Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Сибирский государственный университет телекоммуникаций и информатики»

(СибГУТИ)

Институт информатики и вычислительной техники

09.03.01 "Информатика и вычислительная техника"

профиль "Программное обеспечение средств вычислительной техники и автоматизированных систем"

**Контрольная работа**

по дисциплине «Программирование»

Выполнил:

Студент гр. ИП-313 \_\_\_\_\_\_\_\_\_\_\_\_\_/Широков К. В./

ФИО студента

« » декабря 2023 г.

Проверил:

Старший преподаватель

кафедры ПМиК \_\_\_\_\_\_\_\_\_\_\_\_\_/Агалаков А. А./

ФИО преподавателя

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 г. Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Новосибирск 2023

**Оглавление**

Задание 3

Теория 3

Ход работы 4

Демонстрация работы 5

Вывод 7

Литература 7

Приложение 8

# Задание:

1. Сформировать на экране меню, состоящее из следующих пунктов:

|  |
| --- |
| 1. Свободное движение объекта |
| 2. Движение объекта с помощью клавиатуры |
| 3. Выход из программы |

2. В соответствии с вариантом выбрать тип объекта и движения и реализовать пункты меню в виде подпрограмм.

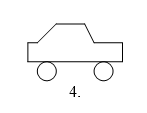
Вариант свободного движения объектов.

7.Движение со сменой направления на границах экрана с диагонального на горизонтальное и наоборот.

Варианты движения объектов с помощью клавиатуры.

4.Управление клавишами с символами s, d, e, x.

Виды графических объектов.



**Теория:**

Для выполнения контрольной работы были реализованы следующие функции:

1. initwindow(x,y)
2. gotoxy()
3. settextcolor()
4. cleardevice()
5. delay()
6. kbhit()
7. getch()
8. closegraph()
9. setlocale()
10. printf()

1)Функция initwindow открывает дополнительное окно, в котором можно рисовать. В скобках нужно указать ширину и высоту этого окна в пикселях.

2) Функция gotoxy помещает курсор текстового экрана в точку с координатами х,у. Если хотя бы одна из координат недействительна, то никаких действий не выполняется.

3) Функция settextcolor() устанавливает цвет текста.

4) cleardevice() - функция очистки окна от предыдущих отрисованных объектов.

5) delay() - функция задержки выполнения программы на некоторое миллисекунд.

6) kbhit() - функция, которая проверяет, была ли нажата клавиша на клавиатуре.

7) getch() - функция, которая считывает нажатую клавишу с клавиатуры.

8) closegraph() - функция, которая закрывает окно графического вывода и завершает программу.

9) setlocale() - функция, которая устанавливает локаль для корректного отображения символов

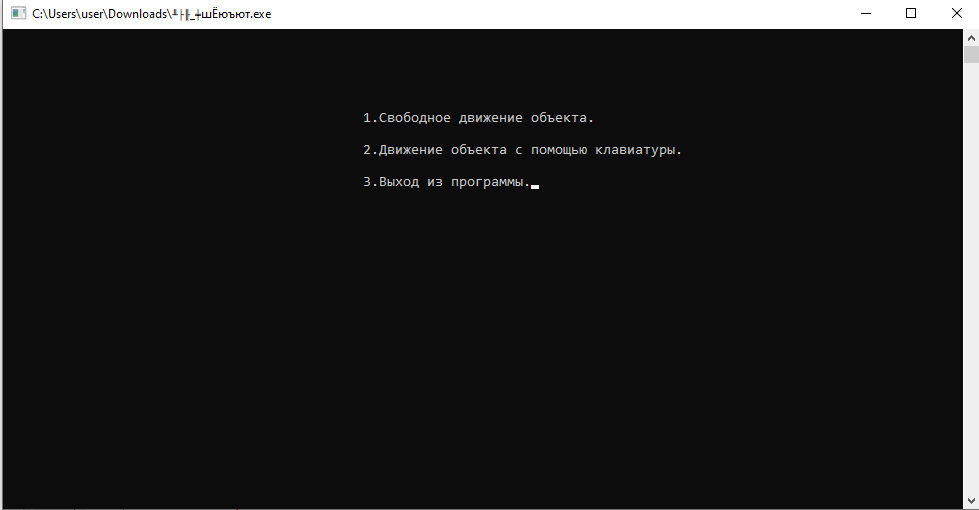
10) printf() - функция для вывода форматированного текста на консольный экран.

