

Колледж космического машиностроения и технологий

**ОТЧЁТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ УП.01**

**По модулю ПМ.01 «Разработка программных модулей программного обеспечения для компьютерных систем»**

**09.02.03 Программирование в компьютерных системах**

**Студенты гр.** П2-16

Головатюк М.В.

(Фамилия, И. О.)

**Руководитель практики**

Гусятинер Л. Б.

(Фамилия, И. О.)

**Оценка**

**Королёв, 2020 г.**

СОДЕРЖАНИЕ

[Занятие 1. Числа Фибоначчи 4](#_Toc7210386)

[Занятие 2. Наибольший общий делитель 7](#_Toc7210387)

[Занятие 3. Linux. Создание динамической библиотеки 9](#_Toc7210388)

[Занятие 4. Linux. Поиск ppid с использованием profs 10](#_Toc7210389)

[Занятие 5. Введение в синтаксис C++, часть 2 11](#_Toc7210390)

[Занятие 6. Набор задач для программирования на С 15](#_Toc7210391)

[Занятие 7. Ветвление в С 20](#_Toc7210392)

[Занятие 8. Циклы в С 25](#_Toc7210393)

[Занятие 9. Одномерные массивы в С 30](#_Toc7210394)

[Занятие 10. Матрицы в С 32](#_Toc7210395)

[Занятие 11. Транспонирование матрицы 34](#_Toc7210396)

[Занятие 12. Создание презентации 36](#_Toc7210397)

[Литература и интернет-источники ч.1 37](#_Toc7210398)

[Дневник УП.01 (2 часть) 38](#_Toc7210399)

[Занятие 13. Постановка задачи программы “Калькулятор” 40](#_Toc7210400)

[Занятие 14. Разработка отдельных спецификаций компонент программы “Калькулятор” 42](#_Toc7210401)

[Занятие 15. Разработка кода программного продукта программы “Калькулятор” 43](#_Toc7210402)

[Занятие 16. Отладка программных модулей программы “Калькулятор” 45](#_Toc7210403)

[Занятие 17. Тестирование программных модулей. Оптимизация программного кода модуля программы “Калькулятор” 47](#_Toc7210404)

[Занятие 18. Разработка руководства программы “Калькулятор”. Создание презентации 49](#_Toc7210405)

[Занятие 19. Постановка задачи “Графический редактор” 51](#_Toc7210406)

[Занятие 20. Разработка спецификаций отдельных компонент программы “Графический редактор” 53](#_Toc7210407)

[Занятие 21. Разработка кода программного продукта программы “Графический редактор” 55](#_Toc7210408)

[Занятие 22. Отладка программных модулей программы “Графический редактор” 57](#_Toc7210409)

[Занятие 23. Тестирование программных модулей, оптимизация программного кода модуля программы “Графический редактор” 59](#_Toc7210410)

[Занятие 24. Разработка руководства оператора программы “Графический редактор”. Создание презентации 61](#_Toc7210411)

[Занятие 25. Постановка задачи проекта “Семейный бюджет” 63](#_Toc7210412)

[Занятие 26. Добавление основных элементов в проект “Семейный бюджет” 64](#_Toc7210413)

[Занятие 27. Настройка подключения к базе данных в проекте “Семейный бюджет” 65](#_Toc7210414)

[Занятие 28. Функция выбора конкретной таблицы, функция добавления записи в проекте “Семейный бюджет” 66](#_Toc7210415)

[Занятие 29. Функция удаления записи в проекте “Семейный бюджет” 67](#_Toc7210416)

[Занятие 30. Создание SQL-запроса в проекте “Семейный бюджет” 68](#_Toc7210417)

[Занятие 31. Вывод результата запроса в XML 69](#_Toc7210418)

[Занятие 32. Вывод результата запроса в Word 70](#_Toc7210419)

[Занятие 33. Вывод результата запроса в Excel 72](#_Toc7210420)

[Занятие 34. Внешний вид проекта “Семейный бюджет” 73](#_Toc7210421)

[Занятие 35. Подготовка презентации 74](#_Toc7210422)

[Занятие 36. Защита 75](#_Toc7210423)

[Литература и интернет-источники ч.2 76](#_Toc7210424)

# **Занятие 1. Числа Фибоначчи**

**Задание №1:**

Дано целое число 1≤n≤40, необходимо вычислить n-е число Фибоначчи (напомним, что F0=0, F1=1 и Fn=Fn−1+Fn−2 при n≥2).

**Блок-схема:**

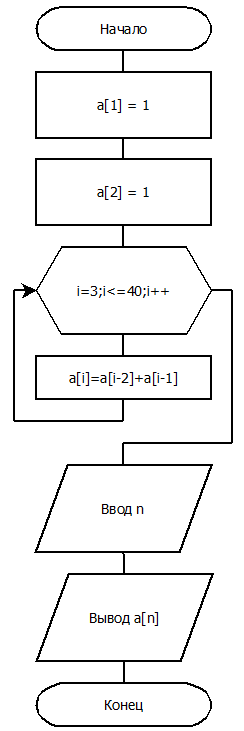


Рис. 1.1.1 Блок-схема к листингу 1.1

**Листинг 1.1(1.1.cpp)**

/\*\*

Дано целое число 1≤n≤40, необходимо вычислить n-е число Фибоначчи.

Работа выполнена 10.12.18 студентом группы П2-16 Дуругяном А.В.

\*/

#include <iostream>

using namespace std;

int main(){

int a[50], i, n;

a[1] = a[2] = 1;

for (i = 3; i <= 40; i++)

a[i] = a[i - 2] + a[i - 1];

cin >> n;

cout << a[n];

}

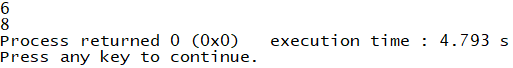


Рис. 1.1.2 Результат работы программы

**Задание №2:**

Дано число 1≤n≤107, необходимо найти последнюю цифру n-го числа Фибоначчи.

**Листинг 1.2(1.2.cpp)**

/\*\*

Дано число 1≤n≤107, необходимо найти последнюю цифру n-го числа Фибоначчи.

Работа выполнена 10.12.18 студентом группы П2-16 Дуругяном А.В.

\*/

#include <iostream>

using namespace std;

int main(){

int i, n;

cin >> n;

int a[n];

a[1] = 1;

a[2] = 1;

for (i = 3; i <= n; i++)

a[i] = (a[i - 2] + a[i - 1]) % 10;

cout << a[n];

}

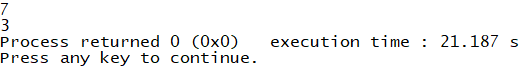


Рис. 1.2.1 Результат работы программы

**Задание №3:**

Даны целые числа 1≤n≤1018 и 2≤m≤105, необходимо найти остаток от деления n-го числа Фибоначчи на m. Листинг 1.2(1.2.cpp)

**Листинг 1.3(1.3.cpp)**

/\*\*

Даны целые числа 1≤n≤1018 и 2≤m≤105,

необходимо найти остаток от деления n-го числа Фибоначчи на m.

Работа выполнена 10.12.18 студентом группы П2-16 Дуругяном А.В.

\*/

#include <iostream>

using namespace std;

int main(){

long i, n, m;

cin >> n >> m;

int a[n];

a[1] = 1;

a[2] = 1;

for (i = 3; i <= n; i++)

a[i] = (a[i - 2] + a[i - 1]) % m;

cout << a[n];

}

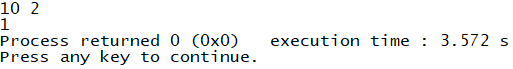


Рис. 1.3.1 Результат работы программы

# **Занятие 2. Наибольший общий делитель**

**Задание:**

По данным двум числам 1≤a,b≤2⋅109 найдите их наибольший общий делитель.

**Блок-схема:**

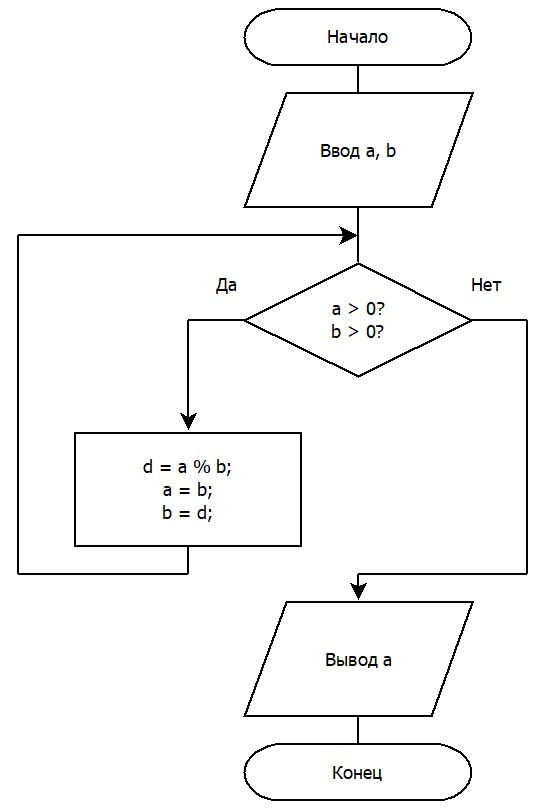


Рис. 2.1.1 Блок-схема к листингу 2.1

**Листинг 2.1(2.1.cpp)**

/\*\*

По данным двум числам 1≤a,b≤2⋅109 найдите их наибольший общий делитель.

Работа выполнена 11.12.18 студентом группы П2-16 Дуругяном А.В.

\*/

#include <iostream>

#include <cmath>

using namespace std;

int main(){

int a, b, d, i;

cin >> a >> b;

while (a > 0 && b > 0){

d = a % b;

a = b;

b = d;

}

cout << a;

}

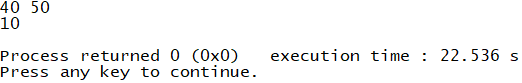


Рис. 2.1.2 Результат работы программы

# **Занятие 3. Linux. Создание динамической библиотеки**

**Задание:**

Необходимо создать разделяемую динамическую библиотеку libsolution.so в которой реализовать функцию со следующим прототипом:

int stringStat(const char \*string, size\_t multiplier, int \*count);

Функция возвращает значение длины строки string, умноженное на multiplier, и увеличивает на 1 значение, на которое указывает count.

**Листинг 3.1.1(solution.c)**

/\* Работа выполнена 12.12.18 студентом группы П2-16 Дуругяном А.В.

#include <stddef.h>

#include <stdio.h>

#include <string.h>

int stringStat(const char \*string, int multiplier, int \*count)

{

\*count = \*count+1;

return strlen(string)\*multiplier;

}

**Листинг 3.1.2(Makefile)**

lib: solution.c

gcc -shared solution.c -fPIC -o libsolution.so

# **Занятие 4. Linux. Поиск ppid с использованием profs**

**Задание:**

Разработать программу solution, которая осуществляет поиск родительского PID текущего процесса в файловой системе proc и выводит найденное значение на консоль. Результат, возвращаемый функций должен быть выведен отдельной строкой(должен оканчиваться символом перевода строки \n) в stdio.

**Листинг (ppid.c)**

/\* Работа выполнена 13.12.18 студентом группы П2-16 Дуругяном А.В.

