

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Оренбургский государственный университет»

Кафедра прикладной информатики в экономике и управлении

С.А. Вдович

ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ ПО ДИСЦИПЛИНЕ «ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ»

Методические указания

Рекомендовано к изданию редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего образования «Оренбургский государственный университет» для обучающихся по образовательной программе высшего образования по направлению подготовки 38.03.05 Бизнес-информатика

Оренбург
2018

УДК 004.43
ББК 32.973.3
В 27

Рецензент – доцент, кандидат экономических наук Т.В. Омельченко

- Вдович, С.А.**
В27 Выполнение курсовой работы по дисциплине «Объектно-ориентированное программирование»: методические указания / С.А. Вдович; Оренбургский гос. ун-т. – Оренбург: ОГУ, 2018. – 36 с.

Методические указания предназначены для выполнения курсовой работы по дисциплине «Объектно-ориентированное программирование». Методические указания содержат краткие сведения по основам объектно-ориентированного программирования в инструментальной среде разработки C++ Builder, описание этапов выполнения курсовой работы, а также практический пример реализации курсовой работы, список вариантов заданий на курсовую работу.

Методические указания по дисциплине «Объектно-ориентированное программирование» предназначены для обучающихся по направлению подготовки 38.03.05 Бизнес-информатика очной и заочной форм обучения.

УДК 004.43
ББК 32.973.3

©Вдович С.А., 2018
© ОГУ, 2018

Содержание

Введение	4
1 Теоретическая часть	5
2 Пример выполнения курсовой работы.....	8
2.1 Постановка задачи.....	8
2.2 Практическая часть	9
3 Оформление курсовой работы.....	19
4 Варианты заданий	20
Список использованных источников	36

Введение

Дисциплина «Объектно-ориентированное программирование» относится к базовой части блока 1 «Дисциплины (модули)», предусмотренных Государственным стандартом подготовки бакалавров по направлению 38.03.05 «Бизнес-информатика».

Целью освоения дисциплины является формирование у обучающихся теоретических знаний и практических навыков по разработке программного обеспечения (ПО) для решения экономических и расчетных задач с применением современных методов и технологий программирования.

Задачами курса являются изучение принципов объектно-ориентированного подхода к разработке программного обеспечения; приобретение навыков визуальной разработки приложений для решения экономических задач; овладение опытом создания программ с развитым графическим интерфейсом.

В рамках курса обучающиеся должны получить навыки использования базовых понятий и определений объектно-ориентированного программирования, разработки программного обеспечения с применением современных методов и технологий программирования.

Данные методические указания предназначены для выполнения курсовой работы по дисциплине «Объектно-ориентированное программирование» для обучающихся вторых курсов направления подготовки 38.03.05 «Бизнес-информатика». Методические указания содержат теоретический материал по основам объектно-ориентированного программирования на языке С++ и описание примера выполнения курсовой работы, а также описание требований по оформлению и содержанию пояснительной записки.

1 Теоретическая часть

Одной из технологий программирования является технология объектно-ориентированного программирования (ООП), которая основана на моделировании реального мира, при котором детали его реализации скрыты. ООП основано на трех концепциях: инкапсуляции, наследовании, полиморфизме. Эти принципы реализованы в программных языках высокого уровня, таких как С, С++, Паскаль, Java. Основными понятиями ООП являются понятия класса и объекта. Класс представляет собой тип, создаваемый пользователем. Объект или экземпляр класса является переменной некоторого типа, определенного пользователем. Таким образом, класс это объектно-ориентированный инструмент для создания новых типов данных, являющихся объектами.

Существует множество инструментальных средств разработки программ, ориентированных на определенный программный язык. Для объектно-ориентированного программирования на С++ можно выделить наиболее популярные среды разработки Visual Studio С++ и Borland С++ Builder.

С++ Builder - программный продукт, инструмент быстрой разработки приложений (RAD), система, используемая программистами для разработки программного обеспечения на языке С++. Программный продукт принадлежит компании Embarcadero Technologies. Borland С++ Builder объединяет в себе комплекс объектных библиотек, таких как STL, VCL, CLX, MFC и др., компилятор, отладчик, редактор кода и многие другие компоненты.

Главное окно среды программирования состоит из главного меню, панели инструментов, палитры компонент (библиотека VCL). Палитра компонент оперирует визуальными элементами управления и содержит библиотеку из более 100 компонент.

Основой всех приложений Borland С++ Builder является форма. Ее можно понимать как типичное окно Windows. Форма является основой, на которой размещаются другие компоненты.

Одной из наиболее важных частей среды Borland C++ Builder является окно Редактора Кода, или Текстового редактора. Он предназначен для размещения текстов программных единиц проектов с целью их создания, подключения к проекту, отладки, изменения (редактирования) или переименования.

При создании новых программных единиц Borland C++ Builder формирует заготовки текстов этих программных единиц и помещает их в отдельные страницы активного окна редактора, а также устанавливает необходимые связи между компонентами проекта. Со своей стороны программист дополняет созданные заготовки конкретным содержанием.

Стандартная библиотека шаблонов STL (Standard Template Library, STL) включает четыре основных элемента: контейнеры, итераторы, алгоритмы, функциональные объекты. Контейнеры предназначены для хранения других объектов. Различают такие типы контейнеров как массивы (vector), очереди (queue, priority_queue—очередь с приоритетом), списки (deque— двунаправленный список, list— линейный список, map—ассоциативный список для хранения пар ключ/значение), стек (stack), множество (set). Итераторы предоставляют методы доступа к контейнерам. Каждый итератор имеет один конструктор, функцию Restart – устанавливает итератор на первый объект, перегруженные операторы int и ++. Алгоритмы предоставляют операции для обработки контейнеров. Для использования алгоритмов в программе необходимо подключить заголовочный файл <algorithm>.