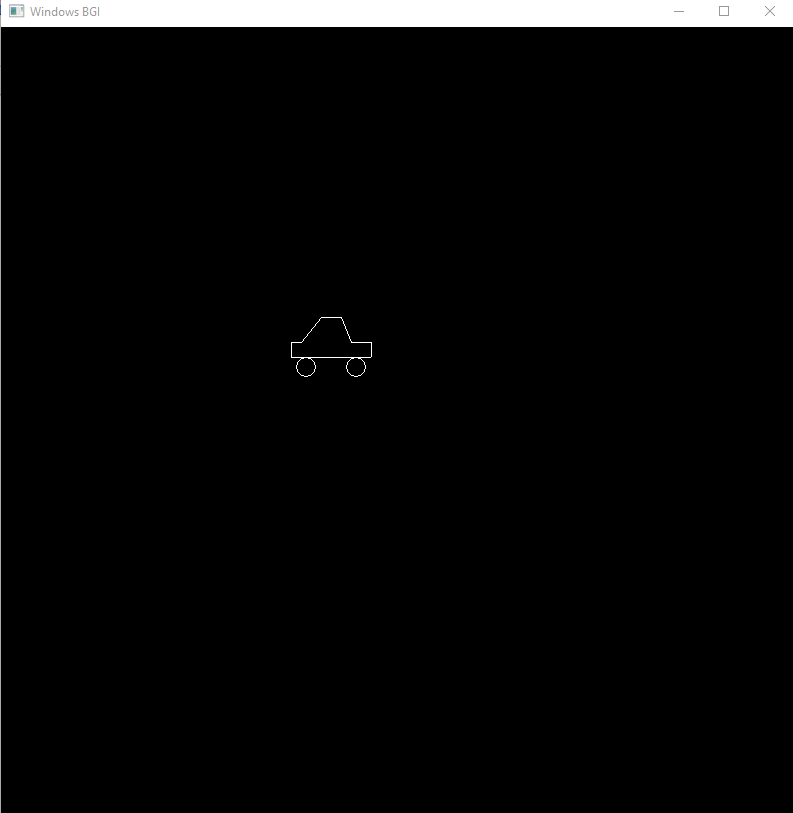
**Ход работы:**

1. На экран выводится меню с выбором действия из трех пунктов.
2. При выборе единицы программа открывает графическое окно с объектом, который свободно двигается в соответствии с вариантом.
3. При выборе двойки программа открывает графическое окно с объектом, который двигается с помощью клавиатуры.
4. При выборе тройки происходит выход из программы.