#include <stdio.h>

#include <stdlib.h>

#include <sys/types.h>

#include <unistd.h>

int main()

{

int pid = getpid();

int ppid = get\_ppid(pid);

printf("%d\n", ppid);

return(0);

}

int get\_ppid(int pid)

{

char str0[100], str1[100], str2[100], str3[100], str4[100];

strcpy(str0, "/proc/");

char value[100];

snprintf(value, 100, "%d", pid);

strcat(str0, value);

strcat(str0, "/stat");

FILE \*fp;

if(!(fp = fopen(str0, "r")))

{

return -1;

}

fscanf(fp, "%s %s %s %s", str1, str2, str3, str4);

fclose(fp);

return atoi(str4);

}

# **Занятие 5.** **Введение в синтаксис C++, часть 2**

**Задание №1:**

Напишите функцию power, реализующую возведение целого числа в неотрицательную целую степень. Функция power должна принимать на вход два целых числа и возвращать целое число (смотрите шаблон кода). При выполнении задания учтите, что функция обязательно должна называться power, функция ничего не должна читать со входа или выводить.

**Листинг 5.1(5.1.cpp)**

/\*\*

Напишите функцию power, реализующую возведение целого числа в

неотрицательную целую степень.

Работа выполнена 14.12.18 студентом группы П2-16 Дуругян А.В.

\*/

#include <iostream>

using namespace std;

int power(int x, int p);

int main(){

int x, p;

cin >> x >> p;

cout << power(x,p);

}

int power(int x, int p) {

int answer;

answer = x;

for (int i = 1; i < p; i++)

answer \*= x;

if (p == 0)

return 1;

return answer;

}

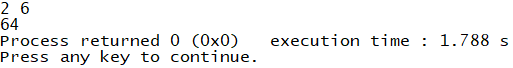


Рис. 5.1.1 Результат работы программы

**Задание №2:**

Напишите программу, которая суммирует целые числа. На вход программе подаются целые числа в следующем формате: на первой строке идет целое число T — количество тестов, далее следует T строк, в каждой из которых через пробел идут два целых числа ai и bi. На выводе для каждой из T строк нужно вывести сумму ai+bi в том порядке, в котором пары поступают на вход. Ничего, кроме этого, выводить не нужно.

**Блок-схема:**

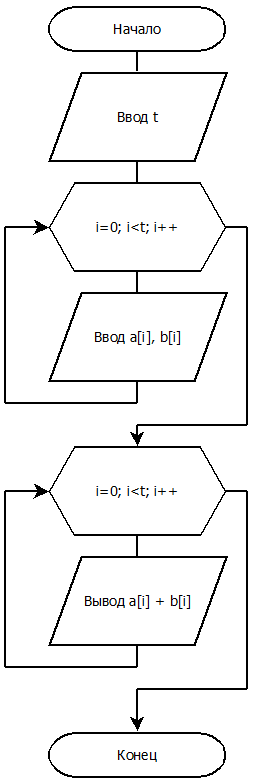


Рис. 5.2.1 Блок-схема к листингу 5.2

**Листинг 5.2(5.2.cpp)**

/\*\*

Напишите программу, которая суммирует целые числа.

Работа выполнена 14.12.18 студентом группы П2-16 Дуругяном А.В.

\*/

#include <iostream>

using namespace std;

int main(){

int a[100], b[100], t;

cin >> t;

for (int i = 0; i < t; i++){

cin >> a[i] >> b[i];

}

for (int i = 0; i < t; i++)

cout << a[i] + b[i] << endl;

}

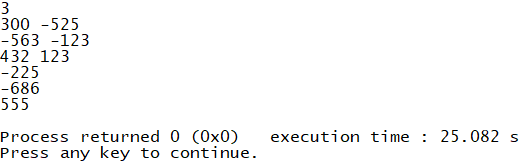


Рис. 5.2.1 Результат работы программы

**Задание №3:**

Напишите программу для решения квадратных уравнений вида ax2+bx+c=0 (относительно x). На вход программа получает три целых числа: a , b и c, соответственно. При этом гарантируется, что a≠0. На вывод программа должна вывести два вещественных корня уравнения, разделённые пробелом. Если вещественных корней нет, то программа должна вывести строку "No real roots". Если у уравнения имеется только один корень (кратный корень), то программа должна вывести его дважды.

**Листинг 5.3(5.3.cpp)**

/\*\*

Напишите программу для решения квадратных уравнений вида ax2+bx+c=0 (относительно x).

На вход программа получает три целых числа: a , b и c, соответственно.

При этом гарантируется, что a≠0. На вывод программа должна вывести

два вещественных корня уравнения, разделённые пробелом.

Если вещественных корней нет, то программа должна вывести

строку "No real roots". Если у уравнения имеется только один корень (кратный корень),

то программа должна вывести его дважды.

Работа выполнена 14.12.18 студентом группы П2-16 Дуругяном А.В.

\*/

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

int a, b, c;

cin >> a >> b >> c;

int d = (b\*b) - (4 \* a \* c);

if (d > 0){

cout << ((b\*(-1)) + sqrt(d)) / (2\*a) <<' ';

cout << ((b\*(-1)) - sqrt(d)) / (2\*a);

}

else if (d == 0){

cout << (b\*(-1)) / (2\*a) <<' ';

cout << (b\*(-1)) / (2\*a);

}

else

cout << "No real roots";

return 0;

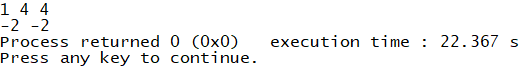


Рис. 5.3.1 Результат работы программы

# **Занятие 6.** **Набор задач для программирования на С**

**Задание №1:**

Идёт k секунда суток. Определить, сколько целых часов и целых минут прошло с начала суток.

**Блок-схема:**

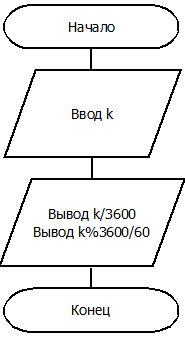


Рис. 6.1.1 Блок-схема к листингу 6.1

**Листинг 6.1(6.1.c)**

/\*\*

Идёт k секунда суток. Определить, сколько целых часов и целых минут прошло с начала суток.

Работа выполнена 15.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int k;

scanf("%d", &k);

printf("%d %d", k/3600, k%3600/60);

return 0;

}

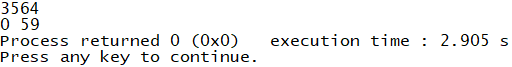


Рис. 6.1.1 Результат работы программы

**Задание №2:**

Написать программу для перевода числа из системы счисления с основанием k в систему счисления с основанием 10.

**Листинг 6.2(6.2.c)**

/\*\*

Написать программу для перевода числа из системы счисления

с основанием k в систему счисления с основанием 10.

Работа выполнена 15.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int a,k;

scanf("%d %d", &k, &a);

printf("%d", a%10+a%100/10\*k+a/100%10\*(k\*k)+a/1000\*(k\*k\*k));

return 0;

}

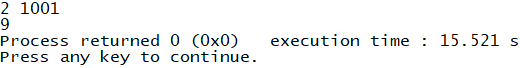


Рис. 6.2.1 Результат работы программы

**Задание №3:**

Напишите программу, которая считывает два натуральных числа, после чего определяет, кратно ли первое второму.

**Листинг 6.3(6.3.c)**

/\*\*

Напишите программу, которая считывает два натуральных числа,

после чего определяет, кратно ли первое второму.

Работа выполнена 15.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int a,b;

scanf("%d %d", &a,&b);

printf("%d", a>b ? a%b : b%a);

return 0;

}

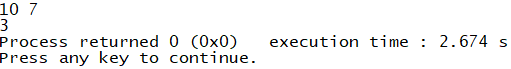


Рис. 6.3.1 Результат работы программы

**Задание №4:**

Идёт k день года. Определить, номер дня недели. Предполагаем, что 1 января выпало на среду. Дни недели нумеруем следующим образом:

0 -- воскресенье

1 -- понедельник

2 -- вторник

3 -- среда

4 -- четверг

5 -- пятница

6 – суббота

**Листинг 6.4(6.4.c)**

/\*\*

Идёт k день года. Определить, номер дня недели. Предполагаем, что 1 января выпало на среду.

Дни недели нумеруем следующим образом:

0 -- воскресенье

1 -- понедельник

2 -- вторник

3 -- среда

4 -- четверг

5 -- пятница

6 – суббота

Работа выполнена 15.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int i;

scanf("%d", &i);

printf("%d", (i + 2) % 7);

return 0;

}

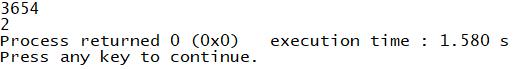


Рис. 6.4.1 Результат работы программы

**Задание №5:**

Написать программу, которая осуществляет операцию циклический сдвиг влево для целого трёхзначного числа k.

**Листинг 6.5(6.5.c)**

/\*\*

Написать программу, которая осуществляет операцию

циклический сдвиг влево для целого трёхзначного числа k.

Работа выполнена 15.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int k;

scanf("%d", &k);

printf("%d", k % 100 \* 10 + k / 100);

return 0;

}

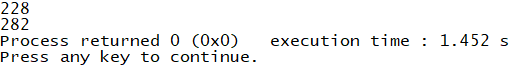


Рис. 6.5.1 Результат работы программы

# **Занятие 7.** **Ветвление в С**

**Задание №1:**

Для целого числа K (от 1 до 99 включительно) напечатать фразу «Мне K лет», учитывая при этом, что при некоторых значениях K слово «лет» надо заменить на слово «год» или «года». Например, 11 лет, 22 года, 51 год.

**Листинг 7.1(7.1.c)**

/\*\*

Для целого числа K (от 1 до 99 включительно) напечатать фразу «Мне K лет»,

учитывая при этом, что при некоторых значениях K слово «лет» надо заменить

на слово «год» или «года». Например, 11 лет, 22 года, 51 год.

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main()

{

int b, a;

scanf("%d", &b);

a = b % 10;

switch (b) {

case 11:

case 12:

case 13:

case 14:

printf("Мне %d лет\n", b);

break;

default:

switch (a) {

case 1:

printf("Мне %d год\n", b);

break;

case 2:

case 3:

case 4:

printf("Мне %d года\n", b);

break;

default:

printf("Мне %d лет\n", b);

break;

}

}

return 0;

}

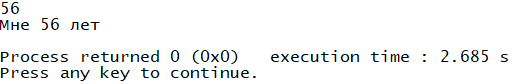


Рис. 7.1.1 Результат работы программы

**Задание №2:**

Напишите программу-калькулятор для четырёх основных арифметических действий.

**Листинг 7.2(7.2.c)**

/\*\*

Напишите программу-калькулятор для четырёх основных арифметических действий.

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int a, b;

char c;

scanf("%d %d %c",&a,&b,&c);

switch(c){

case '+':printf("%.2lf",(double)a+b); break;

case '-':printf("%.2lf",(double)a-b); break;

case '\*':printf("%.2lf", (double)a\*b); break;

case '/':

switch (b) {

default:printf("%.2f",(double)a/b); break;

case 0:printf("ERROR!"); break;

}

break;

default:printf("ERROR!");

}

return 0;

}

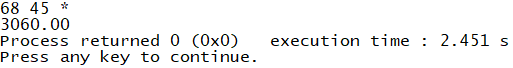


Рис. 7.2.1 Результат работы программы

**Задание №3:**

Напишите программу сравнивающие два целых числа.

