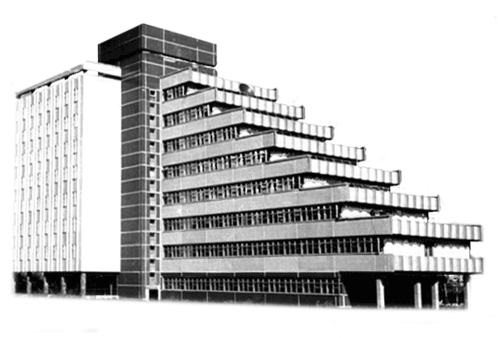
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное агентство по образованию

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Учебный Центр Информационных Технологий «Информатика»



Контрольная работа

по дисциплине «Информатика и программирование (I часть)»

Направление подготовки: 230105 - «Программное обеспечение вычислительной техники

и автоматизированных систем»

Выполнил слушатель: Коренский А.А.

Группа: 1

Вариант: 6

Дата сдачи: .03.2015

Преподаватель: Прытков Д.В.

Новосибирск, 2015г.

# Задание

"Червяк". По экрану движется червяк, направление движения головы которого можно менять (стрелками). Червяк ест призы и растет. Задача – при движении головы не наткнуться на хвост и границы экрана.

# Теоретический материал

При выполнении этой контрольной работы использовался весь ранее пройденный теоретический материал, а также практический опыт выполнения лабораторных работ.

# Описание алгоритма

По экрану движется последовательность символов ‘o’ во главе с символом ‘x’. При «поедании» символов ‘e’, количество символов ‘o’ в «змейке» увеличивается на один. При достижении десяти очков («поедании» десяти символов ‘e’) «голова змеи» меняется на другую латинскую строчную букву. Также на экране присутствует символ ‘@’ («поганка»), при «поедании» которого справа от игрового поля появляется надпись «головокружение» и происходит инверсия управления. При последующем поедании ‘e’ инверсия отменяется и надпись «головокружение» исчезает. Помимо этого справа от игрового поля постоянно присутствуют счетчик размера змеи, счетчик съеденных символов ‘e’ и счетчик съеденных «поганок». При достижении «змейкой» границ игрового поля, обозначенных символами ‘#’, происходит последовательное перемещение «змеи» в противоположную сторону игрового поля и изменение направления движения. Иначе говоря, «змейка» исчезает в стене и выходит из противоположной. Прекратить игровой процесс возможно двумя способами: 1) сделать так, чтобы голова «змейки» столкнулась со своим туловищем; 2) нажать клавишу «Esc».

# 4. Описание реализации

Подключаемые библиотеки:

#include "stdio.h"111

#include "conio.h"

#include "stdlib.h"

#include “windows.h”

Используемые функции:

int main();

printf() – вывод форматированного текста на экран;

gotoxy() – для перехода к заданным координатам экрана (библиотека windows.h);

Sleep() – функция задержки, милисекунды.

getch() – ожидание ввода символа с клавитуры.

Используемые конструкции:

if – условный оператор, для создания ветвления;

while – оператор цикла;

for – оператор цикла;

switch – оператор-переключатель;

#define – препроцессорная директива (подстановка значения по всему коду программы).

Пользовательские функции:

void Dvijenie() – основная функция, отвечающая за многократное выполнение перемещения «змейки»;

void Telo\_Risuet() – функция поочередной отрисовки тела «змейки»;

void Telo\_Stiraet() – функция поочередного стирания тела «змейки»;

void Ramka() – рисует границы игрового поля размером 48 x 40;

void Proverka\_Ramka() – функция делает невозможным выход «змейки» за пределы игрового поля;

void Eda() – функция отвечает за наличие символа еды ‘e’ на игровом поле;

void Lovushka() – функция отвечает за наличие символа поганки ‘@’ на игровом поле;

int Proverka\_Trapeza() – функция проверки, на «поглощение еды змейкой»;

int Proverka\_Trapeza\_poganka() – отвечает за инверсию управления при «поедании поганки»;

int Proverka\_Kanibalizm() – прекращает игровой процесс в случае «поедания змейкой» собственного тела.

# 5. Пример работы программы



Рисунок 1. Результат работы программы

На рисунке 1 показан результат работы игровой программы «Змейка».

**Выводы**

Перед выполнением этой работы необходимо было продумать алгоритм будущей программы, для общего представления ее конструкции. После чего, я приступил к написанию исходного кода.

После было проведено контрольное тестирование программы, которое показало правильность ее работы.

В ходе выполнения контрольной работы были применены опыт и знания, полученные на лекционных и практических занятиях, а также при выполнении прошедших лабораторных работ.