Визуальное программирование предполагает создание программ с помощью графических объектов. Программисту не нужно писать программу «с нуля», а нужно лишь добавить необходимые объекты на форму, настроить их свойства в инспекторе объектов, выбрать события, которые необходимо обработать и приложение автоматически создаст заготовки процедур и функций, выбранных событий. Разработчику останется лишь дописать или изменить программный код в созданных процедурах. Библиотека визуальных компонент VCL ИСР Borland C++ Builder состоит из более чем 100 компонент. Современные средства программирования позволяют использовать OLE-технологии. OLE-технология,

технология, позволяющая создавать сложные составные документы, в которых содержатся разнотипные объекты, созданные различными приложениями. Так в таблицу StringGrid можно вставить таблицу Excel. Для использования OLE-технологии в библиотеке VCL имеется компонент OleContainer, который реализует в программе механизм внедрения и связывания объектов OLE. С его помощью можно передавать данные между различными программами в среде Windows. Компонент OleContainer характеризуется следующими свойствами:

- OleClassName – имя класса объекта;
- AutoActivate – способ активации объекта, может принимать значения: aaManual – программным путем;
- aaGetFocus – при получении фокуса; aaDoubleClick – при двойном щелчке левой кнопкой мыши;
- State – состояние объекта;
- Iconic – отображение объекта в виде иконки.

Так же в среде разработки Borland C++ Builder имеются компоненты для открытия и сохранения файлов: OpenFileDialog и SaveDialog.

Таким образом, используя возможности ИСР и современные методы программирования можно разработать приложение для решения экономических задач.

2 Пример выполнения курсовой работы

В качестве примера предлагается рассмотреть решение задачи определения рентабельности предприятий.

2.1 Постановка задачи

Дана структура данных: "Рентабельность предприятий".

Таблица 1 – Пример исходных данных

Предприятие	Доход, млн. руб.	Расход, млн. руб.	Прибыль, млн. руб.	Рентабельность, %
ОАО "Лакокраска"	152,240	135,250	= (вычисляется)	= (вычисляется)
...				

Примечание: $\text{Прибыль} = \text{Доход} - \text{Расход}$; $\text{Рентабельность} = 100 * \frac{\text{Прибыль}}{\text{Расход}}$

Обработка:

- Вычислить среднюю рентабельность по всем предприятиям.
- Отобразить данные на убыточные предприятия (рентабельность меньше 100%).
- Отсортировать данные по убыванию значений в колонке "Рентабельность".
- Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Необходимо разработать Windows-приложение в среде C++ Builder, которое позволяет:

- создать структуру данных;
- отобразить созданную структуру данных в таблице (компонент StringGrid);
- редактировать данные;
- сохранить данные в файл на диск при помощи компонента SaveDialog;

- открыть с диска файл с данными при помощи компонента OpenDialog;
- подготовить файл с данными для проверки работы программы;
- выбрать нужное пользователю действие через главное меню программы, спроектированное при помощи компонента MainMenu;
- обработать данные.

Алгоритм и сценарий работы программы можно описать следующим образом:

- 1) Ввод данных о номере, наименовании предприятия, доходах и расходах;
- 2) Вычисление прибыли и рентабельности предприятия;
- 3) Вывод данных об убыточных предприятиях;
- 4) Сортировка по возрастанию значений рентабельности предприятий и №, п/п;
- 5) Сохранение информации в файл.

Алгоритм работы программы приведен на рисунке 1.

2.2 Практическая часть

2.2.1 Проектирование интерфейса пользователя

Стандартное консольное приложение является простейшим типом приложений, создаваемых средой C++ Builder. Чтобы создать в среде C++ Builder консольное приложение необходимо выполнить следующие действия:

1. Запустить Borland C++ Builder.
2. Открыть новое приложение, выполнив команду File/NewApplication.

Проектирование интерфейса пользователя начинается на главной форме Form1. На форму необходимо добавить такие компоненты как: MainMenu, StringGrid, OpenDialog, SaveDialog, Edit.

