В следующем месяце использовалась торговая наценка T_2 в размере 25% .

Вычислить выручку магазина от реализации товара за каждый месяц V_1 и V_2 . Наценки T_1 и T_2 объявить как константы.

Вариант 3

Себестоимость продукции, реализованной предприятием за месяц, составила S р., а выручка от реализации этой продукции составила V р. ($V > S$). Из суммы выручки предприятие должно уплатить следующие налоги:

- налог N_1 в размере $K_1\%$ от выручки;
- налог N_2 в размере $K_2\%$ от разности $V - N_1$;
- налог N_3 в размере $K_3\%$ от разности $V - N_1 - N_2$,

где $K_1 = 5$, $K_2 = 10$, $K_3 = 20$.

Вычислить: общую сумму налогов SN , которую нужно уплатить; прибыль P , остающуюся в распоряжении предприятия, как разность между выручкой, себестоимостью и общей суммой налогов. K_1 , K_2 , K_3 объявить как константы.

Контрольные вопросы

1 Перечислите требования к идентификаторам (допустимые символы, первый символ, длина идентификатора).

2 Что такое константа? Приведите примеры констант различных типов (логическая, целая десятичная, вещественная десятичная, вещественная экспоненциальная, символьная, строковая).



3 Покажите способы объявления и инициализации переменных и констант.

4 Что определяет тип данных? Перечислите базовые типы данных языка.

5 Какие ключевые слова уточняют внутреннее представление и диапазон значений базовых типов?

6 Какой объем памяти занимают основные типы данных?

7 Опишите общий формат функций ввода-вывода.

8 Что может содержать управляющая строка в функциях ввода-вывода?

9 Опишите общий вид спецификации преобразования (или спецификации формата).

10 Перечислите и опишите управляющие последовательности.

11 Опишите символы, которые используются для задания типа значения в спецификации формата.

12 Приведите примеры функций форматированного ввода-вывода.

13 Опишите синтаксис и приведите примеры использования операций и выражений (простейшие операции языка, приоритет операций, операции присваивания, операций инкремента ++ и декремента --, операции остатка от деления %, операции *sizeof*, операции приведения типа).

14 Приведите структуру программы на языке C.

15 Опишите и дайте характеристику директив препроцессора.

5 Лабораторная работа № 2. Программа с ветвлениями

Цель работы: изучить операторы ветвления, разработать программу с несколькими вычислительными ветвями.

Задания на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 2.1–2.4 темы 2 «Управляющие операторы» из файла «Конспект лекций по ТиТП.docx».

Вариант 1

Минимальная зарплата сотрудника составляет Z_{\min} р. в месяц. Оклад сотрудника равен Z р. в месяц. Подоходный налог N_p с оклада вычисляется следующим образом:

1) если оклад не превышает $10Z_{\min}$, то

$$N_p = Z \cdot 9 \%;$$

2) если оклад находится в пределах $10Z_{\min} < Z < 15Z_{\min}$, то

$$N_p = 10Z_{\min} \cdot 9 \% + (Z - 10Z_{\min}) \cdot 15 \%;$$

3) если оклад превышает $15Z_{\min}$, то

$$N_p = 10Z_{\min} \cdot 9 \% + 5Z_{\min} \cdot 15 \% + (Z - 15Z_{\min}) \cdot 20 \%.$$

Вычислить сумму подоходного налога N_p .



Вариант 2

Оклад сотрудника равен Z р. в месяц, а стаж работы составляет St лет. В случае заболевания в течение N дней больничный лист оплачивается работнику следующим образом: $Vb = N \cdot Z / Nm \cdot K \%$, где Nm – число дней в месяце. При этом:

- 1) если стаж работы меньше 7 лет, то $K = 50$;
- 2) если стаж находится в пределах $7 \leq St \leq 10$, то $K = 75$;
- 3) если стаж превышает 10 лет, то $K = 100$.

Вычислить сумму выплаты по больничному листу.

Вариант 3

Организация покупает со склада партию товара в количестве N единиц стоимостью Z р. за единицу. Этот товар затем будет продан с торговой наценкой T . Если размер партии N не меньше 1000 единиц, то организация при покупке товара получает скидку K в размере 10 % от общей стоимости товара, а торговая наценка T в этом случае будет равна 20 %. Если же размер партии меньше 1000 и не меньше 500 единиц, то организация при покупке товара получает скидку K в размере 5 % от общей стоимости товара, а торговая наценка T в этом случае будет равна 25 %. Если же размер партии меньше 500 единиц, то товар покупается за полную стоимость, а торговая наценка T в этом случае составляет 30 %.