**Листинг 7.3(7.3.c)**

/\*\*

Напишите программу сравнивающие два целых числа.

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int x,y;

scanf("%d %d", &x, &y);

putchar('0'+(x==y));

}

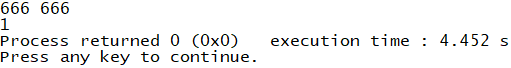


Рис. 7.3.1 Результат работы программы

**Задание №4:**

Написать программу, моделирующую работу лифта, находящегося на первом этаже. Допустим, что каждый заходящий в кабину лифта человек нажимает кнопку с номером этажа. После этого лифт определяет наиболее быстрый порядок транспортирования пассажиров, расставляя соответствующим образом этажи, на которых он должен остановиться.

**Блок-схема:**

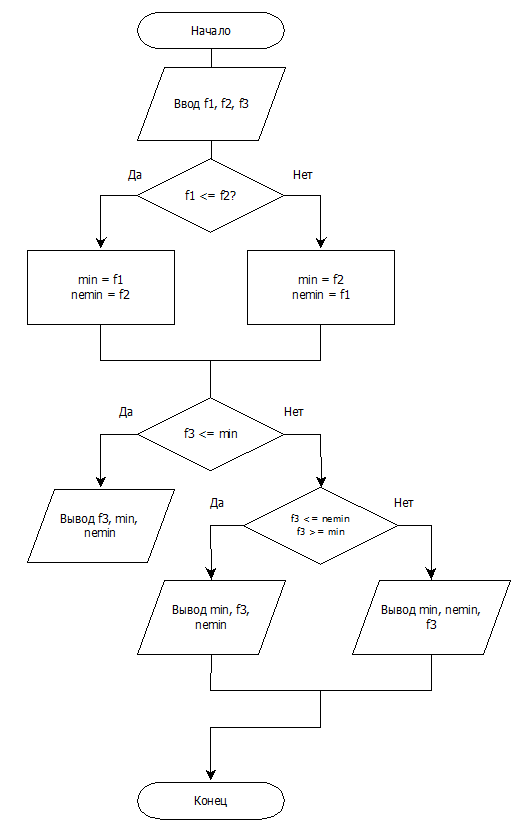


Рис. 7.4.1 Блок-схема к листингу 7.4

**Листинг 7.4(7.4.c)**

/\*\*

Написать программу, моделирующую работу лифта,

находящегося на первом этаже. Допустим, что каждый

заходящий в кабину лифта человек нажимает кнопку с номером этажа.

После этого лифт определяет наиболее быстрый порядок транспортирования пассажиров,

расставляя соответствующим образом этажи, на которых он должен остановиться.

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main(){

int f1, f2, f3, min, nemin;

scanf("%d %d %d", &f1, &f2, &f3);

if (f1 <= f2){

min = f1;

nemin = f2;

}

else {

min = f2;

nemin = f1;

}

if ( f3 <= min)

printf("%d %d %d", f3, min, nemin);

else if (f3 <= nemin && f3 >= min)

printf("%d %d %d", min, f3, nemin);

else printf("%d %d %d", min, nemin, f3);

}

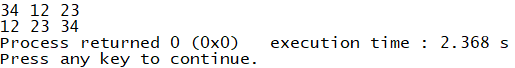


Рис. 7.4.2 Результат работы программы

# **Занятие 8.** **Циклы в С**

**Задание №1:**

Для заданного числа n проверить, является ли оно простым.

**Листинг 8.1(8.1.c)**

/\*\*

Для заданного числа n проверить, является ли оно простым.

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int n, k = 0;

scanf("%d", &n);

int i;

for ( i = 2; i < n; i++) {

if (n % i == 0) {

k++;

}

}

printf(k == 0 ? "1" : "0");

}

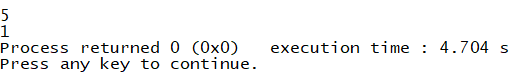


Рис. 8.1.1 Результат работы программы

**Задание №2:**

Факториал.

Для целого числа k(0≤k≤12) посчитать k!.

**Листинг 8.2(8.2.c)**

/\*\*

Факториал.

Для целого числа k(0≤k≤12) посчитать k!.

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int n;

scanf("%d", &n);

int fac = 1;

while (n)

fac \*= n--;

printf("%d", fac);

return 0;

}

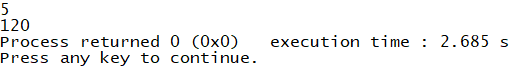


Рис. 8.2.1 Результат работы программы

**Задание №3:**

С клавиатуры вводятся***a*** и ***k***. Возвести число ***a*** в степень ***k***.

**Листинг 8.3(8.3.c)**

/\* 15.11.2018 Выполнил студент группы П2-16, Вологузов В.В.

#include <stdio.h>

int main() {

int a = 0, k = 0;

int res = 1;

scanf("%d %d", &a, &k);

while(k > 0){

if (k % 2 == 1){

res = res \* a;

}

a = a \* a;

k = k / 2;

}

printf("%d\n", res);

}

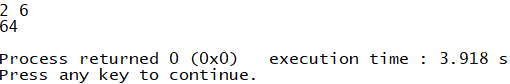
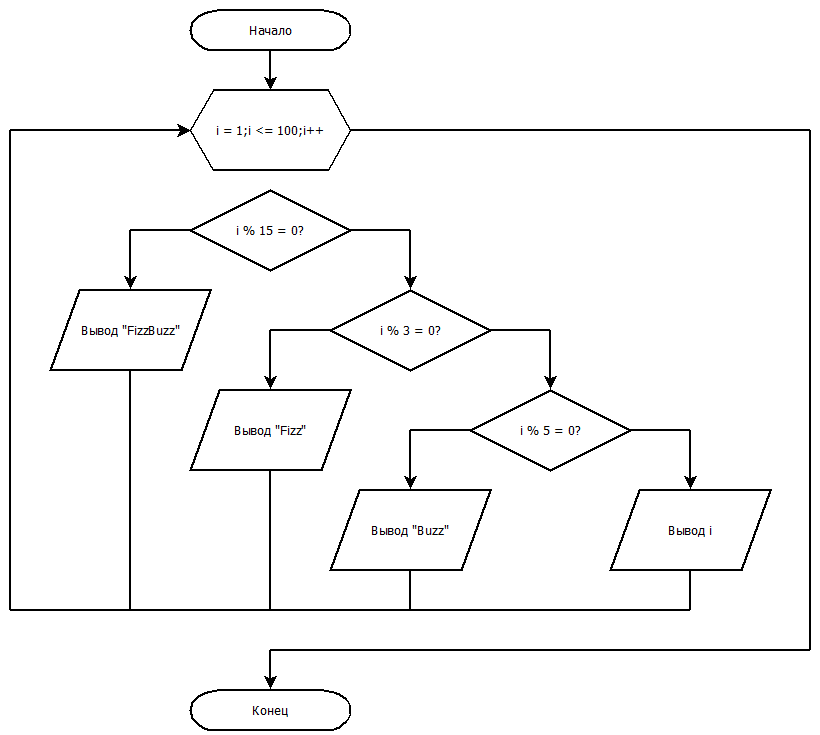


Рис. 8.3.1 Результат работы программы

**Задание №4:**

Напишите программу, которая выводит на экран числа от 1 до 100. При этом вместо чисел, кратных трем, программа должна выводить слово «Fizz», а вместо чисел, кратных пяти — слово «Buzz». Если число кратно и 3, и 5, то программа должна выводить слово «FizzBuzz».

**Блок-схема:**

Рис. 8.4.1 Блок-схема к листингу 8.4

**Листинг 8.4(8.4.c)**

/\*\*

Напишите программу, которая выводит на экран числа от 1 до 100.

При этом вместо чисел, кратных трем, программа должна выводить слово «Fizz»,

а вместо чисел, кратных пяти — слово «Buzz». Если число кратно и 3, и 5,

то программа должна выводить слово «FizzBuzz».

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int i;

for (i = 1;i <= 100;i++)

{

if (i % 15 == 0)

printf("FizzBuzz ");

else if (i % 3 == 0)

printf("Fizz ");

else if (i % 5 == 0)

printf("Buzz ");

else printf("%d ", i);

}

}

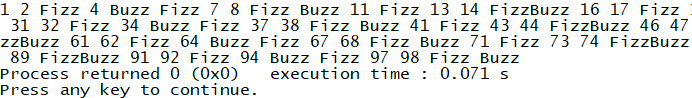


Рис. 8.4.1 Результат работы программы

# **Занятие 9.** **Одномерные массивы в С**

**Задание №1:**

Напишите программу, выводящую на экран все элементы массива, которые меньше последнего элемента и больше первого элемента массива.

**Блок-схема:**

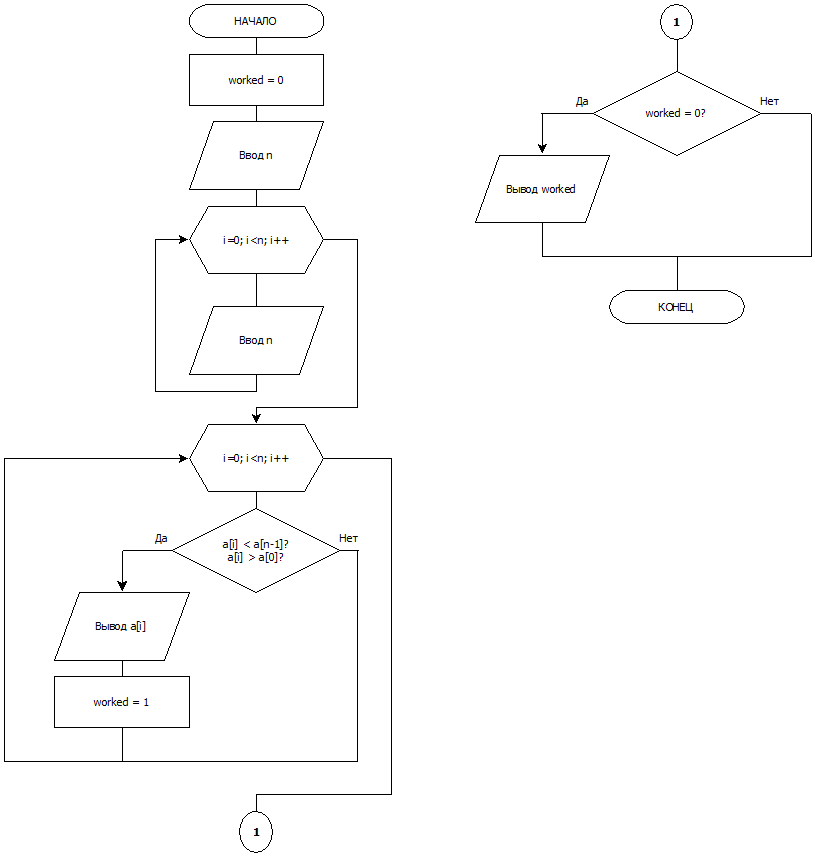


Рис. 9.1.1 Блок-схема к листингу 9.1

**Листинг 9.1(9.1.c)**

/\*\*

Напишите программу, выводящую на экран все элементы массива,

которые меньше последнего элемента и больше первого элемента массива.