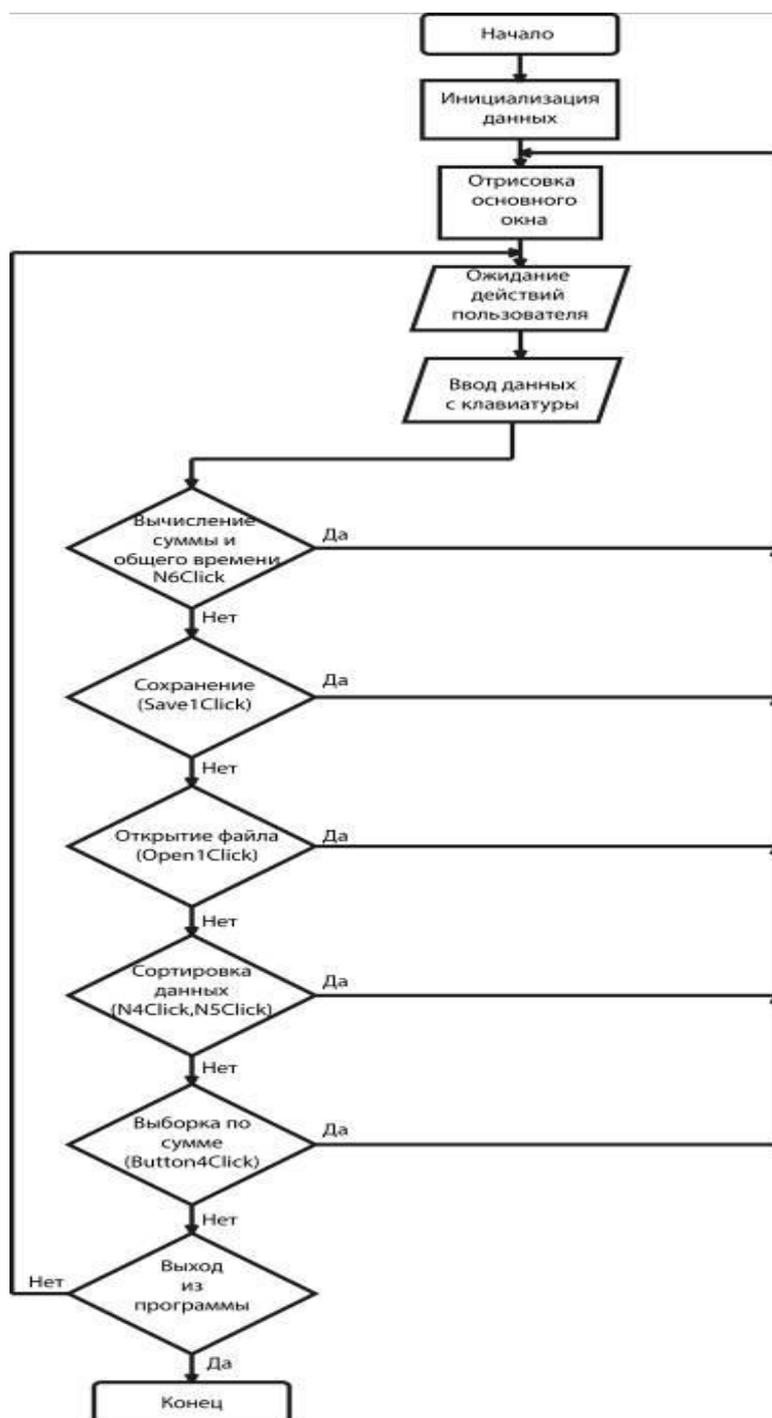


Рисунок 1 - Алгоритм работы программы

Описание компонент:

1 MainMenu1 - это главное меню программы, с помощью конструктора необходимо создать следующие пункты:

- Раздел «Вычисление» включает в себя: «Прибыль» {N2}, «Рентабельность» {N9}, «Средняя рентабельность» {N10}.

– Раздел «Сортировка» включает в себя: «по убыванию» {N4}, «по № п/п» {N5}.

– Раздел «Данные» включает в себя: «Структура данных» {N7}, «Убыточные предприятия» {N8}.

– Раздел «Выход» выполняет выход из программы.

2 StringGrid1- основная таблица для отображения данных.

3 StringGrid2- таблица для отображения убыточных предприятий.

4 Edit1- это поле для вывода результата вычисления средней рентабельности.

5 OpenFileDialog1- окно диалога для открытия файла.

6 SaveDialog1- окно диалога для сохранения файла.

Для добавленных компонент настроить свойства в Инспекторе объектов.

Свойства представлены в таблице 2.

Таблица 2 – Свойства компонент библиотеки VCL

Компонент	Свойство	Значение
StringGrid	Options: ColCount, RowCount FixedCols, FixedRows	Параметры настройки объекта. Для изменения данных в таблице. Изменение количества строк и столбцов в таблице. Фиксированные строки и столбцы в таблице.
Edit	ReadOnly	Установка этого свойства в true не дает пользователю изменять содержимое Edit.
MainMenu	Caption	Отвечает за название пункта в меню.

Таким образом, была спроектирована главная форма Form1, представлена на рисунке 2.

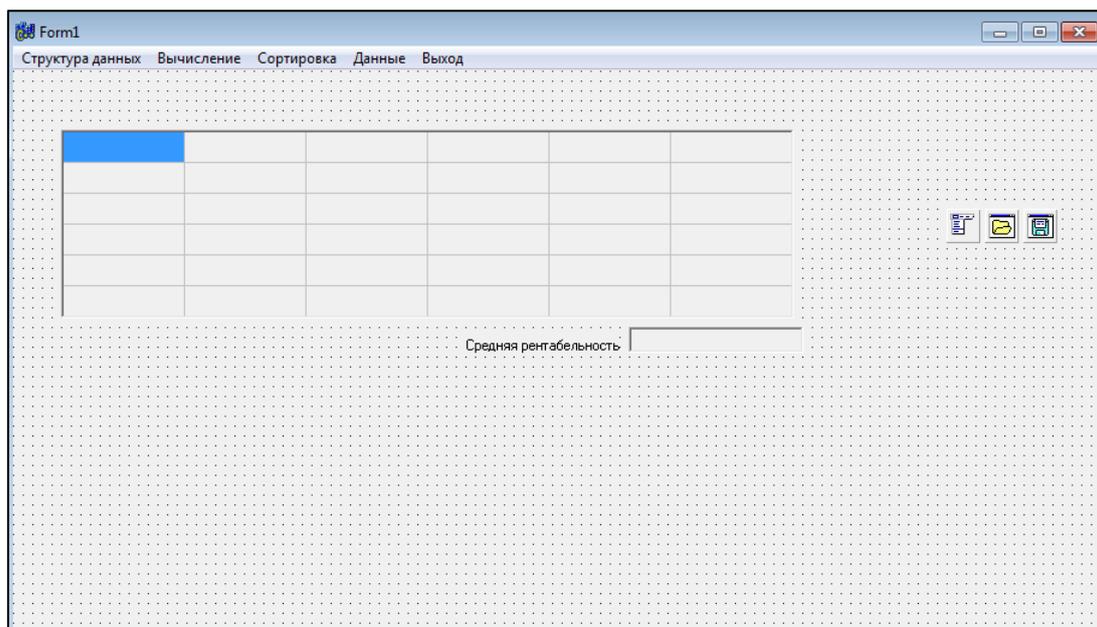


Рисунок 2 - форма Form1

2.2.2 Написание программных модулей

1. Сохранение во внешний файл из таблицы

Использование компонента `SaveDialog` является необходимым условием для функционирования программы. С помощью данного компонента данные импортируются в файл `MSExcel`.

Листинг обработчика события для открытия файла:

```
void __fastcall TForm2::N2Click(TObject *Sender)
{ WideString;
  TStringList *sl = new TStringList;
  for(int i = 0 ; i<StringGrid1->RowCount ;++i)
  { for(int j = 0; j< StringGrid1->ColCount;++j)
    str +=StringGrid1->Cells[j][i]+ "\t";
    sl->Add(str);
    str = "\r" ;
    str = "";}
  if (SaveDialog1->Execute())
  sl->SaveToFile(SaveDialog1->FileName+ ".xls");
  deletesl;}
```

2. Закрытие программы

При нажатии пункта меню «Выход» основного меню программы, экранная форма закрывается. Программный модуль данного действия:

```
void __fastcall TForm2::N3Click(TObject *Sender)
{Form2->Close();}
```

3. Расчет прибыли осуществляется на основании следующего программного кода:

```
void __fastcall TForm2::N5Click(TObject *Sender)
{for(int j=1; j<StringGrid1->RowCount;j++)
StringGrid1->Cells[4][j] = FloatToStr(StrToFloat(StringGrid1->Cells[2][j])-
StrToFloat(StringGrid1->Cells[3][j]));}
```