Вычислить стоимость покупки партии товара S и выручку от его реализации V .

Контрольные вопросы

1 Перечислите операции отношения, используемые в языке, приведите примеры их использования.

2 Перечислите логические операции, используемые в языке, приведите примеры их использования.

3 Приведите таблицу истинности логических операций И, ИЛИ, ИСКЛЮЧАЮЩЕЕ ИЛИ, НЕ.

4 Как записать сложное логическое выражение и как определить результат его вычисления?

5 Опишите синтаксис и приведите пример операции ? (вопросительный знак).

6 Опишите синтаксис условного оператора *if*, приведите примеры его использования.

7 Опишите синтаксис условного оператора *if-else*, приведите примеры его использования.

8 Приведите примеры использования вложенных операторов *if* для организации нескольких вычислительных ветвей.

9 Опишите синтаксис оператора множественного выбора *switch* и приведите пример его использования.

10 Какой из операторов *if* или *switch* является более универсальным и почему?



6 Лабораторная работа № 3. Программа с циклами

Цель работы: изучить операторы цикла *for*, *while* и *do-while*, разработать программу с использованием операторов цикла.

Задания на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 2.5–2.10 темы 2 «Управляющие операторы» из файла «Конспект лекций по ТиТП.docx».

Вариант 1

Разработать программу с использованием операторов циклических операций (*while*, *do-while*), вычисляющую с заданной точностью ϵ значение функции $\sin(x)$. Исходные данные – значение аргумента функции и заданная точность вычислений. В качестве результата работы программы вывести на экран: значение аргумента, значение функции и значение, которое даёт библиотечная функция $\sin(x)$:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots + (-1)^{n-1} \frac{x^{2n-1}}{(2n-1)!} + \dots$$

Вариант 2

Разработать программу с использованием операторов циклических операций (*while*, *do-while*), вычисляющую с заданной точностью ϵ значение функции $\cos(x)$. Исходные данные – значение аргумента функции и заданная точность вычислений. В качестве результата работы программы вывести на экран: значение аргумента, значение функции и значение, которое даёт библиотечная функция $\cos(x)$:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \dots + (-1)^n \frac{x^{2n}}{(2n)!} + \dots$$

Вариант № 3

Разработать программу с использованием операторов циклических операций (*while*, *do-while*), вычисляющую с заданной точностью ϵ значение натурального логарифма. Исходные данные – значение аргумента функции и заданная точность вычислений. В качестве результата работы программы вывести на экран: значение аргумента, значение функции и значение, которое даёт библиотечная функция $\log(x)$:

$$\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} + \dots + (-1)^{n+1} \frac{x^n}{n} + \dots$$

Контрольные вопросы

- 1 Что необходимо для организации цикла?
- 2 Нарисуйте схему алгоритма, опишите синтаксис и приведите пример цикла с предусловием *while*.
- 3 Нарисуйте схему алгоритма, опишите синтаксис и приведите пример цикла с постусловием *do-while*.
- 4 Нарисуйте схему алгоритма, опишите синтаксис и приведите пример цикла *for*.
- 5 Какой из операторов цикла необходимо использовать в том или ином случае?
- 6 Нарисуйте схему алгоритма и приведите пример вложенных циклов.
- 7 Опишите синтаксис и приведите пример использования операторов передачи управления *break*, *continue*, *goto*.

7 Лабораторная работа № 4. Программа с одномерными массивами

Цель работы: изучить одномерные массивы, разработать программу с вводом, выводом и обработкой одномерных массивов.

Задания на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 3.1 темы 3 «Массивы» из файла «Конспект лекций по ТиТП.docx».

Вариант 1

Ввести с клавиатуры и вывести на экран значения элементов одномерного массива. Количество элементов массива N определяется пользователем. Исходный массив отсортировать в порядке возрастания и вывести на экран. Значения элементов массива, кратных некоторому заданному пользователем значению, заменить нулями.

Вариант 2

Ввести с клавиатуры и вывести на экран значения элементов одномерного массива. Количество элементов массива N определяется пользователем. Исходный массив отсортировать в порядке убывания и вывести на экран. Определить количество четных значений элементов массива.

