МИНИСТЕРСТВО ОБРАЗОВАНИЯ ТВЕРСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕСИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
«ТВЕРСКОЙ КОЛЛЕДЖ ИМЕНИ А.Н.КОНЯЕВА»

# Методическая разработка

для проведения лабораторной работы

по дисциплине:

**«Основы алгоритмизации и программирования»**

по теме:

**«Borland C++ Builder. Функции».**

для специальности
09.02.03 «Программное обеспечение вычислительной техники и автоматизированных систем»

**2015г.**

**Содержание**

## Введение

Методическая разработка для проведения лабораторной работы по дисциплине «Основы алгоритмизации и программирования» по теме «Borland C++ Builder. Функции» состоит из теоретического введения и экспериментальной части, состоящей из трёх заданий, предлагаемых к выполнению в определенном порядке с нарастающей сложностью. Выполнение лабораторной работы ориентировано на визуальную среду программирования Borland C++ Builder. Посредством выполнения лабораторной работы происходит ознакомление с приемами описания и использования функций и приобретение навыков создания приложений с использованием функций. Лабораторная работа обеспечивает приобретение навыков, которые могут быть использованы при курсовом проектировании по дисциплинам : «Технология разработки программного продукта», «Математические методы». Методическая разработка по теме «Borland C++ Builder. Функции» рассчитана на пользователей, владеющих навыками программирования в среде Borland C++ Builder, и может быть предложен а студентам 2-х, 3 курсов специальности 09.02.03.

*ЛАБОРАТОРНАЯ РАБОТА*

**Основы Алгоритмизации и программирования**

**Тема:** Borland C++ Builder. Функции**.**

**Цель занятия**: Приобрести навыки составления программ с использованием функций.

**Оборудование**: ПК платформы IBM PC.

**Теоретическое введение**

***Функция*** *представляет собой программный блок, который может вызываться из разных частей программы.*

Функция может принимать параметры и возвращать значение.

Функция описывается следующим образом:

**type_**возвращаемого значения **имя функции(список параметров)**

```
{
 операторы тела функции
}
```

**Список параметров**, заключенный в скобки, представляет собой разделенный запятыми список вида:

**type_**параметра **Идентификатор_**параметра

Выход из функции, которая должна возвращать значение, Осуществляется оператором

**return** выражение;

,где выражение должно формировать возвращаемое значение и соответствовать типу, объявленному в заголовке функции, например,

return **m*s;**

Описание функции, возвращающей значение, может иметь вид:

```
float ff(int m, float s, char h)  {
  тело функции
  return m*s;
}
```

Описание функции, не возвращающей никакого значения, имеет вид:

```
void имя_функции(список параметров)     {
        тело функции
}
```

Описание функции, не принимающей никакого параметра, имеет вид:

**type**_возвращаемого значения **имя**_функции**(void)**
**{**
тело функции
**}**

Помимо описания функции в текст программы включается также **прототип функции –** её предварительное объявление. Прототип функции представляет собой тот же заголовок функции, но с **(;)** в конце. Например,

**float ff(int m, float s, char h);**

В прототипе можно не указывать имена параметров, например,

**float ff(int , float , char );**

Заголовок описания функции, в которую необходимо передать массив **M[],** который не будет изменяться в функции может иметь вид:

t**ype_**возв.знач. **имя_функции(const type *M, int n)**

а для массива, который будет изменяться,

t**ype_**возв.знач. **имя_функции( type *M, int n) ,**

где **n-** число элементов массива
Обратится к функции и в том и в другом случае можно указав
**имя_функции( M,n)**

Пример модуля, в котором описываются функции и используются, имеет следующую структуру:

```
{  float ff(int , float , char );        /*прототипы функций*/
   void fm(int *M, int n);
   float s,f;              /*текст программы,вызывающей функции*/
   int  n,k,M[10];
   s+=ff(n,f," ");
   fm(M);
   ……..
  }
  float ff(int m, float s, char h)  {
          тело функции
          return m*s;
        }
  void fm(int *M, int n)    {
  ……..
 }
```

**Используемые источники**

1. В. Шелест «Программирование. Структурный подход. Turbo Pascal., Borland C++».
2. С.А. Павловская «С/С++ Программирование на языке высокого уровня».
3. А.А. Архангельский. «Практикум по программированию в среде С++ Builder 5

**Контрольные вопросы**

1. Что называется функцией?
2. Параметры какого типа можно передавать в функцию?
3. Какое ключевое слово необходимо указать для возвращаемого значения функции, если она не возвращает никакого значения?
4. Что такое прототип функции?
5. Где можно размещать прототип функции, а где описание функции?
6. Как можно обратиться к функции (вызвать функцию)?