4. Расчет рентабельности

Листинг программного кода расчета рентабельности:

```
void __fastcall TForm2::N6Click(TObject *Sender)
{for(int j=1;j<StringGrid1->RowCount;j++)
StringGrid1->Cells[5][j]=FloatToStr((StrToFloat(StringGrid1->Cells[4][j]) /
StrToFloat(StringGrid1->Cells[3][j]))*100);}
```

5. Отображение данных по убыточным предприятиям:

```
void __fastcall TForm2::N7Click(TObject *Sender)
{float i,j;
for (i=1;j<StringGrid1->ColCount;i++)
for(j=1;j<StringGrid1->RowCount;j++)
if (StrToFloat(StringGrid1->Cells[5][j])<100)
ListBox1->Items->Add(StringGrid1->Cells[5][j]);}
```

6. Сортировка по порядковому номеру:

```

void __fastcall TForm2::N9Click(TObject *Sender)
{
    TStringList *SL = new TStringList;
    AnsiString S;
    for (inti=0; i< StringGrid1->RowCount; i++)
    {
        S="";
        for (int j=0; j < StringGrid1->ColCount; j++) S += StringGrid1->Cells[j][i]+" ";
        SL->Add(S);}
    SL->Sort();
    for (inti=0; i< StringGrid1->RowCount; i++)
    {
        StringGrid1->Rows[i]->DelimitedText = SL->Strings[i];}
    deleteSL;}

```

7.Сортировка по возрастанию значений процента рентабельности

```

{int j;
String temp;
for (int j = 1; j < StringGrid1->RowCount; j++)
if(StrToFloat( StringGrid1->Cells[5][j]) >StrToFloat( StringGrid1->Cells[5][j+1] ))
{for(int k=0;k<StringGrid1->ColCount;k++)
{ temp=StringGrid1->Cells[k][j];
StringGrid1->Cells[k][j]= StringGrid1->Cells[k][j+1];
StringGrid1->Cells[k][j+1]=temp;}}}

```

8.Добавлениестрокиивтаблицу

```

void __fastcall TForm1::Button1Click(TObject *Sender)
{
    introw_count = StringGrid1 ->RowCount;
    for (inti = 1;i<row_count-1;i++)
        StringGrid1 -> Rows[i] = StringGrid1 ->Rows[i + 1];
    StringGrid1 ->RowCount = row_count+1;}

```

9. Удалениестрокиизтаблицы

```

void __fastcall TForm1::Button2Click(TObject *Sender)
{
    int row_count = StringGrid1 ->RowCount;
    for (inti = 1;i<row_count-1;i++)
        StringGrid1 -> Rows[i] = StringGrid1 ->Rows[i + 1];
    StringGrid1 ->RowCount = row_count-1;}

```

10. Очищение данных из таблицы

```

void __fastcall TForm1::Button3Click(TObject *Sender)
{
    for(inti=0;i<StringGrid1->ColCount;i++) {
    for(int j=1; j<StringGrid1->RowCount;j++) {
    StringGrid1->Cells[i][j] = "";} }

```

2.2.3 Результаты тестирования программы

Первоначально необходимо заполнить исходные данные по предприятиям. Эту информацию можно загрузить из файла MSExcel. Для этого необходимо в главном меню выбрать пункт «Структура данных».

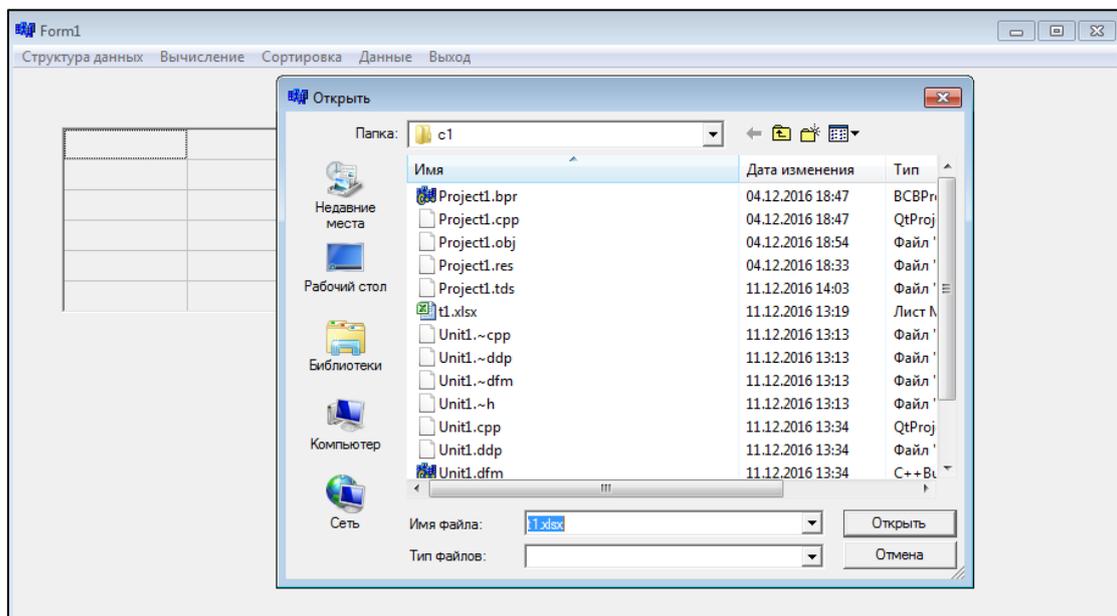


Рисунок 2 - Импорт данных в программу

Загруженные данные представлены на рисунке 3. Пустые столбцы будут рассчитаны.

№ п/п	Предприятие	Доход, млн. руб.	Расход, млн. руб.	Прибыль, млн. руб.	Рентабельность, %
1	ОАО "Лакокраска"	152,24	115,25		
2	ОАО "Лента"	252,76	120,39		
3	ГК "Рольф"	176,66	106,87		
4	ОАО "Виктория"	138,12	100,43		
5	ООО "ИТЕРА"	210,47	102,88		

Средняя рентабельность

Рисунок 3 - Исходная таблица.