Вариант 3

Ввести с клавиатуры и вывести на экран значения элементов одномерного массива. Количество элементов массива N определяется пользователем. Исходный массив отсортировать в порядке возрастания и вывести на экран. Нулевые значения элементов массива заменить на некоторое заданное пользователем значение.



Вариант 4

Ввести с клавиатуры и вывести на экран значения элементов одномерного массива. Количество элементов массива N определяется пользователем. Исходный массив отсортировать в порядке убывания и вывести на экран. Определить произведение нечетных значений элементов массива.

Вариант 5

Ввести с клавиатуры и вывести на экран значения элементов одномерного массива. Количество элементов массива N определяется пользователем. Исходный массив отсортировать в порядке возрастания и вывести на экран. Определить количество значений элементов массива, находящихся в интервале от $N1$ до $N2$.

Вариант 6

Ввести с клавиатуры и вывести на экран значения элементов одномерного массива. Количество элементов массива N определяется пользователем. Исходный массив отсортировать в порядке убывания и вывести на экран. Определить количество элементов массива, кратных заданному пользователем значению.

Контрольные вопросы

- 1 Дайте определение и приведите примеры объявления и инициализации одномерных массивов.
- 2 Приведите примеры ввода и вывода одномерных массивов.
- 3 Приведите алгоритм и пример программы накопления суммы и произведения элементов массива.
- 4 Приведите алгоритм и пример программы линейного поиска элемента в массиве.
- 5 Приведите алгоритм и пример программы поиска минимального элемента массива.
- 6 Приведите алгоритм и пример программы пузырьковой сортировки элементов массива.

8 Лабораторная работа № 5. Программа с многомерными массивами

Цель работы: изучить многомерные массивы, разработать программу с вводом, выводом и обработкой многомерных массивов.

Задания на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 3.2 темы 3 «Массивы» из файла «Конспект лекций по ТиТП.docx».



Вариант 1

Разработать программу, обеспечивающую поиск номера строки матрицы с минимальной суммой элементов этой строки, а также вычисление количества и суммы неотрицательных элементов матрицы.

В программе необходимо обеспечить ввод матрицы с клавиатуры и вывод ее на экран. Результатом работы программы является номер строки матрицы с минимальной суммой элементов и сумма элементов этой строки, а также количество и сумма неотрицательных элементов матрицы.

Вариант 2

Разработать программу, обеспечивающую поиск номера столбца матрицы с максимальным произведением элементов этого столбца, а также вычисление количества и произведения отрицательных элементов матрицы.

В программе необходимо обеспечить ввод матрицы с клавиатуры и вывод ее на экран. Результатом работы программы является номер столбца матрицы с максимальным произведением элементов и произведение элементов этого столбца, а также количество и произведение отрицательных элементов матрицы.

Контрольные вопросы

1 Дайте определение и приведите примеры объявления и инициализации многомерных массивов.

2 Приведите примеры ввода и вывода многомерных массивов.

3 Приведите пример линейного поиска в многомерном массиве.

4 Приведите примеры заголовков циклов для перемещения по матрице с любого угла в любом направлении, с заходом или без захода в главную диагональ.

5 Приведите примеры заголовков циклов для перестановки определенных строк и столбцов матрицы.

9 Лабораторная работа № 6. Программа с функциями

Цель работы: изучить механизмы работы функций, разработать программу с использованием функций.

Задание на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 4.1–4.2 темы 4 «Указатели и ссылки» и тему 5 «Функции» из файла «Конспект лекций по ТИП.docx».

В интервале от X_n до X_k с шагом H вычислить значение функции, оформив вычисление суммы степенного ряда в виде пользовательской функции (см. вариант задания для лабораторной работы № 3). Начальное, конечное значение аргумента X и шаг его изменения, а также заданную точность вычислений E ввести с клавиатуры. В качестве результатов работы программы



вывести: значения аргумента с заданным шагом, значения суммы ряда для каждого значения аргумента и значения, полученные с помощью библиотечной функции $\sin(x)$, $\cos(x)$ или $\log(x)$ для каждого значения аргумента.