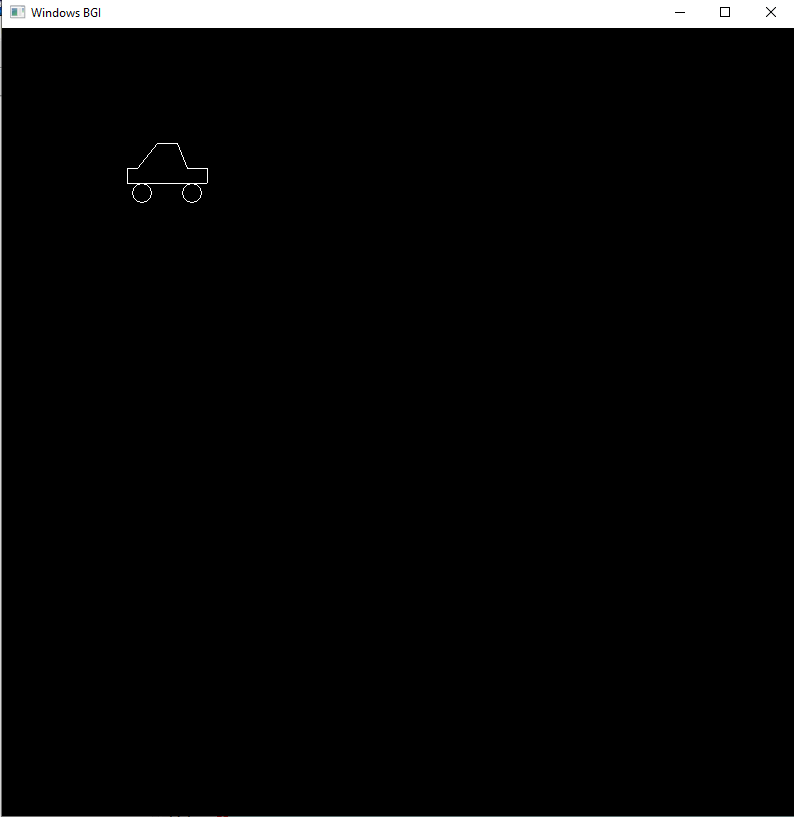
**Демонстрация работы:**

1) Открывается окно с меню.

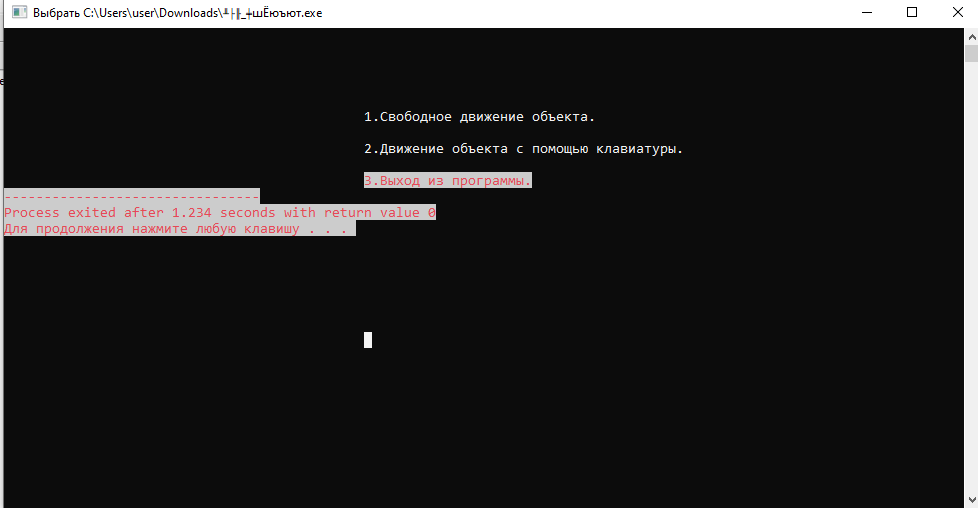
 2) После выбора пункта 1, открывается окно, в котором объект перемещается без нажатия клавиш (по варианту движения 1)



3) Свободное движение, после выбора пункта 2, открывается окно, в котором объект перемещается с помощью нажатия клавиш (по варианту движения №2). Нажимая клавиши (e, s, d, x), объект перемещается в окне.



4)При выборе 3 пункта происходит выход из программы.



**Вывод:**

При выполнении данной работы были изучены базовые графические функции. Также, была реализована программа, в которой пользователь может выбрать ход программы, в которой описаны функции для разных видов движения объекта.

**Литература:**

1. Герберт Шилдт. С++ для начинающих. Шаг за шагом

2. Язык программирования C (2-е издание), Брайан В. Керниган и Деннис М. Ричи

3. Лекции преподавателя Перцева И. В.

4. Методические материалы из лабораторных работ по программированию.