Далее необходимо вычислить значение прибыли и рентабельности для каждого предприятия. Для вычисления прибыли предприятию необходимо воспользоваться пунктом в главном меню «Вычисление» / «Прибыль». Результат вычисления представлен на рисунке 4.

№ п/п	Предприятие	Доход, млн. руб.	Расход, млн. руб.	Прибыль, млн. руб.	Рентабельность, %
1	ОАО "Лакокраска"	152,24	115,25	36,99	
2	ОАО "Лента"	252,76	120,39	132,37	
3	ГК "Рольф"	176,66	106,87	69,79	
4	ОАО "Виктория"	138,12	100,43	37,69	
5	ООО "ИТЕРА"	210,47	102,88	107,59	

Средняя рентабельность

Рисунок 4 - Экранная форма результата вычисления прибыли предприятий

Далее можно вычислить рентабельность предприятий, выбрав пункт главного меню «Вычисление»/ «Рентабельность». Результат вычисления представлен на рисунке 5.

№ п/п	Предприятие	Доход, млн. руб.	Расход, млн. руб.	Прибыль, млн. руб.	Рентабельность, %
1	ОАО "Лакокраска"	152,24	115,25	36,99	32,0954446854664
2	ОАО "Лента"	252,76	120,39	132,37	109,950992607359
3	ГК "Рольф"	176,66	106,87	69,79	65,3036399363713
4	ОАО "Виктория"	138,12	100,43	37,69	37,5286269043115
5	ООО "ИТЕРА"	210,47	102,88	107,59	104,578149300156

Средняя рентабельность

Рисунок 5 - экранная форма результата вычислений рентабельности предприятий

Для определения средней рентабельности всех предприятий, необходимо воспользоваться пунктом главного меню «Вычисление»/ «Средняя рентабельность».

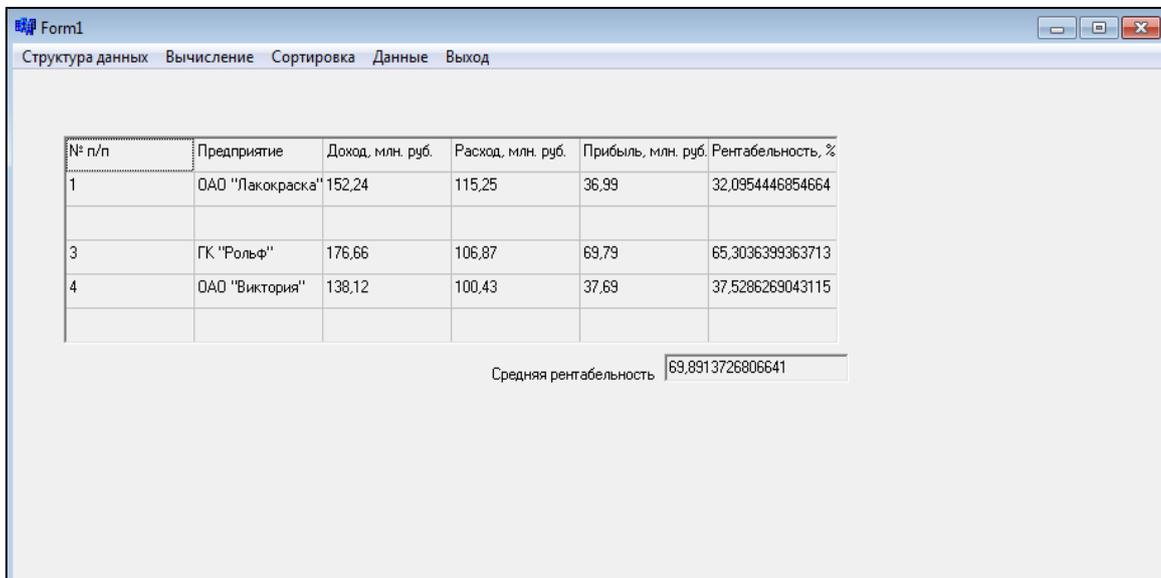
Для выполнения сортировки данных необходимо выбрать пункт меню «Сортировка» / По убыванию. Результаты работы программы представлены на рисунке 6.

№ п/п	Предприятие	Доход, млн. руб.	Расход, млн. руб.	Прибыль, млн. руб.	Рентабельность, %
2	ОАО "Лента"	252,76	120,39	132,37	109,950992607359
5	ООО "ИТЕРА"	210,47	102,88	107,59	104,578149300156
3	ГК "Рольф"	176,66	106,87	69,79	65,3036399363713
4	ОАО "Виктория"	138,12	100,43	37,69	37,5286269043115
1	ОАО "Лакокраска"	152,24	115,25	36,99	32,0954446854664

Средняя рентабельность 69,8913726806641

Рисунок 6 - Сортировка предприятий по убыванию значения рентабельности

Для возврата к исходным данным: пункт главного меню «Сортировка»/ «по № п/п». Для отображения убыточных предприятий пункт главного меню: «Данные» / «Убыточные предприятия». Эти данные представлены на рисунке 7.



The screenshot shows a window titled 'Form1' with a menu bar containing 'Структура данных', 'Вычисление', 'Сортировка', 'Данные', and 'Выход'. The main area contains a table with the following data:

№ п/п	Предприятие	Доход, млн. руб.	Расход, млн. руб.	Прибыль, млн. руб.	Рентабельность, %
1	ОАО "Лакокраска"	152,24	115,25	36,99	32,0954446854664
3	ГК "Рольф"	176,66	106,87	69,79	65,3036399363713
4	ОАО "Виктория"	138,12	100,43	37,69	37,5286269043115
Средняя рентабельность					69,8913726806641

Рисунок 7 - Экранная форма по убыточным предприятиям

Для выхода из программы необходимо воспользоваться пунктом главного меню «Выход»

3 Оформление курсовой работы

Курсовая работа оформляется в соответствии с общими требованиями и правилами оформления студенческих работ «[СТО 02069024.101–2015 РАБОТЫ СТУДЕНЧЕСКИЕ. Общие требования и правила оформления](#)».