Контрольные вопросы

- 1 Что такое функция?
- 2 Приведите синтаксис функций (объявление, определение, вызов).
- 3 Поясните разницу в вызове типизированных и нетипизированных функций.
- 4 Для чего предназначен оператор *return* и как правильно его использовать в типизированных и нетипизированных функциях?
- 5 Для глобальных, локальных и статических локальных переменных перечислите: место их объявления в программе, место хранения в оперативной памяти, время появления и исчезновения, а также область видимости в программе.
- 6 Опишите и приведите пример передачи параметров функциям по копии.
- 7 Опишите и приведите пример передачи параметров функциям по адресу.
- 8 Опишите и приведите пример передачи параметров функциям по ссылке.
- 9 Опишите и приведите пример передачи массивов функциям.
- 10 Приведите пример использования рекурсивной функции.
- 11 Приведите пример использования встроенной (*inline*) функции.
- 12 Приведите пример использования аргументов функции по умолчанию.
- 13 Приведите пример использования перегрузки функций.
- 14 Опишите неопределенности, возникающие при перегрузке функций и способы их устранения.

10 Лабораторная работа № 7. Отладка программ

Цель работы: изучить возможности знатока кода, механизмы отладки программ с использованием окна просмотра и точек прерывания.

Задание на лабораторную работу

Поскольку объем методических рекомендаций для выполнения этой лабораторной работы очень велик, то они вынесены в отдельный файл «Отладка программ.docx».

Контрольные вопросы

- 1 Опишите возможности знатока кода (*Code Insight*).
- 2 Для чего используются закладки (*Bookmarks*)?
- 3 Как установить (или снять) в редакторе кода точки прерывания?
- 4 Как можно просмотреть значения переменных во время выполнения, не прибегая к специальным средствам?



- 5 Как добавить переменные в окно просмотра (*Watch List*)?
- 6 Как настроить свойства отображения переменных в окне просмотра?
- 7 Как встроить окно просмотра в инспектор объектов?
- 8 Какие варианты трассировки программ используются в среде разработки и чем они отличаются?
- 9 Как установить условия на точки прерывания?
- 10 Как выполнить досрочное завершение работы программы?

11 Лабораторная работа № 8. Программа с текстовыми файлами

Цель работы: изучить способы работы с текстовыми файлами, разработать программу с вводом, выводом и обработкой текстовых файлов.

Задания на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 6.1–6.3 темы 6 «Файловый ввод-вывод» из файла «Конспект лекций по ТиТП.docx».

Вариант 1

Разработать программу, содержащую следующие функции:

- 1) ввод матрицы заданной размерности (количество столбцов чётное);
- 2) запись матрицы в текстовый файл;
- 3) чтение матрицы из текстового файла;
- 4) вывод матрицы на экран;
- 5) перестановка чётных и нечётных столбцов матрицы;
- 6) добавление обработанной матрицы в тот же файл.

Пользовательский интерфейс программы обеспечить с помощью меню.

Имя файла определяется пользователем.

Вариант 2

Разработать программу, содержащую следующие функции:

- 1) ввод матрицы заданной размерности (количество столбцов чётное);
- 2) запись матрицы в текстовый файл;
- 3) чтение матрицы из текстового файла;
- 4) вывод матрицы на экран;
- 5) «зеркальное отображение» матрицы по вертикали;
- 6) добавление обработанной матрицы в тот же файл.

Пользовательский интерфейс программы обеспечить с помощью меню.

Имя файла определяется пользователем.

Вариант 3

Разработать программу, содержащую следующие функции:

- 1) ввод матрицы заданной размерности (количество столбцов чётное);
- 2) запись матрицы в текстовый файл;



- 3) чтение матрицы из текстового файла;
- 4) вывод матрицы на экран;
- 5) перестановка чётных и нечётных строк матрицы;
- 6) добавление обработанной матрицы в тот же файл.

Пользовательский интерфейс программы обеспечить с помощью меню.

Имя файла определяется пользователем.

Вариант 4

Разработать программу, содержащую следующие функции:

- 1) ввод матрицы заданной размерности (количество столбцов чётное);
- 2) запись матрицы в текстовый файл;
- 3) чтение матрицы из текстового файла;
- 4) вывод матрицы на экран;
- 5) «зеркальное отображение» матрицы по горизонтали;
- 6) добавление обработанной матрицы в тот же файл.

Пользовательский интерфейс программы обеспечить с помощью меню.

Имя файла определяется пользователем.

Вариант 5

Разработать программу, содержащую следующие функции:

- 1) ввод матрицы заданной размерности (количество столбцов чётное);
- 2) запись матрицы в текстовый файл;
- 3) чтение матрицы из текстового файла;
- 4) вывод матрицы на экран;
- 5) перестановка левой и правой половин матрицы;
- 6) добавление обработанной матрицы в тот же файл.