**Приложение. Текст программы с комментариями**

#include "stdio.h"

#include "conio.h"

#include "stdlib.h"

#include <windows.h>

#define N 500

#define ESC 27

//Контрольная работа: игровая программа:"Змейка" Вариант 6

//Задание: "Змейка". По экрану движется червяк, направление

//движения головы которого можно менять (стрелками).

//Червяк ест призы и растет. Задача - при движении

//головы не наткнуться на хвост и границы экрана.

void gotoxy(int x, int y){ //перемещение курсора в координаты 'x' и 'y'

COORD scrn;

HANDLE hOuput = GetStdHandle(STD\_OUTPUT\_HANDLE);

scrn.X = x; scrn.Y = y;

SetConsoleCursorPosition(hOuput,scrn);

}

void Dvijenie();

void Telo\_Risuet(int x[], int y[], int j, int Size);//поочередно отрисовывает части тела

void Telo\_Stiraet(int x[], int y[], int j, int Size); //поочередно стирает части тела

void Ramka(); //рисует рамку поля игры (48 x 40)

void Proverka\_Ramka(int x[], int y[]); //делает невозможным выход

//за пределы игрового поля

void Eda(int x[], int y[], int &xe, int &ye, int Size, int xp, int yp); //рисует 1 еду на игровом

//поле, если ее там нет

void Lovushka(int x[], int y[], int xe, int ye, int Size, int &xp, int &yp); //рисует поганку на игровом поле, если голова змеи изменилась на n - z

int Proverka\_Trapeza(int x[], int y[], int &xe, int &ye, int &pp, int &Size, int &S, int P, char &ZZ); //проверка на поглощение еды

int Proverka \_Trapeza\_poganka(int x[], int y[], int &xp, int &yp, int &P, int S, int Size, int &pp); //делает инверсию управления после поедания поганки

int Proverka\_Kanibalizm(int j, int Size, int x[], int y[], int &exit); //проверка на канибализм

int main (){

Ramka(); //рисует рамку поля игры (48 x 40)

Dvijenie(); //многократно: рисует, стирает, перемещается

}

void Dvijenie(){

int code, dx=1, dy=0, x[N]={15}, y[N]={15}, exit=0, Size=1, j, xe=0, ye=0, S=0, xp=0, yp=0, P=0, pp=1;

char ZZ='x';

while (!exit){ //пока не нажат ESC делать следующее...

for(int i=0;i<200;i++){ //многократный запрос на нажатие клавиши

if (kbhit()){

code=getch();

if (pp==1)

switch(code){

case ESC: exit=1; //при нажатии ESC завершает программу

case 'w': if (dy!=1) {dx=0; dy=-1;} break; //поворачивает вверх, если до этого не двигался вниз

case 's': if (dy!=-1) {dx=0; dy=1;} break; //поворачивает вниз, если до этого не двигался вверх

case 'a': if (dx!=1) {dx=-1; dy=0;} break; //поворачивает влево, если до этого не двигался вправо

case 'd': if (dx!=-1) {dx=1; dy=0;} break; //поворачивает вправо, если до этого не двигался влево

}

else { //инверсия управления при поедании поганки

switch(code){

case ESC: exit=1;

case 's': if (dy!=1) {dx=0; dy=-1;} break;

case 'w': if (dy!=-1) {dx=0; dy=1;} break;

case 'd': if (dx!=1) {dx=-1; dy=0;} break;

case 'a': if (dx!=-1) {dx=1; dy=0;} break;

}

}

}

}

Telo\_Risuet(x, y, j, Size); //поочередно отрисовывает части тела

gotoxy(x[0],y[0]); //переходим в координаты головы змеи

printf("%c", ZZ); //печатаем голову змейки

Eda(x, y, xe, ye, Size, xp, yp); //рисует 1 еду на игровом поле, если ее там нет

Proverka\_Trapeza(x, y, xe, ye, pp, Size, S, P, ZZ);

//проверка на поглощение еды

for(int i=110;i<122;i++)

//если голова змеи изменилась на n - z, то появляется поганка

if (ZZ==i){

Lovushka(x, y, xe, ye, Size, xp, yp);

}

Proverka\_Trapeza\_poganka(x, y, xp, yp, P, S, Size, pp);

//делает инверсию управления после поедания поганки

Sleep(90); //ждем 90 миллисекунд

gotoxy(x[0],y[0]); //возвращаемся в координаты головы змеи

printf(" "); //стираем голову змейки

Telo\_Stiraet(x, y, j, Size); //поочередно стирает части тела

x[0]+=dx; //двигаемся в выбранном ранее направлении

y[0]+=dy;