Структура курсовой работы:

- титульный лист;
- лист задания;
- аннотация (краткая справка о курсовой работе).
- содержание (должно строиться средствами редактора MS Word).
- введение (содержит цель выполняемой работы, описание используемых средств).
- 1 Теоретическая часть, в которую входят подразделы:
 - 1.1 Постановка задачи (содержит формулировку задания, описание исходных данных и результатов выполнения программы).
 - 1.2 Основные аспекты обработки структур и файлов в среде C++ Builder.
 - 1.3 Алгоритм и сценарий работы программы.
- 2 Практическая часть, в которую входят подразделы:
 - 2.1 Проектирование интерфейса пользователя. (Описание созданных форм и установленных на них компонентов).
 - 2.2 Написание программных модулей. (Описание всех созданных функций и обработчиков событий).
 - 2.3 Результаты тестирования программы. (Описание и скриншоты тестовых исходных данных и полученных результатов их обработки).
- Заключение (содержит краткие выводы по результатам курсовой работы).
- Список используемых источников.
- Приложение (содержит полный текст программных модулей).

4 Варианты заданий

Общая часть задания:

Разработать Windows-приложение в среде C++ Builder, которое позволяет:

- создать структуру данных, определенную в вашем варианте;
- отобразить созданную структуру данных в таблице (компонент StringGrid);
- редактировать данные;
- сохранить данные в файл на диск при помощи компонента SaveDialog;
- открыть с диска файл с данными при помощи компонента OpenDialog;
- подготовить файл с данными для проверки работы программы;
- выбрать нужное пользователю действие через главное меню программы, спроектированное при помощи компонента MainMenu;
- обработать данные (в чем заключается обработка, определено в вашем варианте).

Варианты индивидуальной части задания:

(Номер варианта определяется номером в алфавитном списке учебного журнала группы).

Вариант 1.

Структура данных: "Заказ стройматериалов".

№ п/п	Материал	Ед.измерения	Количество	Цена ед., руб.	Сумма, руб.
1	Краска	л	3	95000	= (вычисляется)
...	...				

Обработка:

Вычислить общую сумму заказа.

Отобразить данные, у которых сумма заказа больше заданного значения.

Отсортировать данные по убыванию цены за ед.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 2.

Структура данных: "Инвентаризационная ведомость".

№ п/п	Наименование	Инвентарный номер	Количество	Цена ед., \$.	Сумма, руб.
1	ПЭВМ "ICS"- PIV/2400M	312000345	1	780	= (вычисляется)
2	"Мышь" Genius-PS\2	-	12	5.2	...

Обработка:

Вычислить общую сумму по всей ведомости.

Отобразить только те данные, у которых имеется инвентарный номер и количество равно 1.

Отсортировать данные по убыванию суммы.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 3.

Структура данных: "Квитанция об оплате коммунальных услуг".

№№ п/п	Наименование	Ед. изм.	Количество	Тариф за ед., руб.	Сумма, руб.
1	Отопление	кв. м.	58	1500	= (вычисляется)

Обработка:

Вычислить общую сумму оплаты по квитанции.

Отобразить данные, у которых сумма оплаты меньше заданного значения.

Отсортировать данные по возрастанию суммы.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 4.

Структура данных: "Товарная накладная".

№ п/п	Товар	Ед. изм.	Количество	Цена ед., руб.	Сумма, руб.
1	Печенье "Василёк"	кг	32	24000	= (вычисляется)
...					

Обработка:

Вычислить общую сумму по накладной.

Отобразить только те данные, у которых сумма больше заданного значения.

Отсортировать данные по возрастанию цены за ед.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 5.

Структура данных: "Расход бензина по автопарку".

№ п/п	Номер автомобиля	Марка	Расход, л на 100 км	Пробег, км	Расход, л
1	15-45 ГКЛ	ГАЗ-53	9	420	= (вычисляется)
...	...				

Обработка:

Вычислить общий расход бензина по автопарку.

Отобразить данные на автомобили, заданной марки.

Отсортировать данные по убыванию значений в колонке "Расход".

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 6.

Структура данных: "Рентабельность предприятий".

Предприятие	Доход, млн. руб.	Расход, млн. руб.	Прибыль, млн. руб.	Рентабельность, %
ОАО "Лакокраска"	152,240	135,250	= (вычисляется)	= (вычисляется)
...				

Примечание: Прибыль = Доход - Расход; Рентабельность = $100 \cdot \frac{\text{Прибыль}}{\text{Расход}}$ *

Прибыль/Расход

Обработка:

Вычислить среднюю рентабельность по всем предприятиям.

Отобразить данные на убыточные предприятия (рентабельность меньше 100%).

Отсортировать данные по убыванию значений в колонке "Рентабельность".

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 7.

Структура данных: "Ведомость начисления стипендии".

№ п/п	Ф.И.О.	Коэфф. успеваемости	Начислено	Проф. взнос (1.5%)	К выдаче
	Карпуть С.А.	1,4	= (вычисляется)	= (вычисляется)	= (вычисляется)
...	...				

Примечание: Начислено = Размер мин. стипендии (задается) * Коэффициент успеваемости

Обработка:

Вычислить средний коэффициент успеваемости для всех учащихся.

Отобразить данные на уч-ся с заданным коэффициентом успеваемости.

Отсортировать данные в алфавитном порядке колонки "Ф.И.О".

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 8.

Структура данных: "Декларация о доходах".

№ п/п	Сделка (вид услуги)	Доход, руб.	Расход, руб.	Прибыль, руб.	Налог, руб.
1	Ремонт ПЭВМ	285000	124000	= (вычисляется)	= (вычисляется)
...	...				

Обработка:

Вычислить суммарную прибыль, в зависимости от её размера вычисляется налог для каждой сделки. Если размер суммарной прибыли меньше 3 000 000 руб., то налог составляет 12% от прибыли. При размере суммарной прибыли от 3 000 000 до 5 000 000 руб. - 15%; более 5 000 000 руб. 20%.

Вычислив налог для каждой сделки, определить суммарный.

Отсортировать данные по возрастанию прибыли.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 9.

Структура данных: "Ведомость начисления заработной платы".

№ п/п	Ф.И.О.	Разряд	Козф. ставки	Начислено, руб.	Налог (13%), руб.	К выдаче, руб.
1	Авдеева А.И.	12	1,2	= (вычисляется)	= (вычисляется)	= (вычисляется)

Примечание: Начислено = 58000 * Коэффициент ставки * Разряд

Обработка:

Определить сотрудника с наибольшей суммой "К выдаче".