**Приложение**

**//GNU coding standards**

**#include <stdio.h>**

**#include <graphics.h>**

**#include <locale.h>**

**#include <conio.h>**

**#include "altconio.h"**

**void draw(int x, int y) { //рисуем машину**

**line(x, y + 40, x + 80, y + 40);**

**circle(x + 15, y + 50, 10);**

**circle(x + 65, y + 50, 10);**

**line(x, y + 40, x, y + 25);**

**line(x, y + 25, x + 10, y + 25);**

**line(x + 10, y + 25, x + 30, y);**

**line(x + 30, y, x + 50, y);**

**line(x + 50, y, x + 60, y + 25);**

**line(x + 60, y + 25, x + 80, y + 25);**

**line(x + 80, y + 25, x + 80, y + 40);**

**}**

**void freemove() { //вариант движения 1(свободное движение)**

**initwindow(800, 800);**

**setfillstyle(1, 0);**

**int x = 0, y = 0;**

**int derect\_x = 5, derect\_y = 5;**

**bool is\_horizontal = false;**

**int prev\_x, prev\_y;**

**while(1) {**

**cleardevice();**

**draw(x, y);**

**delay(20);**

**prev\_x = x;**

**prev\_y = y;**

**if (is\_horizontal) { //проверка движения (горизонтальное или диагональное)**

**x += derect\_x;**

**}**

**else {**

**x += derect\_x;**

**y += derect\_y;**

**}**

**if ((x > 710 || x < 0) && x != prev\_x) { //проверяем достигнута ли граница**

**derect\_x \*= -1;**

**is\_horizontal = !is\_horizontal;**

**}**

**if ((y > 725 || y < 0) && y != prev\_y) {**

**derect\_y \*= -1;**

**is\_horizontal = !is\_horizontal;**

**}**

**if (kbhit()) { //закрытие окна если нажат esc**

**char ch = getch();**

**if (ch == 27)**

**break;**

**}**

**}**

**closegraph();**

**}**

**void withkey() { //вариант движения 2(по клавишам)**

**initwindow(800, 800);**

**setfillstyle(1, 0);**

**int x = 0, y = 0;**

**char ch;**

**while(1) {**

**cleardevice();**

**draw(x, y);**

**delay(10);**

**if (kbhit()) {**

**ch = getch();**

**if (ch == 27)**

**break;**

**switch(ch) { //изменения положения в зависимости от того какая клавиша нажата**

**case 'e': //вверх**

**y -= 5;**

**break;**

**case 'd': //вперед**

**x += 5;**

**break;**

**case 's': //назад**

**x -= 5;**

**break;**

**case 'x': //вниз**

**y += 5;**

**break;**

**}**

**if (x > 710) { //проверка достижения границ**

**x = 710;**

**}**

**else if (x < 0) {**

**x = 0;**

**}**

**if (y > 725) {**

**y = 725;**

**}**

**else if (y < 0) {**

**y = 0;**

**}**

**}**

**}**

**closegraph();**

**}**

**int main() {**

**setlocale(LC\_ALL, "Rus");**

**int x, y, i, ch;**

**for(i = 1; i <= 3; i++) { //выводим меню**

**gotoxy(45, 3 + i \* 2); // формируем отступы пунктов**

**switch(i) { // выводим пункты в консоль**

**case 1:**

**printf("1.Свободное движение объекта.");**

**break;**

**case 2:**

**printf("2.Движение объекта с помощью клавиатуры.");**

**break;**

**case 3:**

**printf("3.Выход из программы.");**

**break;**

**}**

**}**

**do { //Основной цикл обработки меню**

**ch = \_getch();**

**gotoxy(45, 3 + i \* 2); //перемещаем курсор**

**settextcolor(15, 0); //дилаем заливку текста**

**switch(i) {**

**case 1:**

**printf("1.Свободное движение объекта.");**

**break;**

**case 2:**

**printf("2.Движение объекта с помощью клавиатуры.");**

**break;**

**case 3:**

**printf("3.Выход из программы.");**

**break;**

**}**

**if(ch == 72) //вниз по меню**

**i--;**

**if(ch == 80) //вверх по меню**

**i++;**

**if(i > 3) //зацикливание меню**

**i = 1;**

**if(i < 1) //зацикливание меню**

**i = 3;**

**gotoxy(45, 3 + i \* 2); //перемещаем курсор**

**settextcolor(12, 7); //дилаем заливку текста**

**switch(i) {**

**case 1:**

**printf("1.Свободное движение объекта.");**

**break;**

**case 2:**

**printf("2.Движение объекта с помощью клавиатуры.");**

**break;**

**case 3:**

**printf("3.Выход из программы.");**

**break;**

**}**

**} while(ch != 13); //пока не нажат enter**

**switch(i) { //выполение программы в завимости какой пункт выбран**

**case 1: {**

**freemove();**

**break;**

**}**

**case 2: {**

**withkey();**

**break;**

**}**

**case 3: {**

**return 0;**

**break;**

**}**

**}**

**}**