**Порядок выполнения работы**

1. Внимательно прочитать тему занятия и цель, записать в тетрадь.
2. Ознакомиться с теоретическим введением.
3. Ответить на контрольные вопросы.
4. Выполнить Задание 1 экспериментальной части, показать преподавателю, записать в тетрадь.
5. Выполнить Задание 2 ( один из вариантов) экспериментальной части, показать преподавателю, записать в тетрадь.
6. Выполнить Задание 3 экспериментальной части, показать преподавателю, записать в тетрадь.
7. Составить отчет о проделанной работе посредством редактора Word.

**Оформление результата**

Записать тему, цель занятия, приложить листинг исходной программы с результат работы. Ответить на контрольные вопросы.

**Экспериментальная часть**

**Задание 1.** Вычислить площадь круга для радиусов:

**r=5; r=10; r=15;**

Вычисление площади круга оформить в виде функции.

Результаты вывести в окнах компонента Label**.**

**Задание 2. Создать функцию**

**Вариант 1.** Создать функцию для определения максимума среди элементов массива. Определить максимумы для массивов:

**A[]={3,2,5,3,1,6};**

**B[]={4,5,1,1,8,7};**

Максимумы вывести в окнах компонента **Label.**

**Вариант 2.** Составить функцию определения среднего арифметического элементов массива. Используя функцию определить среднее арифметическое для массивов:

**C[]={10,55,77,99};**

**B[]={45,88,11,20};**

Наибольшее полученное значение вывести в окне компонента **Label.**

**Вариант 3.** Составить функцию произведения всех элементов массива. Используя функцию определить произведения элементов массивов:

**A[]={3,2,5,3,1,6};**

**B[]={4,5,1,1,8,7};**

Среднее арифметическое полученных значений вывести в окне компонента **Label..**

**Вариант 4.** Создать функцию для определения минимума среди элементов массива. Определить минимумы для массивов:

**A[]={3,2,5,3,1,6};**

**B[]={4,5,1,1,8,7};**

Наименьшее полученное значение вывести в окне компонента **Label.**

**Вариант 5.** Создать функцию сортировки массива по возрастанию. Используя функцию отсортировать массив **M[]={6,3,1,7,2,9,4}** и вывести среднее значение его максимального и минимального элементов в окне компонента **Label.**

**Вариант 6.** Создать функцию сравнения соответствующих элементов двух массивов и определения количества неравных элементов. Сравнить два массива:

**A[]={1,2,3,7,8,9};**

**B[]={1,2,3,6,7,8};**

, количество неравных элементов вывести в окне компонента **Label.**

**Вариант 7.** Создать функцию сортировки массива по убыванию. Используя функцию отсортировать массив **M[]={6,3,1,7,2,9,4}** и вывести среднее значение его максимального и минимального элементов в окне компонента **Label.**

**Вариант 8.** Создать функцию, которая в любом массиве заменяет значения, большие 50 на значение (50), а значения, меньшие 0 на значение (1). Подсчитать количество элементов, замененных на (1) и на (50), вывести эти значения в окнах компонента **Label..**

**Задание 3. Описать функцию подготовки строки, содержащей элементы массива типа int в строковом виде, для отображения исходного массива в окне компонента Label. Применить полученную функцию для отображения исходных массивов при выполнении Задания 2.**

**Выставление оценки**

Оценка «3» выставляется за выполнение Задания 1.
Оценка «4» выставляется за выполнение Заданий 1,2.
Оценка «5» выставляется за выполнение Заданий 1,2,3.