Работа выполнена 16.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int n, i;

int worked = 0;

scanf("%d", &n);

int a[n];

for(i = 0; i < n; i++)

scanf("%d", &a[i]);

for(i = 0; i < n; i++)

if(a[i] < a[n - 1] && a[i] > a[0])

{

printf("%d ", a[i]);

worked = 1;

}

if(worked == 0) printf("%d", worked);

return 0;

}

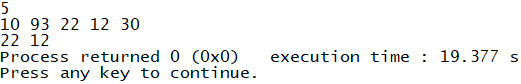


Рис. 9.1.2 Результат работы программы

# **Занятие 10.** **Матрицы в С**

**Задание №1:**

Столбцы

Для заданной матрицы посчитать сумму элементов каждого столбца.

**Блок-схема:**

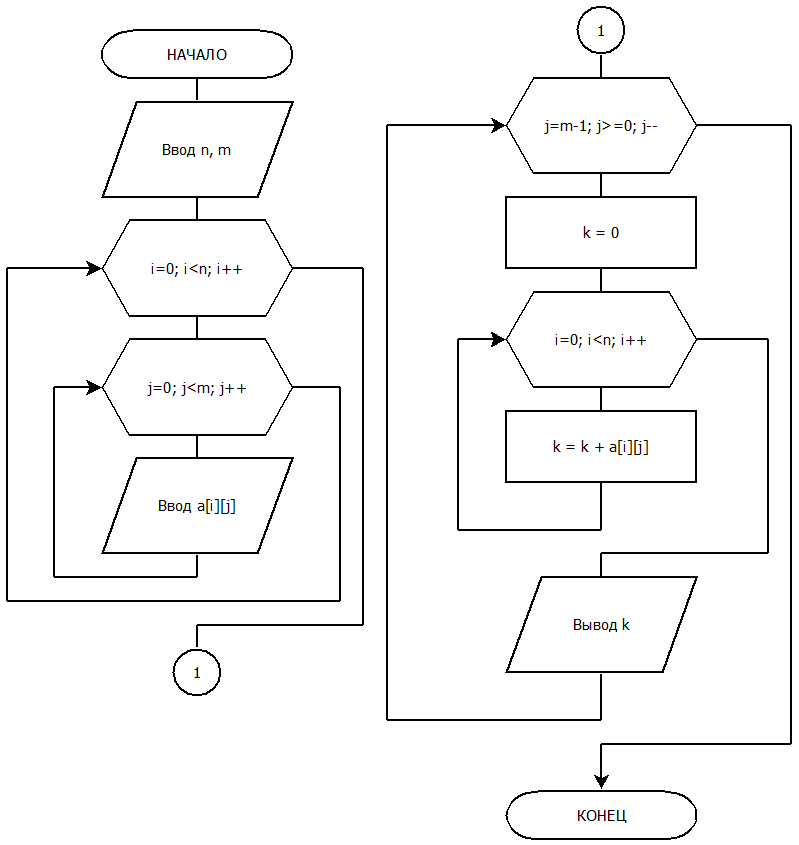


Рис. 10.1.1 Блок-схема к листингу 10.1

**Листинг 10.1(10.1.c)**

/\*\*

Столбцы

Для заданной матрицы посчитать сумму элементов каждого столбца.

Работа выполнена 20.12.18 студентом группы П2-16 Вологузовым В.В.

\*/

#include <stdio.h>

int main() {

int a[100][100],i,j,n,m,k;

scanf("%d %d", &n,&m);

for(i=0;i<n; i++)

for(j=0;j<m; j++)

scanf("%d", &a[i][j]);

printf("\n");

for(j=m-1;j>=0; j--)

{ k=0;

for(i=0;i<n; i++)

k=k+a[i][j];

printf("%d ", k);}

}

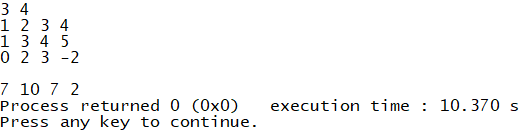


Рис. 10.1.2 Результат работы программы

# **Занятие 11.** **Транспонирование матрицы**

**Задание №1:**

Для заданной матрицы вывести на экран транспонированную матрицу.

**Блок-схема:**

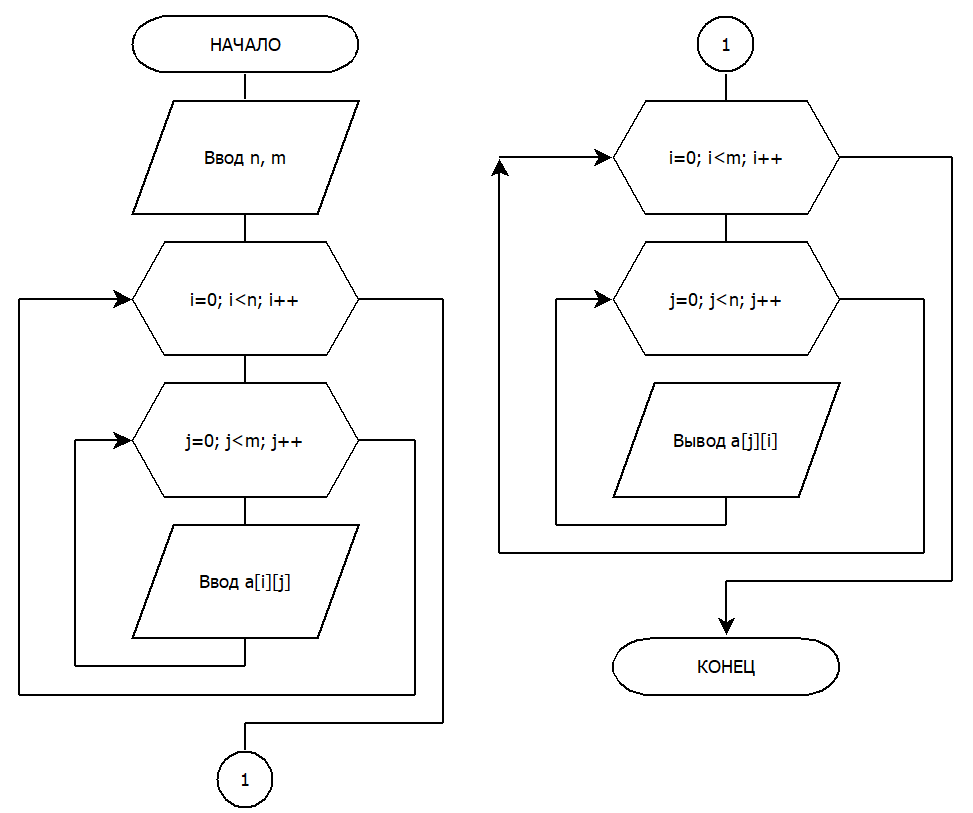


Рис. 11.1.1 Блок-схема к листингу 11.1

**Листинг 11.1(11.1.c)**

/\*\*

Для заданной матрицы вывести на экран транспонированную матрицу.

Работа выполнена 21.12.18 студентом группы П2-16 Дуругяном А.В.

\*/

#include <stdio.h>

int main() {

int n, m;

scanf("%d %d", &n, &m);

int a[n][m];

for(int i = 0; i < n; i++)

for(int j = 0; j < m; j++)

scanf("%d", &a[i][j]);

printf("\n");

for(int i = 0; i < m; i++)

{

for(int j = 0; j < n; j++)

printf("%d ", a[j][i]);

printf("\n");

}

return 0;

}

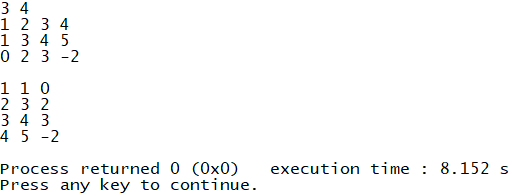


Рис. 11.1.2 Результат работы программы

# **Занятие 12. Создание презентации**

СЛАЙД №1

Тема

СЛАЙД №2

Числа Фибоначчи

СЛАЙД №3

Linux

СЛАЙД №4

Язык Си

СЛАЙД №5

“Спасибо за внимание”

# **Литература и интернет-источники ч.1**

1. Курс «Алгоритмы: теория и практика. Методы».

Автор: Куликов А.С.  
<https://stepik.org/course/217/syllabus>

1. Курс «Основы программирования на Си»

Автор: Кирилл Постнов

<https://stepik.org/course/3078/syllabus>

1. Курс «Основы программирования на языке С++»

Автор: [Computer Science Center (CS центр)](https://stepik.org/users/736914)

<https://stepik.org/course/7/syllabus>

1. Курс «[Основы программирования для Linux](https://stepik.org/course/548)»

Автор: [Computer Science Center (CS центр)](https://stepik.org/users/736914)

<https://stepik.org/course/548/syllabus>

# Дневник УП.01 (2 часть)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№№ пп** | **Наименование разделов, тем, занятий** | **Часы** | **Дата** | **Факт** | **Оценка** |
|  | **Неделя 3. Разработка калькулятора** | **36** |  |  |  |
| 13 | Постановка задачи | 6 | 30.03 |  |  |
| 14 | Разработка спецификаций отдельных компонент | 6 | 01.04 |  |  |
| 15 | Разработка кода программного продукта | 6 | 02.04 |  |  |
| 16 | Отладка программных модулей | 6 | 03.04 |  |  |
| 17 | Тестирование программных модулей.  Оптимизация программного кода модулей | 6 | 04.04 |  |  |
| 18 | Разработка руководства оператора  Создание презентации | 6 | 05.04 |  |  |
|  | **Неделя 4. Разработка графического редактора** | **36** |  |  |  |
| 19 | Постановка задачи | 6 | 06.04 |  |  |
| 20 | Разработка спецификаций отдельных компонент | 6 | 08.04 |  |  |
| 21 | Разработка кода программного продукта | 6 | 09.04 |  |  |
| 22 | Отладка программных модулей | 6 | 10.04 |  |  |
| 23 | Тестирование программных модулей.  Оптимизация программного кода модуля | 6 | 11.04 |  |  |
| 24 | Разработка руководства оператора.  Создание презентации | 6 | 12.04 |  |  |
|  | **Неделя 5.Разработка проекта «Семейный бюджет»** | **36** |  |  |  |
| 25 | Постановка задачи. | 6 | 13.04 |  |  |
| 26 | Добавление основных элементов | 6 | 15.04 |  |  |
| 27 | Настройка подключения к БД | 6 | 16.04 |  |  |
| 28 | Функция выбора конкретной таблицы.  Функция добавления записи | 6 | 17.04 |  |  |
| 29 | Функция удаления записи. | 6 | 18.04 |  |  |
| 30 | Создание SQL-запроса для нахождения определенных расходов | 6 | 19.04 |  |  |
|  | **Неделя 6. Доработка проекта «Семейный бюджет»** | **36** |  |  |  |
| 31 | Вывод результата запроса в XML | 6 | 20.04 |  |  |
| 32 | Вывод результата запроса в Word | 6 | 22.04 |  |  |
| 33 | Вывод результата запроса в Excel | 6 | 23.04 |  |  |
| 34 | Внешний вид программы | 6 | 24.04 |  |  |
| 35 | Подготовка презентации | 6 | 25.04 |  |  |
| 36 | Защита | 6 | 26.04 |  |  |
|  | **Итого за вторую часть практики** | **144** |  |  |  |

# **Занятие 13. Постановка задачи программы “Калькулятор”**

Постановка задачи состоит из двух документов:

1) техническое задание (“путь”)

2) пояснительная записка (“путь”)

1) Краткое содержание документа “Техническое задание”:

В разделе «Введение» указано наименование, краткая характеристика области применения программы (программного изделия).