Отобразить данные на сотрудников с заданным разрядом.

Отсортировать данные в алфавитном порядке колонки "Ф.И.О".

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 10.

Структура данных: "Справочная ж/д вокзала".

№ п/п	Пункт назначения	Время отправления	Тип вагона	Наличие мест
1	Гродно	12.15	спальный	нет
...	...			

Обработка:

Клиент вводит запрос, в котором указывает: требуемый ему пункт назначения, интервал времени отправления и требуемый тип вагона (например, "Минск", с 10.00 до 11.30, "общий"). Программа отображает данные из таблицы, удовлетворяющие запрос клиента с наличием мест.

Отсортировать данные по возрастанию времени отправления.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 11.

Обработка:

Вычислить среднее время выработки одного изделия по данным всей таблицы.

Отобразить данные, у которых сумма меньше указанной суммы.

Отсортировать данные по возрастанию общего времени на изготовление изделий.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Структура данных: "Ведомость выработки изделий".

Наименование изделия	Норма времени на 1 изд., час	Расценка на 1 изд., руб.	Кол-во изделий	Общее время, час	Сумма, руб.
Микросхема МП-1201	0,8	49000	25	= (вычисляется)	= (вычисляется)

Вариант 12.

Структура данных: "Ведомость определения себестоимости продукции".

Наименование изделия	Себестоимость единицы изделия, руб.		Кол-во изделий	Себестоимость партии изделий	
	плановая	фактическая		плановая	фактическая
Линза очковая -1,5D	1100	1210	1200	= (вычисляется)	= (вычисляется)

Обработка:

Вычислить суммарную разность между плановой и фактической себестоимостью единицы изделия.

Отобразить данные на те изделия, у которых фактическая себестоимость меньше плановой себестоимости.

Отсортировать данные в алфавитном порядке наименования изделия.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 13.

Обработка:

Вычислить средний предлагаемый оклад.

Отобразить данные по запросу клиента. В запросе клиент указывает следующие свои данные: "профессия", "образование", "имеющийся стаж", "минимально возможное значение оклада".

Отсортировать данные по убыванию оклада.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Структура данных: "Список вакантных рабочих мест на бирже труда".

Предприятие (организация)	Должность (профессия)	Требуемое образован ие	Требуемый стаж, лет	Оклад, руб.
ОАО "Виктория"	Программист	Высшее	3	5100000
...				

Вариант 14.

Структура данных: "Сведения об учащихся".

Ф.И.О.	Адрес	Телефон	Средний балл за семестр	Коэффициент на стипендию
Ивашевич Ю.А.	ул. Советская, 23-12	2-13-45	6,82	= (вычисляется)
...				

Примечание: Коэффициент на стипендию (k) зависит от значения среднего балла за семестр. Если средний балл < 5 , то $k=0$ (стипендия не начисляется); от 5 до 6 - $k=1$; от 6 до 8 - $k=1.2$; от 8 до 9 - $k=1.4$; от 9 до 10 - $k=1.8$

Обработка:

Определить количество учащихся, получающих повышенную стипендию ($k>1$).

Отобразить данные на уч-ся, у которых средний балл выше 7.00.

Отсортировать данные в алфавитном порядке колонки "Ф.И.О".

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 15.

Структура данных: "Результаты сдачи экзаменационной сессии учащихся группы".

Ф.И.О.	Программирование	Математика	Электроника	Средний балл
Ивашевич Ю.А.	9	5	6	= (вычисляется)
...				

Обработка:

Определить средний балл группы.

Отобразить данные на неуспевающих студентов (у кого хотя бы одна оценка меньше 4).

Отсортировать данные в порядке убывания среднего балла.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 16.

Структура данных: "Результаты сдачи экзаменационной сессии учащихся факультета".

№ п/п	Ф.И.О.	группа	Программирован ие	Электрони ка	Средний балл
1	Ивашевич Ю.А.	141ИИТ	5	6	= (вычисляется)

Обработка:

Вычислить общий средний балл.

Отобразить данные по группам в алфавитном порядке "Ф.И.О."

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 17.

Структура данных: "Список деталей для обработки на станке".

№ п/п	Код детали	Время наладки станка, мин	Время обработки, мин	Количество деталей
1	Д12-М	45	52	10
...	...			

Обработка:

Определить суммарное время в часах для обработки всех деталей. Учесть, что наладка станка выполняется один раз для целой партии деталей одного кода.

Отобразить данные для деталей, время наладки которых больше заданного значения.

Отсортировать данные в порядке возрастания времени обработки.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 18.

Обработка:

Покупатель вводит запрос, в котором указывает требуемые ему параметры: модель, максимальную стоимость и максимальный расход горючего. Программа отображает данные из таблицы, удовлетворяющие запрос покупателя.

Отсортировать данные в порядке убывания стоимости.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Структура данных: "Список автомобилей для продажи".

№ п/п	Модель	Стоимость, \$	Расход горючего, л/100 км	Комфортность
1	BMW-4	45200	7	Отличная
...	...			

Вариант 19.

Структура данных: "Справочная автовокзала".

№ п/п	Пункт назначения	Время отправления	Время прибытия	Стоимость, руб.
1	Гродно	12.15	14.45	30 800
...	...			

Обработка:

Клиент вводит запрос, в котором указывает: требуемый ему пункт назначения и интервал возможного времени отправления (например, "Минск", с 10.00 до 11.30). Программа отображает данные из таблицы, удовлетворяющие запрос клиента.

Отсортировать данные по возрастанию времени отправления.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 20.

Структура данных: "Справочная ж/д вокзала".

№ п/п	Пункт назначения	Время отправления	Тип вагона	Наличие мест
1	Гродно	12.15	спальный	нет
...	...			

Обработка:

Клиент вводит запрос, в котором указывает: требуемый ему пункт назначения, интервал времени отправления и требуемый тип вагона (например, "Минск", с 10.00 до 11.30, "общий"). Программа отображает данные из таблицы, удовлетворяющие запрос клиента с наличием мест.

Отсортировать данные по возрастанию времени отправления.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 21

Структура данных: "Ведомость начислений и отчислений".