Пользовательский интерфейс программы обеспечить с помощью меню.

Имя файла определяется пользователем.

Контрольные вопросы

- 1 Что такое потоки и файлы?
- 2 Перечислите и поясните основные функции файловой системы.
- 3 Что такое указатель файла (файловая переменная)?
- 4 Опишите синтаксис и приведите пример функции открытия файла.
- 5 Опишите основные режимы работы с файлами.
- 6 Опишите режимы редактирования при работе с файлами.
- 7 Опишите способы задания имени файла.
- 8 Опишите синтаксис и приведите пример функции закрытия файла.
- 9 Опишите функцию форматированного ввода для работы с текстовыми файлами.
- 10 Опишите функции форматированного вывода для работы с текстовыми файлами.
- 11 Опишите пример записи матрицы в текстовый файл.
- 12 Опишите пример чтения матрицы из текстового файла.



12 Лабораторная работа № 9. Программа с двоичными файлами

Цель работы: изучить способы работы с двоичными файлами, разработать программу с вводом, выводом и обработкой двоичных файлов.

Задание на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 6.4–6.6 темы 6 «Файловый ввод-вывод» из файла «Конспект лекций по ТиТП.docx».

Разработать программу, выполняющую запись, чтение и редактирование двоичного файла. Программа должна содержать следующие функции:

- 1) ввод одномерного массива;
- 2) запись массива в двоичный файл;
- 3) чтение массива из двоичного файла;
- 4) вывод массива на экран;
- 5) добавление нового элемента массива в файл;
- 6) изменение элемента массива в файле;
- 7) удаление элемента массива из файла.

Пользовательский интерфейс программы обеспечить с помощью меню.

Контрольные вопросы

- 1 Как создаются двоичные файлы?
- 2 Как создаются текстовые файлы?
- 3 Опишите синтаксис и приведите примеры использования функции блочного вывода *fwrite()*.
- 4 Опишите синтаксис и приведите примеры использования функции блочного ввода *fread()*.
- 5 Приведите варианты блочной записи одномерного массива в двоичный файл.
- 6 Приведите пример чтения одномерного массива из двоичного файла в случае, когда известен тип массива, но не известно количество элементов.
- 7 Опишите назначение, синтаксис и приведите пример использования функции *fseek()*.
- 8 Опишите назначение и особенности применения временных файлов.
- 9 Опишите функции редактирования двоичного файла в лабораторной работе.



13 Лабораторная работа № 10. Программа с динамическими массивами

Цель работы: изучить использование указателей, разработать программу с вводом, выводом и обработкой динамических массивов.

Задание на лабораторную работу

Перед выполнением лабораторной работы изучить подразд. 4.5 темы 4 «Указатели и ссылки» из файла «Конспект лекций по ТиТП.docx».

Разработать программу, которая создаёт динамическую матрицу, а также обеспечивает выполнение операций ввода, вывода и обработки матрицы согласно условиям лабораторной работы № 7. Передача матрицы функциям выполняется с помощью указателя на указатель.

Дополнить программу функциями создания динамической матрицы и удаления динамической матрицы.

Контрольные вопросы

- 1 Перечислите особенности динамических переменных.
- 2 Опишите синтаксис операции выделения динамической памяти *new*.
- 3 Опишите синтаксис операции освобождения динамической памяти *delete*.
- 4 Приведите пример создания, использования и удаления динамической переменной.
- 5 Приведите пример создания, использования и удаления динамического одномерного массива.
- 6 Приведите пример создания, использования и удаления динамической матрицы.

14 Лабораторная работа № 11. Программа со структурами

Цель работы: изучить структуры, разработать программу с вводом, выводом и обработкой структур.

Задание на лабораторную работу

Перед выполнением лабораторной работы изучить тему 7 «Пользовательские типы данных» из файла «Конспект лекций по ТиТП.docx».

Разработать программу, выполняющую анализ деятельности торгового предприятия. Информация для анализа должна быть представлена в виде массива структур, содержащих следующие поля: наименование товара и прибыль за каждый месяц года.