В разделе «Основания для разработки» указаны документы, на основании которых ведется разработка, наименование и условное обозначение темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программы (программного изделия).

Раздел «Требования к программе» содержит следующие подразделы:

* требования к функциональным характеристикам;
* требования к надежности;
* условия эксплуатации;
* требования к составу и параметрам технических средств;
* требования к информационной и программной совместимости;
* специальные требования.

В разделе «Требования к программной документации» указаны предварительный состав программной документации и специальные требования к ней.

В разделе «Технико-экономические показатели» указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки.

В разделе «Стадии и этапы разработки» установлены необходимые стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

2) Краткое содержание документа “Пояснительная записка”:

В разделе «Введение» указано наименование программы и условное обозначение темы разработки.

В разделе «Назначение и область применения» указано назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

* постановка задачи на разработку программы, с описанием применяемых математических методов и описанием допущений и ограничений, связанных с выбранным математическим материалом;
* описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
* описание и обоснование выбора метода организации входных и выходных данных;
* описание и обоснование выбора состава технических и программных средств на основании проведенных расчетов и анализов.

В разделе «Ожидаемые технико-экономические показатели» указаны технико-экономические показатели, обосновывающие выбранного варианта технического решения, а также, ожидаемые оперативные показатели.

В разделе «Источники, использованные при разработке» указан перечень научно-технических публикаций, нормативно-технических документов и других научно-технических материалов, на которые есть ссылки в основном тексте.

# **Занятие 14. Разработка отдельных спецификаций компонент программы “Калькулятор”**

1) Внешний вид программы “Калькулятор”

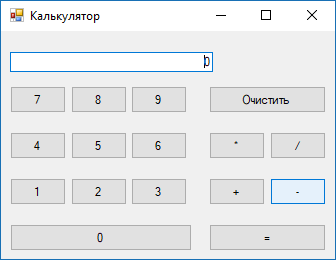


Рис 12. Внешний вид программы “Калькулятор”

2) Интерфейс

В верхней части:

* Поле ввода/вывода;

В нижней части:

* Кнопки с цифрами от “0” до “9”;
* Кнопка полной очистки – “Очистить”;
* Кнопка деления – “ / ”;
* Кнопка вычитания – “ - ”;
* Кнопка сложения – “ + ”;
* Кнопка умножения – “ \* ”;
* Кнопка равенства – “ = ”;

# **Занятие 15. Разработка кода программного продукта программы “Калькулятор”**

**Листинг (Калькулятор.sln/Form1.cs)**

/\*

Программа "Калькулятор"

Работа выполнена студентами группы П2-16 Корзухин А.И. и Дуругяном А.В.

\*/

using System;

using System.Windows.Forms;

namespace Калькулятор

{

public partial class Form1 : Form

{

// Объявление внешних переменных

String Znak = String.Empty; // - знак арифметической операции:

// Признак того, что пользователь вводит новое число:

Boolean Начало\_Ввода = true;

// Первое и второе числа, вводимые пользователем:

Double Число1, Число2;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

this.Text = "Калькулятор";

button1.Text = "1"; button2.Text = "2"; button3.Text = "3";

button4.Text = "4"; button5.Text = "5"; button6.Text = "6";

button7.Text = "7"; button8.Text = "8"; button9.Text = "9";

button10.Text = "0"; button11.Text = "="; button12.Text = "+";

button13.Text = "-"; button14.Text = "\*"; button15.Text = "/";

button16.Text = "Очистить";

textBox1.Text = "0";

textBox1.TextAlign = HorizontalAlignment.Right;

this.button1.Click += new System.EventHandler(this.ЦИФРА);

this.button2.Click += new System.EventHandler(this.ЦИФРА);

this.button3.Click += new System.EventHandler(this.ЦИФРА);

this.button4.Click += new System.EventHandler(this.ЦИФРА);

this.button5.Click += new System.EventHandler(this.ЦИФРА);

this.button6.Click += new System.EventHandler(this.ЦИФРА);

this.button7.Click += new System.EventHandler(this.ЦИФРА);

this.button8.Click += new System.EventHandler(this.ЦИФРА);

this.button9.Click += new System.EventHandler(this.ЦИФРА);

this.button10.Click += new System.EventHandler(this.ЦИФРА);

this.button12.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button13.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button14.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button15.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button11.Click += new System.EventHandler(this.РАВНО);

this.button16.Click += new System.EventHandler(this.ОЧИСТИТЬ);

}

private void ЦИФРА(object sender, EventArgs e)

{

// Обработка события нажатия кнопки-цифры.

Button Кнопка = (Button)sender;

String Digit = Кнопка.Text;

if (Начало\_Ввода == true)

{

textBox1.Text = Digit;

Начало\_Ввода = false; return;

}

if (Начало\_Ввода == false)

textBox1.Text = textBox1.Text + Digit;

}

private void ОПЕРАЦИЯ(object sender, EventArgs e)

{

// Обработка события нажатия кнопки арифметической операции:

Число1 = Double.Parse(textBox1.Text);

Button Кнопка = (Button)sender;

Znak = Кнопка.Text;

Начало\_Ввода = true;

}

private void РАВНО(object sender, EventArgs e)

{

// Обработка события нажатия клавиши "равно"

double Результат = 0;

Число2 = Double.Parse(textBox1.Text);

if (Znak == "+") Результат = Число1 + Число2;

if (Znak == "-") Результат = Число1 - Число2;

if (Znak == "\*") Результат = Число1 \* Число2;

if (Znak == "/") Результат = Число1 / Число2;

Znak = null;

// Отображение результата в текстовом поле:

textBox1.Text = Результат.ToString();

Число1 = Результат; Начало\_Ввода = true;

}

private void ОЧИСТИТЬ(object sender, EventArgs e)

{

// Обработка события нажатия клавиши "Очистить"

textBox1.Text = "0"; Znak = null; Начало\_Ввода = true;

}

}

}

# **Занятие 16. Отладка программных модулей программы “Калькулятор”**

**Листинг (Калькулятор.sln/Form1.cs)**

/\*

Программа "Калькулятор"

Работа выполнена студентами группы П2-16 Корзухин А.И. и Дуругяном А.В.

\*/

using System;

using System.Windows.Forms;

namespace Калькулятор

{

public partial class Form1 : Form

{

// Объявление внешних переменных

String Znak = String.Empty; // - знак арифметической операции:

// Признак того, что пользователь вводит новое число:

Boolean Начало\_Ввода = true;

// Первое и второе числа, вводимые пользователем:

Double Число1, Число2;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

this.Text = "Калькулятор";

button1.Text = "1"; button2.Text = "2"; button3.Text = "3";

button4.Text = "4"; button5.Text = "5"; button6.Text = "6";

button7.Text = "7"; button8.Text = "8"; button9.Text = "9";

button10.Text = "0"; button11.Text = "="; button12.Text = "+";

button13.Text = "-"; button14.Text = "\*"; button15.Text = "/";

button16.Text = "Очистить";

textBox1.Text = "0";

textBox1.TextAlign = HorizontalAlignment.Right;

this.button1.Click += new System.EventHandler(this.ЦИФРА);

this.button2.Click += new System.EventHandler(this.ЦИФРА);

this.button3.Click += new System.EventHandler(this.ЦИФРА);

this.button4.Click += new System.EventHandler(this.ЦИФРА);

this.button5.Click += new System.EventHandler(this.ЦИФРА);

this.button6.Click += new System.EventHandler(this.ЦИФРА);

this.button7.Click += new System.EventHandler(this.ЦИФРА);

this.button8.Click += new System.EventHandler(this.ЦИФРА);

this.button9.Click += new System.EventHandler(this.ЦИФРА);

this.button10.Click += new System.EventHandler(this.ЦИФРА);

this.button12.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button13.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button14.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button15.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button11.Click += new System.EventHandler(this.РАВНО);

this.button16.Click += new System.EventHandler(this.ОЧИСТИТЬ);

}

private void ЦИФРА(object sender, EventArgs e)

{

// Обработка события нажатия кнопки-цифры.

Button Кнопка = (Button)sender;

String Digit = Кнопка.Text;

if (Начало\_Ввода == true)

{

textBox1.Text = Digit;

Начало\_Ввода = false; return;

}

if (Начало\_Ввода == false)

textBox1.Text = textBox1.Text + Digit;

}

private void ОПЕРАЦИЯ(object sender, EventArgs e)

{

// Обработка события нажатия кнопки арифметической операции:

Число1 = Double.Parse(textBox1.Text);

Button Кнопка = (Button)sender;

Znak = Кнопка.Text;

Начало\_Ввода = true;

}

private void РАВНО(object sender, EventArgs e)

{

// Обработка события нажатия клавиши "равно"

double Результат = 0;

Число2 = Double.Parse(textBox1.Text);

if (Znak == "+") Результат = Число1 + Число2;

if (Znak == "-") Результат = Число1 - Число2;

if (Znak == "\*") Результат = Число1 \* Число2;

if (Znak == "/") Результат = Число1 / Число2;

Znak = null;

// Отображение результата в текстовом поле:

textBox1.Text = Результат.ToString();

Число1 = Результат; Начало\_Ввода = true;

}

private void ОЧИСТИТЬ(object sender, EventArgs e)

{

// Обработка события нажатия клавиши "Очистить"

textBox1.Text = "0"; Znak = null; Начало\_Ввода = true;

}

}

}

# **Занятие 17. Тестирование программных модулей. Оптимизация программного кода модуля программы “Калькулятор”**

**Листинг (Калькулятор.sln/Form1.cs)**

/\*

Программа "Калькулятор"

Работа выполнена студентами группы П2-16 Корзухин А.И. и Дуругяном А.В.

\*/

using System;

using System.Windows.Forms;

namespace Калькулятор

{

public partial class Form1 : Form

{

// Объявление внешних переменных

String Znak = String.Empty; // - знак арифметической операции:

// Признак того, что пользователь вводит новое число:

Boolean Начало\_Ввода = true;

// Первое и второе числа, вводимые пользователем:

Double Число1, Число2;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

this.Text = "Калькулятор";

button1.Text = "1"; button2.Text = "2"; button3.Text = "3";

button4.Text = "4"; button5.Text = "5"; button6.Text = "6";

button7.Text = "7"; button8.Text = "8"; button9.Text = "9";

button10.Text = "0"; button11.Text = "="; button12.Text = "+";

button13.Text = "-"; button14.Text = "\*"; button15.Text = "/";

button16.Text = "Очистить";

textBox1.Text = "0";

textBox1.TextAlign = HorizontalAlignment.Right;

this.button1.Click += new System.EventHandler(this.ЦИФРА);

this.button2.Click += new System.EventHandler(this.ЦИФРА);

this.button3.Click += new System.EventHandler(this.ЦИФРА);

this.button4.Click += new System.EventHandler(this.ЦИФРА);

this.button5.Click += new System.EventHandler(this.ЦИФРА);

this.button6.Click += new System.EventHandler(this.ЦИФРА);

this.button7.Click += new System.EventHandler(this.ЦИФРА);

this.button8.Click += new System.EventHandler(this.ЦИФРА);

this.button9.Click += new System.EventHandler(this.ЦИФРА);

this.button10.Click += new System.EventHandler(this.ЦИФРА);

this.button12.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button13.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button14.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button15.Click += new System.EventHandler(this.ОПЕРАЦИЯ);

this.button11.Click += new System.EventHandler(this.РАВНО);

this.button16.Click += new System.EventHandler(this.ОЧИСТИТЬ);

}

private void ЦИФРА(object sender, EventArgs e)

{

// Обработка события нажатия кнопки-цифры.