Табельный номер	Зарплата а, руб.	Премия, руб.	Начислено, руб.	Налог (12%), руб.	К выдаче, руб.
254	2850000	540000	= (вычисляется)	= (вычисляется)	= (вычисляется)
...					

Обработка:

Определить сотрудника с наименьшей суммой "К выдаче".

Отобразить данные о сотрудниках, у которых премия составляет более 20% от зарплаты.

Отсортировать данные по убыванию премии.

Отсортировать данные по № п/п (вернуться к исходной последовательности строк).

Вариант 22

1 Разработать программу для итерационного поиска размеров равномерных выплат по кредиту. Выплаты и начисления процентов на оставшуюся сумму производятся ежемесячно. Исходные данные: размер кредита, срок выплат (от 1 до 5

лет), процентная ставка. В диалоговом окне изобразить график изменения остатка от суммы кредита.

Вариант 23

Написать программу, которая обучает правилам перевода числа в десятичной системе счисления в следующие системы счисления: двоичную, восьмеричную и шестнадцатеричную.

Вариант 24

Класс `Bill` представляет собой разовый платёж за телефонный разговор и включает поля: фамилия плательщика, номер телефона, тариф за минуту разговора, скидка в процентах), время начала разговора, время окончания разговора, сумма к оплате. Используя класс `Bill`, реализовать класс `ListPayer`. Класс `ListPayer` содержит список плательщиков за телефонные услуги, дату создания списка, номер списка. Один элемент списка включает информацию о плательщике (класс `Bill`), статус оплаты, дату платежа, сумму платежа. Реализовать методы добавления плательщиков в список и удаления из него; метод поиска плательщика по номеру телефона и по фамилии, по дате платежа; метод вычисления полной стоимости платежей всего списка. Реализовать операцию объединения и операцию пересечения списков. Реализовать операцию генерации конкретного объекта `Group` (группа), содержащего список плательщиков, из объекта типа `ListPayer`. Должна быть возможность выбирать группу плательщиков по признаку оплаты, по дате платежа, по номеру телефона. Дата представляется структурой с тремя полями типа `unsignedint`: год, месяц, день. Время представляется структурой с тремя полями типа `unsignedint`: час, минута, секунда.

Вариант 25

Результаты соревнований по прыжкам в длину представлены в виде матрицы 5x3 (5 спортсменов по 3 попытки у каждого). Указать, какой спортсмен и в какой попытке показал наилучший результат.

Вариант 26

Разработать программу, в которой будет организовано меню, выбор функций меню должно быть организовано по функциональной клавише. Вся информация должна храниться в массиве структур, с возможностью их записи в файл.

Функции программы:

функция для ввода данных пользователем;

функция для записи данных в файл;

чтение данных из файла;

вывод данных на экран;

дополнение данных;

удаление данных;

поиск информации по атрибуту;

Состав и структура исходных данных:

Таблица «штатное расписание» с полями:

«ФИО работника»

«кол-во отработанных часов»

«должность»

«оплата за час работы»

«присваиваемый id»

вычисляемое поле «зарплата»

Примечание: зарплата = количество отработанных часов умножить на оплату за час работы.

Вариант 27

Необходимо создать класс - зоомагазин. В классе должны быть следующие поля: животное (волк, пингвин, собака), пол, имя, цена, количество. Включить в состав класса необходимый минимум методов, обеспечивающий полноценное функционирование объектов указанного класса:

Конструкторы (по умолчанию, с параметрами, копирования);

Деструктор;

Переопределить возможные для класса операции, продумать порядок их выполнения;

Добавить необходимые методы.

Предоставить возможность вводить данные с клавиатуры или из файла (с помощью конструктора или операцией).

Вариант 28

Создать структуру, которая будет хранить информации о знаке зодиака, с именем `Znak`. Структура `Znak` должна содержать следующие поля:

- `name` - фамилия, имя и отчество;
- `zodiac` - знак зодиака;
- `bday` - день рождения (массив из трёх чисел).

Программа, должна уметь выполнять следующие действия:

- ввод с клавиатуры данных в массив `book`, который состоит из восьми элементов типа `Znak`;
- вывод на экран информации о людях, родившихся в месяц, значения которого введено с клавиатуры, если таких нет, выдать на дисплей соответствующее сообщение.
- упорядочивание записей по знакам Зодиака.

Вариант 29

Разработать три класса, которые следует связать между собой, используя наследование:

- 1) класс `Product`, который имеет три элемента данных: имя, цена и вес товара (базовый класс для всех классов);
- 2) класс `Buy`, содержащий данные о количестве покупаемого товара в штуках, о цене за весь купленный товар и о весе товара (производный класс для класса `Product` и базовый класс для класса `Check`);

3) класс Check, не содержащий никаких элемент-данных. Данный класс должен выводить на экран информацию о товаре и о покупке (производный класс для класса Buy);

Для взаимодействия с данными классов разработать set- и get-методы. Все элемент-данные классов объявлять как private.

Список использованных источников

1 Ашарина И.В. Объектно-ориентированное программирование в С++: лекции и упражнения / И.В. Ашарина. - М.: Горячая линия -Телеком, 2008. - 320 с. ISBN 978-5-9912-0038-7

2 Павловская Т.А. С++. Объектно-ориентированное программирование: практикум / Т.А. Павловская, Ю.А. Щупак. - Спб.: Питер, 2008. - 265 с. ISBN 978-5-94723-842-6

3 Хорев, П. Б. Технологии объектно-ориентированного программирования/ П. Б. Хорев .- 2-е изд., стер. - М.: Академия, 2008. - 448 с. - (Высшее профессиональное образование). - ISBN 978-5-7695-5262-5.

4 Бабушкина, И.А. Практикум по объектно-ориентированному программированию [Электронный ресурс] / И. А. Бабушкина, С. М. Окулов. - 3-е изд. (эл.). - М.: БИНОМ. Лаборатория знаний, 2012. - 366 с.- Режим доступа:

<http://znanium.com/catalog.php?bookinfo=366434>

5 Галявов, И. Р. Borland С++ для себя [Электронный ресурс] / И. Р. Галявов. - М.: ДМК Пресс, 2009. - 432 с.- Режим доступа:

<http://znanium.com/bookread2.php?book=408232>