|  |  |
| --- | --- |
| Uni21b | МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ,  ФИЛИАЛ В Г.НИЖНЕВАРТОВСКЕ  КАФЕДРА «ИНФОРМАТИКА» |

****

**Практикум по программированию на языках высокого уровня**

Варианты заданий и методические указания к выполнению

курсового проекта

для студентов направления «Информатика и вычислительная техника»

# Нижневартовск

# 2014

УДК 519.2

© Кротова А.П.

Одобрено

редакционно-издательским советом филиала

(протокол № 2 от 16.10.2014)

Практикум по программированию на языках высокого уровня: Варианты заданий и методические указания к выполнению курсового проекта для студентов направления «Информатика и вычислительная техника» / А.П. Кротова – Нижневартовск, 2014. – 27 с.

Задания составлены в соответствии с ФГОС-3 по направлению обучения «Информатика и вычислительная техника» и предназначены для формирования компетенций ПК-3, 5.

Утверждено на заседании кафедры

Протокол №2

«16» октября 2014 год

Содержание

[Курсовой проект 3](#_Toc409734149)

[Разработка Windows-приложения на языке C++. Задания для курсового проекта 4](#_Toc409734150)

[Образец выполнения задания 19](#_Toc409734151)

[Учебно-методическое и информационное обеспечение дисциплины 27](#_Toc409734152)

**Курсовой проект**

В рамках курсового проекта студентами проектируется алгоритм решения задания по варианту и разрабатывается Windows-приложение на C++, реализующее этот алгоритм. После чего студент составляет пояснительную записку о ходе проектирования и результатах разработки, и сдает ее преподавателю в печатном виде.

Пояснительная записка должна состоять из следующих пунктов:

1 ПОСТАНОВКА ЗАДАЧИ

1.1 Исходные данные (входные данные)

1.2 Перечень результатов (выходные данные)

1.3 Сбои (случаи возникновения ошибок)

1.4 Средства разработки

2 ПРОЕКТИРОВАНИЕ

2.1 Разработка алгоритма

2.2 Блок-схема алгоритма

3 РЕАЛИЗАЦИЯ

4 ТЕСТИРОВАНИЕ

4.1 Тестирование алгоритма

4.2 Тестирование программы

ЗАКЛЮЧЕНИЕ

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

**Разработка Windows-приложения на языке C++.   
Задания для курсового проекта**

**Вариант 1.** Электронная таблица.

Составьте программу для реализации электронной таблицы, изображенной на рис. 1.



Рис. 1. Пример интерфейса приложения «Электронная таблица».

**Вариант 2.** Игра «Сапер».

Составьте программу для реализации игры «Сапер», изображенной на рис. 2.

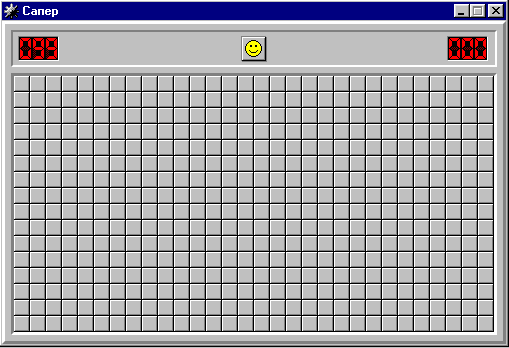


Рис. 2. Игра «Сапер».

**Вариант 3.** Графический редактор с карандашом и ластиком.

Составьте программу для реализации графического редактора, изображенного на рис. 3.

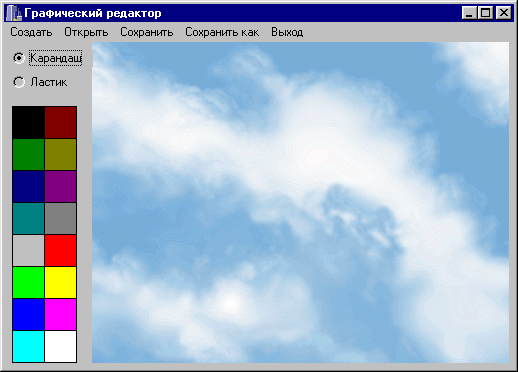


Рис. 3. Пример интерфейса приложения «Графический редактор».

**Вариант 4.** Папки и файлы в дереве.

Составьте программу для реализации древовидной структуры папок и файлов. Пример изображен на рис. 4.

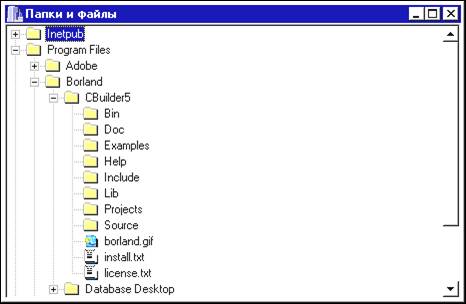


Рис. 4. Пример дерева папок и файлов.

Примечание. Для получения списка папок и файлов используйте функции findfirst и findnext.

**Вариант 5.** Графический редактор линий.

Составьте программу для реализации графического редактора линий, изображенного на рис. 5.