Button Кнопка = (Button)sender;

String Digit = Кнопка.Text;

if (Начало\_Ввода == true)

{

textBox1.Text = Digit;

Начало\_Ввода = false; return;

}

if (Начало\_Ввода == false)

textBox1.Text = textBox1.Text + Digit;

}

private void ОПЕРАЦИЯ(object sender, EventArgs e)

{

// Обработка события нажатия кнопки арифметической операции:

Число1 = Double.Parse(textBox1.Text);

Button Кнопка = (Button)sender;

Znak = Кнопка.Text;

Начало\_Ввода = true;

}

private void РАВНО(object sender, EventArgs e)

{

// Обработка события нажатия клавиши "равно"

double Результат = 0;

Число2 = Double.Parse(textBox1.Text);

if (Znak == "+") Результат = Число1 + Число2;

if (Znak == "-") Результат = Число1 - Число2;

if (Znak == "\*") Результат = Число1 \* Число2;

if (Znak == "/") Результат = Число1 / Число2;

Znak = null;

// Отображение результата в текстовом поле:

textBox1.Text = Результат.ToString();

Число1 = Результат; Начало\_Ввода = true;

}

private void ОЧИСТИТЬ(object sender, EventArgs e)

{

// Обработка события нажатия клавиши "Очистить"

textBox1.Text = "0"; Znak = null; Начало\_Ввода = true;

}

}

}

# **Занятие 18. Разработка руководства программы “Калькулятор”. Создание презентации**

**Руководство оператора(краткое)**

Функциональное назначение программы – выполнение простейших математических расчетов.

Программа “Калькулятор” имеет следующие функции:

1. функция сложения
2. функция вычитания
3. функция умножения
4. функция деления
5. функция очистки
6. функция равенства

Полное руководство оператора находится на прикрепленном к отчету диске.

**Создание презентации**

СЛАЙД №1

Содержит наименование программы и ее авторов.

СЛАЙД №2

Демонстрирует внешний вид программы

СЛАЙД №3

Показывает функционал программы (часть 1)

СЛАЙД №4

Показывает функционал программы (часть 2)

СЛАЙД №5

Показывает функционал программы (часть 3)

СЛАЙД №6

Показывает функционал программы (часть 4)

СЛАЙД №7

Показывает функционал программы (часть 5)

СЛАЙД №8

Показывает функционал программы (часть 6)

СЛАЙД №9

“Спасибо за внимание”

# **Занятие 19. Постановка задачи “Графический редактор”**

Постановка задачи состоит из двух документов:

1) техническое задание (“путь”)

2) пояснительная записка (“путь”)

1) Краткое содержание документа “Техническое задание”:

В разделе «Введение» указано наименование, краткая характеристика области применения программы (программного изделия).

В разделе «Основания для разработки» указаны документы, на основании которых ведется разработка, наименование и условное обозначение темы разработки.

В разделе «Назначение разработки» указано функциональное и эксплуатационное назначение программы (программного изделия).

Раздел «Требования к программе» содержит следующие подразделы:

* требования к функциональным характеристикам;
* требования к надежности;
* условия эксплуатации;
* требования к составу и параметрам технических средств;
* требования к информационной и программной совместимости;
* специальные требования.

В разделе «Требования к программной документации» указаны предварительный состав программной документации и специальные требования к ней.

В разделе «Технико-экономические показатели» указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки.

В разделе «Стадии и этапы разработки» установлены необходимые стадии разработки, этапы и содержание работ.

В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

2) Краткое содержание документа “Пояснительная записка”:

В разделе «Введение» указано наименование программы и условное обозначение темы разработки.

В разделе «Назначение и область применения» указано назначение программы и краткая характеристика области применения программы.

В разделе «Технические характеристики» содержатся следующие подразделы:

* постановка задачи на разработку программы, с описанием применяемых математических методов и описанием допущений и ограничений, связанных с выбранным математическим материалом;
* описание алгоритма и функционирования программы с обоснованием выбора схемы алгоритма решения задачи и возможные взаимодействия программы с другими программами;
* описание и обоснование выбора метода организации входных и выходных данных;
* описание и обоснование выбора состава технических и программных средств на основании проведенных расчетов и анализов.

В разделе «Ожидаемые технико-экономические показатели» указаны технико-экономические показатели, обосновывающие выбранного варианта технического решения, а также, ожидаемые оперативные показатели.

В разделе «Источники, использованные при разработке» указан перечень научно-технических публикаций, нормативно-технических документов и других научно-технических материалов, на которые есть ссылки в основном тексте.

# **Занятие 20. Разработка спецификаций отдельных компонент программы “Графический редактор”**

Внешний вид программы “Графический редактор”

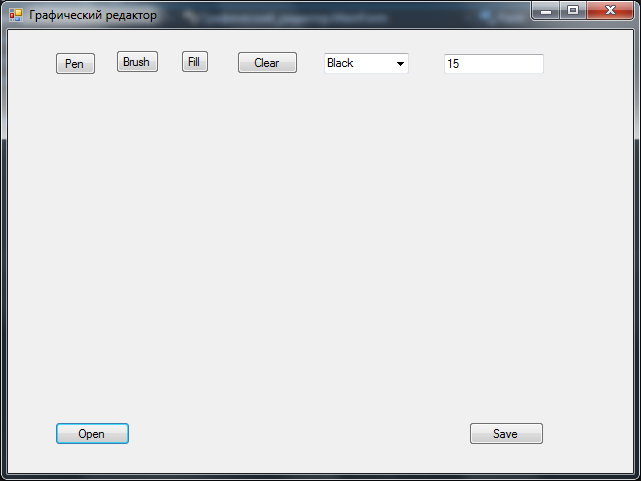


Рис 13. Внешний вид программы “Графический редактор”

ИНТЕРФЕЙС

1) Ручка (Pen):

Позволяет писать в стиле каллиграфической ручки;

2) Кисть (Brush):

Позволяет рисовать в стиле кисти;

3) Заливка (Fill):

Заполняет холст выбранным цветом;

4) Очистка (Clear):

Производит полную очистку холста;

5) Цвет кисти(comboBox):

Позволяет выбрать цвет кисти;

6) Размер кисти(textBox):

Задает размер кисти;

7) Кнопка “Открыть” (Open):

Служит для открытия файл;

8) Кнопка “Сохранить” (Save):

Дает возможность сохранить рисунок;

9) Холст (pictureBox):

Поле для рисования;

# **Занятие 21. Разработка кода программного продукта программы “Графический редактор”**

**Листинг (Графический редактор.sln/Form1.cs)**

/\*

Программа "Графический редактор"

Работа выполнена студентами группы П2-16 Корзухин А.И. и Дуругяном А.В.

\*/

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Графический\_редактор

{

public partial class MainForm : Form

{

string fileName = "";

string w;

bool Paint;

bool Span;

bool flag = false;

public MainForm()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

Canvas.Image = Image.FromFile(openFileDialog1.FileName);

fileName = openFileDialog1.SafeFileName;

}

private void button3\_Click(object sender, EventArgs e)

{

Rectangle rec = Canvas.RectangleToScreen(Canvas.ClientRectangle);

Bitmap bitmap = new Bitmap(Canvas.ClientSize.Width, Canvas.ClientSize.Height);

Graphics g = Graphics.FromImage(bitmap);

g.CopyFromScreen(rec.Location, new Point(0, 0), rec.Size);

saveFileDialog1.FileName = fileName;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

bitmap.Save(saveFileDialog1.FileName);

}

}

private void Form1\_Load(object sender, EventArgs e)

{

Paint = false;

Span = false;

saveFileDialog1.Filter = " Рисунок (\*.jpg)|\*.txt|All files (\*.\*)|\*.\*";

foreach (String Color in Enum.GetNames (typeof(KnownColor)))

if (Color != "Transparent") ColorBox.Items.Add(Color);

ColorBox.Text = "Black";

SizeBox.Text = "15";

}

private void button4\_Click(object sender, EventArgs e)

{

w = PenButton.Text;

Span = true;

}

private void button5\_Click(object sender, EventArgs e)

{

w = BrushButton.Text;

Span = true;

}

private void button6\_Click(object sender, EventArgs e)

{

w = FillButton.Text;

Span = true;

}

private void button7\_Click(object sender, EventArgs e)

{

var Teby = Canvas.CreateGraphics();

Teby.Clear(this.BackColor);

}

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

Paint = true;

if (Paint == true && Span == true && w == "Fill")

{

var color = Color.FromName(ColorBox.Text);

Canvas.BackColor = color;

}

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

if (Paint==true && Span==true && w=="Pen")

{

Graphics g = Canvas.CreateGraphics();

var d = new Pen(Color.FromName(ColorBox.Text), int.Parse(SizeBox.Text));

g.DrawLine(d, e.X, e.Y, e.X+ 1, e.Y+1);

}

else if(Paint == true && Span == true && w == "Brush")

{

Graphics g = Canvas.CreateGraphics();

SolidBrush shadowBrush = new SolidBrush(Color.FromName(ColorBox.Text));

g.FillEllipse(shadowBrush, e.X, e.Y, int.Parse(SizeBox.Text), int.Parse(SizeBox.Text));

}

}

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

Paint = false;

}

}

}

# **Занятие 22. Отладка программных модулей программы “Графический редактор”**

**Листинг (Графический редактор.sln/Form1.cs)**

/\*

Программа "Графический редактор"

Работа выполнена студентами группы П2-16 Корзухин А.И. и Дуругяном А.В.