Анализ деятельности предприятия должен быть двух видов: по товарам и по месяцам. Анализ работы по товарам означает, что для выбранного товара



нужно определить минимальную и максимальную прибыль и месяц, когда она была получена. Анализ работы по месяцам означает, что для выбранного месяца нужно определить минимальную и максимальную прибыль и товар, для которого она была получена.

Пользовательский интерфейс обеспечить с помощью меню. Предусмотреть возможность ввода исходных данных как с клавиатуры, так и из двоичного (или текстового) файла.

Контрольные вопросы

- 1 Приведите определение и опишите синтаксис структуры.
- 2 Приведите пример объявления обычной и инициализированной структурной переменной.
- 3 Как осуществляется доступ к полям структурной переменной?
- 4 Что такое анонимные структуры?
- 5 Как выполняется присваивание структурных переменных?
- 6 Объявите массив структур и покажите, как заполнить его поля.
- 7 Как заполнить поля массива, который является полем структурной переменной?
- 8 Покажите пример объявления и использования вложенных структур.
- 9 Приведите пример объявления указателя на структуру.
- 10 Приведите примеры доступа к полям структурной переменной через указатель.
- 11 Опишите способы передачи информации о структурах функциям.
- 12 Для чего предназначен оператор *typedef*?
- 13 Как объявляются и используются перечисления?
- 14 Как объявляются и для чего используются битовые поля?

15 Лабораторная работа № 12. Программа с элементами управления: окно ввода, командная кнопка и надпись

Цель работы: получить навыки разработки приложений с графическим интерфейсом, включающим окна ввода, командные кнопки и надписи.

Задание на лабораторную работу

Перед выполнением лабораторной работы изучить разд. 1 и 2 из файла «Проектирование приложений с графическим интерфейсом.docx». Проектирование пользовательского интерфейса программы выполнить в соответствии с разд. 3 того же файла.

Разработать программу, выполняющую роль валютного калькулятора. При вводе суммы в долларах и текущего курса доллара результат получают в рублях, а при вводе суммы в рублях и текущего курса доллара результат получают в долларах.

Обеспечить обработку в программе следующих исключений: ввод нечисловых данных, деление на ноль и переполнение.



Контрольные вопросы

- 1 Опишите структуру и назначение палитры компонентов.
- 2 Опишите структуру и назначение формы.
- 3 Опишите структуру и назначение инспектора объектов, установку сложных в настройке свойств.
- 4 Опишите структуру и назначение редактора кода.
- 5 Опишите способы помещения, выделения и выравнивания компонентов на форме.
- 6 Опишите понятие «компонента».
- 7 Опишите структуру приложения: файл описания формы, файлы программных модулей, файл главного программного модуля, файл ресурсов, файл проекта.
- 8 Как генерируются методы обработки событий и как формируются их имена?
- 9 Как обратиться к свойствам объектов из программы?
- 10 Опишите использованные в лабораторной работе основные свойства и методы обработки событий компонентов *Edit, Button, Label*.

16 Лабораторная работа № 13. Программа с элементами управления: список и график

Цель работы: получить навыки разработки приложений с графическим интерфейсом, включающим списки и графики.

Задание на лабораторную работу

Проектирование пользовательского интерфейса программы выполнить в соответствии с разд. 4 файла «Проектирование приложений с графическим интерфейсом.docx».

Разработать программу, которая будет вычислять и выводить на форму в виде списка значения двух функций в заданном интервале значений аргумента. Кроме этого программа должна построить графики этих функций. Обеспечить возможность сохранения результатов вычислений в RTF-файл, копирование графика функций в буфер обмена и сохранение графика в BMP-файл.

Обеспечить обработку в программе следующих исключений: ввод нечисловых данных, деление на ноль и переполнение.

Контрольные вопросы

- 1 Опишите использованные в лабораторных работах основные свойства и методы обработки событий компонента-списка *ListBox*.
- 2 Опишите назначение компонента-списка *RichEdit* и возможности сохранения его содержимого в файле и заполнение его из файла.
- 3 Опишите назначение и настройку компонента для построения графиков *Chart*.



4 Опишите методы очистки и заполнения серий типа *Line* и заполнение заголовка графика из программы.

5 Опишите методы копирования графиков в файл и в буфер обмена.

17 Лабораторная работа № 14. Программа с элементами управления: многостраничный компонент, таблица, зависимый и независимый переключатель, группа, окна диалога

Цель работы: получить навыки разработки приложений с графическим интерфейсом, включающим многостраничный компонент, таблицу, зависимые и независимые переключатели, групповые компоненты, стандартные диалоговые окна сохранения и открытия файлов.