## Приложение

## Отчетная карта

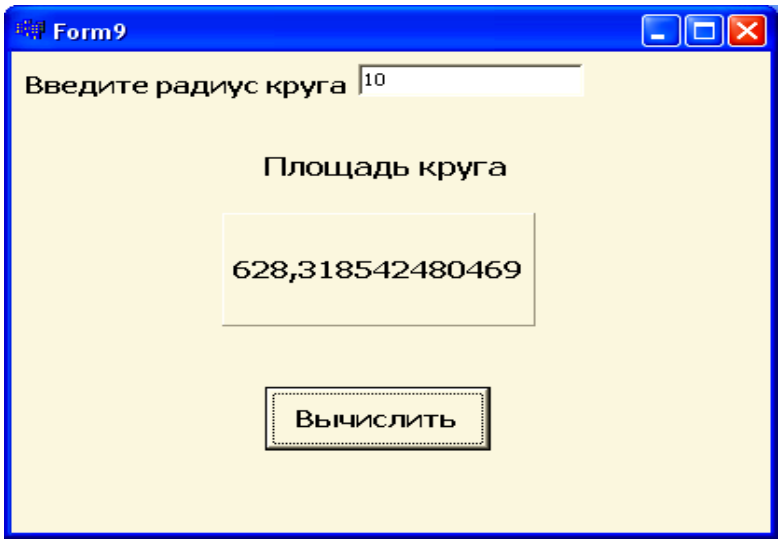
**Лабораторная работа №13**
**Дисциплина:** Основы алгоритмизации и программирования
**Тема:** Borland C++ Builder. Функции.

**Задание 1.** Вычислить площадь круга для радиусов: r=5; r=10; r=15;
Вычисление площади круга оформить в виде функции.
Результаты вывести в окнах компонента **Label.**

*Исходный код программы*

```
#include <vcl.h>
#pragma hdrstop
#include "Unit9.h"
#include "Math.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm9 *Form9;
float ploschad(float r)
{float k;
k=2*M_PI*r*r;
return k;}
//---------------------------------------------------------------------------
__fastcall TForm9::TForm9(TComponent* Owner)
        : TForm(Owner)
{}
//---------------------------------------------------------------------------
void __fastcall TForm9::Button1Click(TObject *Sender)
{
float S,r;
r=StrToFloat(Edit1->Text);
Panel1->Caption=FloatToStr(ploschad(r));    }
```

***Окно результата:***



**Задание 2.** ***Вариант 6:***

Создать функцию сравнения соответствующих элементов двух массивов и определения количества неравных элементов.Сравнить два массива:

A[ ]={1,2,3,7,8,9};  B[ ]={1,2,3,6,7,8};

Количество неравных элементов вывести в окне компонента Edit.

**Задание 3:**

Описать функцию подготовки строки, содержащей элементы массива типа int, для отображения исходного массива в окне компонента Label. Применить полученную функцию для отображения исходных массивов при выполнение Задания 2.

***Исходный код программы с используемыми функциями kol и mas:***

```
#include <vcl.h>
#pragma hdrstop
#include "Unit5.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm9 *Form9;
//---------------------------------------------------------------------------
__fastcall TForm9::TForm9(TComponent* Owner)
        : TForm(Owner)
{}
```

***// Исходных код функции kol поиска неравных элементов массивов:***

```
int kol(const int *A, const int *B, AnsiString &S, int n){
int k=0,i;
for (i = 0; i <n; i++) {
        if (A[i]!=B[i])  {S=S+' '+IntToStr(A[i])+' '+IntToStr(B[i]);
         k++;            }     }
return k;  }
```

***// Исходный код функции mas представления элементов массива в виде строки:***

```
AnsiString mas(const int *A, int n){
int i;  AnsiString S1;
for (i = 0; i < n; i++) {
S1=S1+' '+IntToStr(A[i]);    }
return S1;       }
```

*// Исходный код основной программы:*

```
void __fastcall TForm9::Button1Click(TObject *Sender)
{  AnsiString S;  int A[]={1,2,3,7,8,9},  B[]={1,2,3,6,7,8};
Edit1->Text=IntToStr(kol(A,B,S));
Label1->Caption=S;
Label2->Caption=mas(A,6);
Label3->Caption=mas(B,6);
 }
```

***Окно результата:***