\*/

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Графический\_редактор

{

public partial class MainForm : Form

{

string fileName = "";

string w;

bool Paint;

bool Span;

bool flag = false;

public MainForm()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

Canvas.Image = Image.FromFile(openFileDialog1.FileName);

fileName = openFileDialog1.SafeFileName;

}

private void button3\_Click(object sender, EventArgs e)

{

Rectangle rec = Canvas.RectangleToScreen(Canvas.ClientRectangle);

Bitmap bitmap = new Bitmap(Canvas.ClientSize.Width, Canvas.ClientSize.Height);

Graphics g = Graphics.FromImage(bitmap);

g.CopyFromScreen(rec.Location, new Point(0, 0), rec.Size);

saveFileDialog1.FileName = fileName;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

bitmap.Save(saveFileDialog1.FileName);

}

}

private void Form1\_Load(object sender, EventArgs e)

{

Paint = false;

Span = false;

saveFileDialog1.Filter = " Рисунок (\*.jpg)|\*.txt|All files (\*.\*)|\*.\*";

foreach (String Color in Enum.GetNames (typeof(KnownColor)))

if (Color != "Transparent") ColorBox.Items.Add(Color);

ColorBox.Text = "Black";

SizeBox.Text = "15";

}

private void button4\_Click(object sender, EventArgs e)

{

w = PenButton.Text;

Span = true;

}

private void button5\_Click(object sender, EventArgs e)

{

w = BrushButton.Text;

Span = true;

}

private void button6\_Click(object sender, EventArgs e)

{

w = FillButton.Text;

Span = true;

}

private void button7\_Click(object sender, EventArgs e)

{

var Teby = Canvas.CreateGraphics();

Teby.Clear(this.BackColor);

}

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

Paint = true;

if (Paint == true && Span == true && w == "Fill")

{

var color = Color.FromName(ColorBox.Text);

Canvas.BackColor = color;

}

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

if (Paint==true && Span==true && w=="Pen")

{

Graphics g = Canvas.CreateGraphics();

var d = new Pen(Color.FromName(ColorBox.Text), int.Parse(SizeBox.Text));

g.DrawLine(d, e.X, e.Y, e.X+ 1, e.Y+1);

}

else if(Paint == true && Span == true && w == "Brush")

{

Graphics g = Canvas.CreateGraphics();

SolidBrush shadowBrush = new SolidBrush(Color.FromName(ColorBox.Text));

g.FillEllipse(shadowBrush, e.X, e.Y, int.Parse(SizeBox.Text), int.Parse(SizeBox.Text));

}

}

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

Paint = false;

}

}

}

# **Занятие 23. Тестирование программных модулей, оптимизация программного кода модуля программы “Графический редактор”**

**Листинг (Графический редактор.sln/Form1.cs)**

/\*

Программа "Графический редактор"

Работа выполнена студентами группы П2-16 Корзухин А.И. и Дуругяном А.В.

\*/

using System;

using System.Drawing;

using System.Windows.Forms;

namespace Графический\_редактор

{

public partial class MainForm : Form

{

string fileName = "";

string w;

bool Paint;

bool Span;

bool flag = false;

public MainForm()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

Canvas.Image = Image.FromFile(openFileDialog1.FileName);

fileName = openFileDialog1.SafeFileName;

}

private void button3\_Click(object sender, EventArgs e)

{

Rectangle rec = Canvas.RectangleToScreen(Canvas.ClientRectangle);

Bitmap bitmap = new Bitmap(Canvas.ClientSize.Width, Canvas.ClientSize.Height);

Graphics g = Graphics.FromImage(bitmap);

g.CopyFromScreen(rec.Location, new Point(0, 0), rec.Size);

saveFileDialog1.FileName = fileName;

if (saveFileDialog1.ShowDialog() == DialogResult.OK)

{

bitmap.Save(saveFileDialog1.FileName);

}

}

private void Form1\_Load(object sender, EventArgs e)

{

Paint = false;

Span = false;

saveFileDialog1.Filter = " Рисунок (\*.jpg)|\*.txt|All files (\*.\*)|\*.\*";

foreach (String Color in Enum.GetNames (typeof(KnownColor)))

if (Color != "Transparent") ColorBox.Items.Add(Color);

ColorBox.Text = "Black";

SizeBox.Text = "15";

}

private void button4\_Click(object sender, EventArgs e)

{

w = PenButton.Text;

Span = true;

}

private void button5\_Click(object sender, EventArgs e)

{

w = BrushButton.Text;

Span = true;

}

private void button6\_Click(object sender, EventArgs e)

{

w = FillButton.Text;

Span = true;

}

private void button7\_Click(object sender, EventArgs e)

{

var Teby = Canvas.CreateGraphics();

Teby.Clear(this.BackColor);

}

private void pictureBox1\_MouseDown(object sender, MouseEventArgs e)

{

Paint = true;

if (Paint == true && Span == true && w == "Fill")

{

var color = Color.FromName(ColorBox.Text);

Canvas.BackColor = color;

}

}

private void pictureBox1\_MouseMove(object sender, MouseEventArgs e)

{

if (Paint==true && Span==true && w=="Pen")

{

Graphics g = Canvas.CreateGraphics();

var d = new Pen(Color.FromName(ColorBox.Text), int.Parse(SizeBox.Text));

g.DrawLine(d, e.X, e.Y, e.X+ 1, e.Y+1);

}

else if(Paint == true && Span == true && w == "Brush")

{

Graphics g = Canvas.CreateGraphics();

SolidBrush shadowBrush = new SolidBrush(Color.FromName(ColorBox.Text));

g.FillEllipse(shadowBrush, e.X, e.Y, int.Parse(SizeBox.Text), int.Parse(SizeBox.Text));

}

}

private void pictureBox1\_MouseUp(object sender, MouseEventArgs e)

{

Paint = false;

}

}

}

# **Занятие 24. Разработка руководства оператора программы “Графический редактор”. Создание презентации**

**Руководство оператора(краткое)**

Функциональным назначением программы являетсясоздание, просматривание, обрабатывание и редактирование цифровых изображения

Программа “Графический редактор” имеет следующие функции:

1. функция открытия файла
2. функция сохранения файла
3. функция работы с кистью
4. функция работы с ручкой
5. функция работы с заливкой
6. функция очистки

Полное руководство оператора находится на прикрепленном к отчету диске.

**Создание презентации**

СЛАЙД №1

Содержит наименование программы и ее авторов.

СЛАЙД №2

Демонстрирует внешний вид программы

СЛАЙД №3

Показывает функционал программы (часть 1)

СЛАЙД №4

Показывает функционал программы (часть 2)

СЛАЙД №5

“Спасибо за внимание”

# **Занятие 25. Постановка задачи проекта “Семейный бюджет”**

Семейный бюджет — это план доходов и расходов семьи на определенный промежуток времени. В нем сочетаются как фактические данные за прошедшие периоды, так и запланированные суммы доходов и расходов. Он позволяет семье контролировать движение денежных средств.

Программа “Семейный бюджет” служит для подсчета доходов и расходов семьи за месяц.

# **Занятие 26. Добавление основных элементов в проект “Семейный бюджет”**

Добавление основных элементов на форму Windows Forms

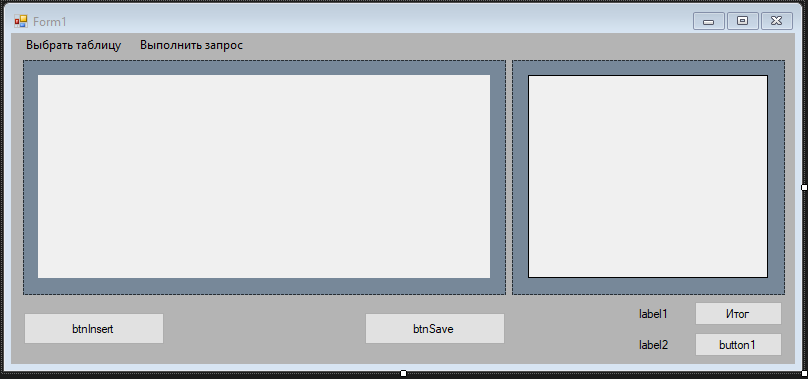


Рис. 14 Основные элементы программы “Семейный бюджет”

# **Занятие 27. Настройка подключения к базе данных в проекте “Семейный бюджет”**

**Листинг (DBComplexProgramm.sln/Form1.cs)**

// Фрагмент кода программы “Семейный бюджет”, отвечающий за подключение к БД

using System;

using System.Data;

using System.Data.OleDb;

using System.Windows.Forms;

namespace DBComplexProgramm

{

public partial class Form1 : Form

{

// Объявляем глобальные переменные

DataSet наборДанных;

OleDbDataAdapter адаптер;

OleDbConnection подключение;

OleDbCommand команда;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

string путь = System.IO.Path.Combine(System.Windows.Forms.Application.StartupPath, "Семейный бюджет2.mdb");

// Создаём подключение

подключение = new OleDbConnection(

"Provider=Microsoft.ACE.OLEDB.12.0;" +

"Data Source=" + путь);

наборДанных = new DataSet();

команда = new OleDbCommand();

if (подключение.State == ConnectionState.Closed)

подключение.Open();

наборДанныхДоходы = new DataSet();

наборДанныхРасходы = new DataSet();

адаптерДоходы = new OleDbDataAdapter(

"Select \* From [Доходы]", подключение);

адаптерДоходы.Fill(наборДанныхДоходы, "Доходы");

dataGridViewДоходы.DataSource = наборДанныхДоходы;

dataGridViewДоходы.DataMember = "Доходы";

адаптерРасходы = new OleDbDataAdapter(

"Select \* From [Расходы]", подключение);

адаптерРасходы.Fill(наборДанныхРасходы, "Расходы");

dataGridViewРасходы.DataSource = наборДанныхРасходы;

dataGridViewРасходы.DataMember = "Расходы";

подключение.Close();

}

# **Занятие 28. Функция выбора конкретной таблицы, функция добавления записи в проекте “Семейный бюджет”**

**Листинг (DBComplexProgramm.sln/Form1.cs)**

// Фрагмент кода программы “Семейный бюджет”

// Загрузка выбранной таблицы

private void ЗагрузитьТаблицу(object sender, EventArgs e)

{

// Получаем название таблицы

назвТабл = Convert.ToString(((ToolStripMenuItem)sender).Text);

подключение.Open();

// Передаём данные в адаптер

адаптер = new OleDbDataAdapter(

"Select \* From [" + назвТабл + "]", подключение);

ОбновитьТаблицу();

подключение.Close();

}

// Добавление новой записи

private void btnInsert\_Click(object sender, EventArgs e)

{

//if (подключение.State == ConnectionState.Closed)

//return;

подключение.Open();

// Добавление записи в соответствии с выбранной таблицей

string rec1 = (dataGridView1.Rows.Count - 1).ToString();

string rec2 = dataGridView1.Rows[dataGridView1.Rows.Count - 2].Cells[1].Value.ToString();

string rec3 = dataGridView1.Rows[dataGridView1.Rows.Count - 2].Cells[2].Value.ToString();

# **Занятие 29. Функция удаления записи в проекте “Семейный бюджет”**

**Листинг (DBComplexProgramm.sln/Form1.cs)**

// Фрагмент кода программы “Семейный бюджет”

// Удаление выбранной записи

private void dataGridView1\_RowsRemoved(object sender, DataGridViewRowsRemovedEventArgs e)

{

if (dataGridView1.RowCount == 0)

return;

MessageBox.Show("Delete");

if (подключение.State == ConnectionState.Closed)

подключение.Open();

// Текст комманды и её выполнение

команда.CommandText = "Delete \* From [" + назвТабл + "] " +

"Where [Номер] = " + curRecord;

команда.Connection = подключение;

команда.ExecuteNonQuery();

// Обновляем dataGrid

dataGridView1.DataSource = наборДанных;

подключение.Close();

}

# **Занятие 30. Создание SQL-запроса в проекте “Семейный бюджет”**

**Листинг (DBComplexProgramm.sln/Form1.cs)**

// Фрагмент кода программы “Семейный бюджет”

// SQL-запрос

private void выполнитьЗапросToolStripMenuItem\_Click(object sender, EventArgs e)