Задание на лабораторную работу

Проектирование пользовательского интерфейса программы выполнить в соответствии с разд. 5 файла «Проектирование приложений с графическим интерфейсом.docx».

Разработать программу для анализа деятельности торгового предприятия. Элементы управления расположить на закладках многостраничного компонента. Исходные данные для анализа вводятся на первой закладке в таблицу. Они могут быть сохранены в файле или прочитаны из файла с помощью стандартных диалоговых окон сохранения и открытия файлов.

На второй закладке с помощью зависимых переключателей, объединенных в группу, происходит выбор критериев анализа по кварталам, по отдельным товарам или по всем товарам. Для ввода номера квартала используется компонент «двуглавая стрелка», а выбор названия товара производится из выпадающего списка. После щелчка по кнопке *Расчет* результаты анализа (общая, средняя, минимальная и максимальная выручка) выводятся на вторую закладку. Отображением результатов управляют независимые переключатели, объединенные в группу *Показатели*. Данные, выбранные для анализа, отображаются в виде диаграммы, которая расположена на третьей закладке.

Контрольные вопросы

- 1 Опишите назначение, основные свойства и методы компонента *PageControl*.
- 2 Опишите назначение, основные свойства и методы компонента *StringGrid*.
- 3 Опишите Флаги свойства *Options* компонента *StringGrid*.
- 4 Опишите назначение, основные свойства и методы компонента *Bevel*.
- 5 Опишите назначение, основные свойства и методы компонентов *OpenDialog* и *SaveDialog*.
- 6 Опишите назначение, основные свойства и методы компонента *GroupBox*.
- 7 Опишите назначение, основные свойства и методы компонента *RadioButton*.
- 8 Опишите назначение, основные свойства и методы компонента *CheckBox*.



- 9 Опишите назначение, основные свойства и методы компонента *UpDown*.
 10 Опишите назначение, основные свойства и методы компонента *ComboBox*.
 11 Опишите алгоритм работы всех реализованных в лабораторной работе функций-обработчиков событий.

Список литературы

1 **Гагарина, Л. Г.** Введение в архитектуру программного обеспечения: учебное пособие / Л. Г. Гагарина, А. Р. Федоров, П. А. Федоров. – Москва: ФОРУМ : ИНФРА-М, 2017. – 320 с.

2 **Орлов, С. А.** Программная инженерия. Технологии разработки программного обеспечения : учебник / С. А. Орлов. – 5-е изд., обновл. и доп. – Санкт-Петербург: Питер, 2016. – 640 с.

3 **Павловская, Т. А.** С/С++. Процедурное и объектно-ориентированное программирование: учебник для вузов / Т. А. Павловская. – Санкт-Петербург: Питер, 2014. – 496 с.

4 **Гагарина Л. Г.** Технология разработки программного обеспечения : учебное пособие для вузов / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул; под ред. Л. Г. Гагариной – Москва: ФОРУМ: ИНФРА-М, 2012. – 400 с.

5 **Хорев, П. Б.** Объектно-ориентированное программирование: учебное пособие / П. Б. Хорев. – 4-е изд. – Москва: Академия, 2012. – 448 с.

6 **Павловская, Т. А.** С/С++. Программирование на языке высокого уровня: учебник для вузов / Т. А. Павловская. – Санкт-Петербург: Питер, 2011. – 464 с.

7 **Иванова, Г. С.** Технология программирования: учебник для вузов / Г. С. Иванова. – Москва: Кнорус, 2011. – 336 с.

8 Информатика (общий курс) : учебник для вузов / А. Н. Гуда [и др.] ; под ред. В. И. Колесникова. – 3-е изд. – Москва; Ростов-на-Дону : Дашков и К : Наука-Спектр, 2009. – 400 с.

9 **Культин, Н. Б.** С/С+ в задачах и примерах / Н. Б. Культин. – Санкт-Петербург: БХВ-Петербург, 2002. – 288 с.

10 **Шилдт, Г.** Теория и практика С++: пер. с англ / Г. Шилдт. – Санкт-Петербург: ВHV, 2001. – 416 с.

11 **Дейл, Н.** Программирование на С++ / Н. Дейл, Ч. Уимз, М. Хедингтон. – Москва: ДМК, 2000. – 672 с.: ил.