Proverka\_Kanibalizm(j, Size, x, y, exit); //проверка на канибализм

Proverka\_Ramka(x, y);

//делает невозможным выход за пределы игрового поля

}

}

void Telo\_Risuet(int x[], int y[], int j, int Size){

for(j=Size;j>=1;j--){

x[j]=x[j-1];

y[j]=y[j-1];

}

for(j=Size;j>=1;j--){

gotoxy(x[j],y[j]);

printf("o");

}

}

void Telo\_Stiraet(int x[], int y[], int j, int Size){

for(j=Size;j>=1;j--){

gotoxy(x[j],y[j]);

printf(" ");

}

}

void Ramka(){

int i, j;

for(i=0;i<50;i++) //верхняя горизонтальная граница

printf("#");

printf("\n");

for(i=0;i<40;i++){ //левая и правая вертикальные границы

printf("#");

for(j=0;j<48;j++)

printf(" ");

printf("#\n");

}

for(i=0;i<50;i++) //нижняя горизонтальная граница

printf("#");

}

void Proverka\_Ramka(int x[], int y[]){

if (x[0]>48)

x[0]=1;

if (x[0]<1)

x[0]=48;

if (y[0]>40)

y[0]=1;

if (y[0]<1)

y[0]=40;

}

void Eda(int x[], int y[], int &xe, int &ye, int Size, int xp, int yp){

int i, t=Size;

if ((xe==0) && (ye==0)){

xe=rand()%46+2;

ye=rand()%38+2;

for(i=Size;i>=1;i--)

if (((xe==x[i]) && (ye==y[i])) || ((xe==x[0]) && (ye==y[0])) || ((xe==xp) && (ye==yp)))

t=0;

while (t!=Size) {

xe=rand()%46+2;

ye=rand()%38+2;

for(i=Size;i>=1;i--){

if (((xe!=x[i]) || (ye!=y[i])) && ((xe!=x[0]) || (ye!=y[0])) && ((xe!=xp) || (ye!=yp)))

t++;

else

t=0;

}

}

gotoxy(xe,ye);

printf("e");

}}

void Lovushka(int x[], int y[], int xe, int ye, int Size, int &xp, int &yp){

int i, t=Size;

if ((xp==0) && (yp==0)){

xp=rand()%46+2;

yp=rand()%38+2;

for(i=Size;i>=1;i--)

if (((xp==x[i]) && (yp==y[i])) || ((xp==x[0]) && (yp==y[0])) || ((xe==xp) && (ye==yp)))

t=0;

while (t!=Size) {

xp=rand()%46+2;

yp=rand()%38+2;

for(i=Size;i>=1;i--){

if (((xp!=x[i]) || (yp!=y[i])) && ((xp!=x[0]) || (yp!=y[0])) && ((xe!=xp) || (ye!=yp)))

t++;

else

t=0;

}

}

gotoxy(xp,yp);

printf("@");

}

}

int Proverka\_Trapeza(int x[], int y[], int &xe, int &ye, int &pp, int &Size, int &S, int P, char &ZZ){

if ((x[0]==xe) && (y[0]==ye)){

if(pp==0){ //возвращаем управление в нормальный режим, если была инверсия

pp=1;

gotoxy(55,5);

printf(" ");

}

Size++; //увеличивает размер змейки, если покушал

S++; //счетчик размера змеи

xe=0; //обнуляем после трапезы координаты еды

ye=0;

gotoxy(52,1);

printf("Skushal: %d Razmer zmei: %d Poganki: %d", S, Size, P); //выводит на экран информацию по игре

if (S%10==0){ //каждые набранные 10 единиц еды меняется тип головы

ZZ=rand()%25+97;

//латиница прописная в ASCII 97 - 122 = 25

}

}

}

int Proverka\_Trapeza\_poganka(int x[], int y[], int &xp, int &yp, int &P, int S, int Size, int &pp){

if ((x[0]==xp) && (y[0]==yp)){

xp=0; //обнуляем после поедания поганки ее координаты

yp=0;

gotoxy(55,5);

printf("GOLOVOKRUJENIE!!!");

P++;

gotoxy(52,1);

printf("Skushal: %d Razmer zmei: %d Poganki: %d", S, Size, P);

pp=0; //маяк управления

}

}

int Proverka\_Kanibalizm(int j, int Size, int x[], int y[], int &exit){

for(j=Size;j>=1;j--){

if ((x[0]==x[j]) && (y[0]==y[j])){

//если координаты головы змеи равны координатам части тела змеи, то выход

exit=1;

}

}

}

Защита:

1)

2)