{

dataGridView2.Visible = true;

saveWord.Visible = true;

panel2.Visible = true;

saveExcel.Visible = true;

labelWord.Visible = true;

labelExcel.Visible = true;

int a1 = 0;

a1++;

MessageBox.Show("" + (char)a1 + "");

подключение.Open();

var командаЗапрос = new OleDbCommand(

"Select \* From [Расходы] Where [Цена] > 0", подключение);

var адаптерЗапрос = new OleDbDataAdapter(командаЗапрос);

var наборДанныхЗапрос = new DataSet();

адаптерЗапрос.Fill(наборДанныхЗапрос, "Сделки");

dataGridView2.DataSource = наборДанныхЗапрос;

dataGridView2.DataMember = "Сделки";

подключение.Close();

}

# **Занятие 31. Вывод результата запроса в XML**

**Листинг (DBComplexProgramm.sln/Form1.cs)**

// Фрагмент кода программы “Семейный бюджет”

//Вывод в XML

private void выполнитьЗапросToolStripMenuItem\_Click(object sender, EventArgs e)

{

int a1 = 0;

a1++;

MessageBox.Show("" + (char)a1 + "");

подключение.Open();

var командаЗапрос = new OleDbCommand(

"Select \* From [Расходы] Where [Цена] > 0", подключение);

var адаптерЗапрос = new OleDbDataAdapter(командаЗапрос);

var наборДанныхЗапрос = new DataSet();

адаптерЗапрос.Fill(наборДанныхЗапрос, "Сделки");

dataGridView2.DataSource = наборДанныхЗапрос;

dataGridView2.DataMember = "Сделки";

try {

string fileName = @"G:\data.xml";

наборДанных = (DataSet)dataGridView1.DataSource;

наборДанных.WriteXml(fileName);

}

catch {

MessageBox.Show("Файл уже существует");

};

подключение.Close();

}

# **Занятие 32. Вывод результата запроса в Word**

**Листинг (DBComplexProgramm.sln/Form1.cs)**

// Фрагмент кода программы “Семейный бюджет”

//Вывод в Word

private void saveWord\_Click(object sender, EventArgs e)

{

string rec1 = (dataGridView2.Rows.Count - 1).ToString();

//MessageBox.Show(dataGridView1.Rows[dataGridView1.Rows.Count-2].Cells[1].Value.ToString());

string rec2 = dataGridView2.Rows[dataGridView2.Rows.Count - 2].Cells[1].Value.ToString();

string rec3 = dataGridView2.Rows[dataGridView2.Rows.Count - 2].Cells[2].Value.ToString();

string rec4 = dataGridView2.Rows[dataGridView2.Rows.Count - 2].Cells[3].Value.ToString();

// Создаем новый экземпляр класса Word.Application:

var Word1 = new Microsoft.Office.Interop.

Word.Application();

Word1.Visible = true;

// Открываем новый документ MS Word:

var Документ = Word1.Documents.Add();

//Word1.ActiveDocument.Paragraphs.Add();

Word1.ActiveDocument.Paragraphs[1].Range.ParagraphFormat.Alignment = Microsoft.Office.Interop.Word.WdParagraphAlignment.wdAlignParagraphCenter;

var t3 = Microsoft.Office.Interop.Word.WdUnits.wdLine;

// Вводим текст в документ MS WORD с текущей позиции:

Word1.Selection.TypeText("ИТОГИ СЕМЕЙНОГО БЮДЖЕТА ЗА МЕСЯЦ");

var w1 = dataGridViewРасходы.Rows.Count - 1;

var h1 = dataGridViewРасходы.Columns.Count;

var w2 = dataGridViewДоходы.Rows.Count - 1;

var h2 = dataGridViewДоходы.Columns.Count;

Word1.ActiveDocument.Paragraphs[1].Range.Font.Name = "Century Gothic";

Word1.ActiveDocument.Paragraphs[1].Range.Font.Size = 15;

Word1.Selection.TypeParagraph();

//Word1.Selection.MoveDown(t3, 10, null);

// Параметр, указывающий покаывать ли границы ячеек:

var t1 = Microsoft.Office.Interop.

Word.WdDefaultTableBehavior.wdWord9TableBehavior;

// Параметр, указывающий будет ли приложение Word автоматически

// изменять размер ячеек в таблице для подгонки содержимого:

var t2 = Microsoft.Office.Interop.

Word.WdAutoFitBehavior.wdAutoFitContent;

Word1.Selection.TypeParagraph();

//Word1.ActiveDocument.Paragraphs[2].Range.Text = "ДОХОДЫ";

Word1.Selection.TypeText("ДОХОДЫ");

// создание таблицы

Word1.ActiveDocument.Tables.Add(Word1.Selection.Range,

w2, h2, t1, t2);

Word1.ActiveDocument.Tables[1].Range.Rows.Alignment = Microsoft.Office.Interop.Word.WdRowAlignment.wdAlignRowCenter;

var sumdoh = 0;

// Заполнять ячейки таблицы можно так:

for (int i = 0; i < w2; i++)

{

for (int j = 0; j < h2; j++)

{

Word1.ActiveDocument.Tables[1].Cell(i + 1, j + 1). Range.InsertAfter(dataGridViewДоходы.Rows[i].Cells[j].Value.ToString());

if (j == h2 - 1)

sumdoh = sumdoh + int.Parse(dataGridViewДоходы.Rows[i].Cells[j].Value.ToString());

}

}

Word1.Selection.MoveDown(t3, 10, null);

//Word1.Selection.Paragraphs.SpaceBefore = 10;

Word1.Selection.TypeParagraph();

Word1.Selection.TypeText("РАСХОДЫ");

//Word1.Selection.MoveRight(t34, w1, null);

Word1.ActiveDocument.Tables.Add(Word1.Selection.Range,

w1, h1, t1, t2);

var sumras = 0;

Word1.ActiveDocument.Tables[2].Range.Rows.Alignment = Microsoft.Office.Interop.Word.WdRowAlignment.wdAlignRowCenter;

for (int i = 0; i < w1; i++)

{

for (int j = 0; j < h1; j++)

{

Word1.ActiveDocument.Tables[2].Cell(i + 1, j + 1).

Range.InsertAfter(dataGridViewРасходы.Rows[i].Cells[j].Value.ToString());

if (j == h1 - 1)

sumras = sumras + int.Parse(dataGridViewРасходы.Rows[i].Cells[j].Value.ToString());

}

}

var sum = sumdoh - sumras;

Word1.Selection.MoveDown(t3, 10, null);

Word1.Selection.TypeParagraph();

//Создание и настройка итоговой таблицы

Word1.ActiveDocument.Tables.Add(Word1.Selection.Range,

1, 2, t1, t2);

Word1.ActiveDocument.Tables[3].Range.Rows.Alignment = Microsoft.Office.Interop.Word.WdRowAlignment.wdAlignRowCenter;

Word1.ActiveDocument.Tables[3].Range.Borders.Enable = 0;

Word1.ActiveDocument.Tables[3].Range.Cells.Borders.Enable = 0;

Word1.ActiveDocument.Tables[3].Cell(1, 1).

Range.InsertAfter("Итог:");

if (sumdoh > sumras)

Word1.ActiveDocument.Tables[3].Cell(1, 2).

Range.InsertAfter("+" + sum + "");

else if (sumdoh < sumras)

Word1.ActiveDocument.Tables[3].Cell(1, 2).

Range.InsertAfter("-" + sum + "");

else

Word1.ActiveDocument.Tables[3].Cell(1, 2).

Range.InsertAfter("" + sum + "");

}

# **Занятие 33. Вывод результата запроса в Excel**

**Листинг (DBComplexProgramm.sln/Form1.cs)**

// Фрагмент кода программы “Семейный бюджет”

//Вывод в Excel

private void saveExcel\_Click(object sender, EventArgs e)

{

var Excel = new Microsoft.Office.Interop.Excel.Application();

Excel.Visible = true;

var t = Type.Missing;

var Книга = Excel.Workbooks.Add(t);

var Листы = Книга.Worksheets;

Worksheet Лист = Листы.Item[1];

var w1 = dataGridViewРасходы.Rows.Count - 1;

var h1 = dataGridViewРасходы.Columns.Count;

var w2 = dataGridViewДоходы.Rows.Count - 1;

var h2 = dataGridViewДоходы.Columns.Count;

int ord = 64;

for (int i = 1; i <= w1; i++)

{

for (int j = 1; j <= h1; j++)

{

ord++;

Лист.Range["" + (char)ord + "" + i, t].Value2 = dataGridViewРасходы.Rows[i-1].Cells[j-1].Value.ToString();

}

ord = 64;

}

}

# **Занятие 34. Внешний вид проекта “Семейный бюджет”**

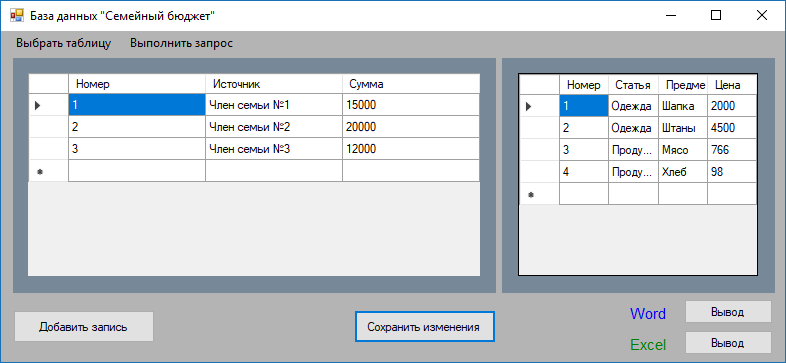


Рис. 15 Внешний вид программы “Семейный бюджет”

# **Занятие 35. Подготовка презентации**

СЛАЙД №1

Тема

СЛАЙД №2

Неделя 3. Программа “Калькулятор”

СЛАЙД №3

Неделя 4. Программа “Графический редактор”

СЛАЙД №4

Неделя 5. Программа “Семейный бюджет”

СЛАЙД №5

Неделя 6. Доработка программы “Семейный бюджет”

СЛАЙД №6

“Спасибо за внимание”

# **Занятие 36. Защита**

Краткий доклад

На **3-й неделе** практики была сделана программа “Калькулятор”, предназначенная для выполнения простейших математических операций и имеющая следующие функции:

1. функция сложения

2. функция вычитания

3. функция умножения

4. функция деления

5. функция очистки

6. функция равенства

На **4-й неделе** практики была сделана программа “Графический редактор”, функциональным назначением которой являетсясоздание, просматривание, обрабатывание и редактирование цифровых изображения

Программа “Графический редактор” имеет следующие функции:

1. функция открытия файла

2. функция сохранения файла

3. функция работы с кистью

4. функция работы с ручкой

5. функция работы с заливкой

6. функция очистки

На **5-й** и **6-й неделях** практики был сделан проект “Семейный бюджет”, служащий для контроля месячного бюджета семьи, которая имеет следующие функции:

1) подключение к БД

2) создание SQL-запроса

3) добавление записи в БД

4) изменение записей в БД

5) вывод результата в Word

6) вывод результата в Excel

# **Литература и интернет-источники ч.2**

1. Книга «Visual C# 2012 на примерах».

Автор: Зиборов В.В.