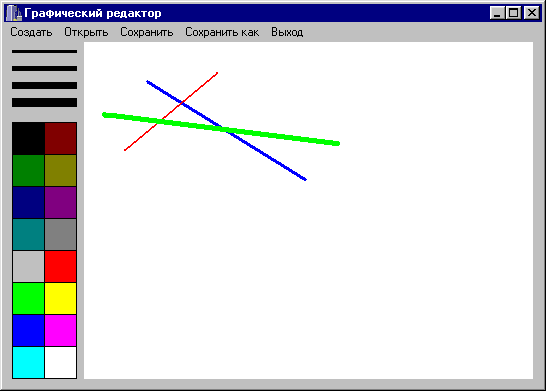


Рис.5. Пример интерфейса приложения «Графический редактор линий».

**Вариант 6.** Обход шахматной доски конем.

Составьте программу для обхода шахматной доски конем. Клеточки в которых конь уже был должны быть помечены крестиком, в них повторный ход запрещен (рис. 6).

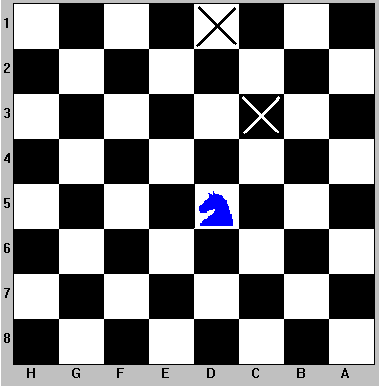


Рис. 6. Программа для обхода шахматной доски конем.

**Вариант 7.** Графический редактор многоугольных фигур.

Составьте программу для реализации графического редактора много­угольных фигур, изображенного на рис. 7.

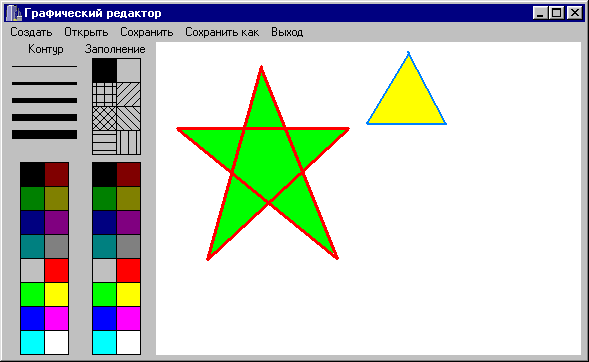


Рис.7. Пример интерфейса приложения «Графический редактор многоугольных фигур».

**Вариант 8.** Игра «Крестики-нолики».

Составьте программу для реализации игры «Крестики-нолики» на поле 3x3, изображенной на рис. 8.

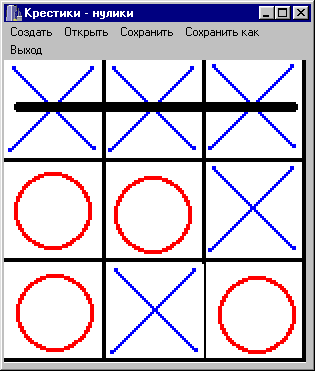


Рис. 8. Пример интерфейса игры «Крестики-нолики».

**Вариант 9.** Калькулятор.

Составьте программу для реализации калькулятора, изображенного на рис. 9.

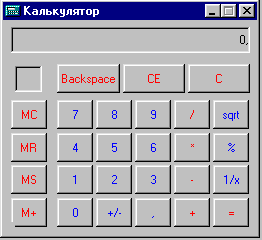


Рис.9. Пример интерфейса приложения «Калькулятор».

**Вариант 10.** Игра «Lines».

Составьте программу для реализации игры «Lines», изображенной на рис. 10.

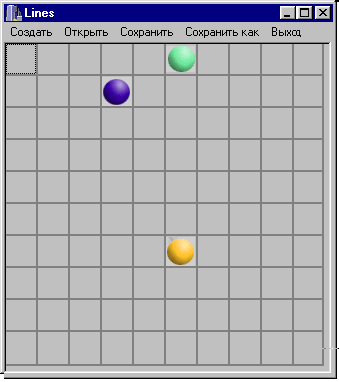


Рис. 10. Игра «Lines».

На каждом шаге игры случайным образом в свободные клеточки добавляются три шарика случайного цвета (всего восемь цветов). Игрок может переставить один любой шарик в любую свободную клетку. Если при этом в ряд будет расположено пять или более шариков, то они сгорают и клеточки становятся свободными.

**Вариант 11.** Ежедневник.

Составьте программу для реализации ежедневника, изображенного на рис. 11.



Рис.11. Пример интерфейса приложения «Ежедневник».

**Вариант 12.** Игра «Реверси».

Составьте программу для реализации игры «Реверси». На поле 10x10 игроки поочередно ставят фишки, первый игрок – белые, второй игрок - черные. Если при постановке очередной фишки игрок окружил фишками своего цвета чужие фишки, то чужие фишки меняют цвет на свой (рис. 12). Игра за­канчивается, когда все поле заставлено фишками. Победителем является тот, чьих фишек больше.



Рис. 12. Игра «Реверси».

**Вариант 13.** Календарь на год.

Составьте программу для реализации календаря на год, изображенного на рис. 13.

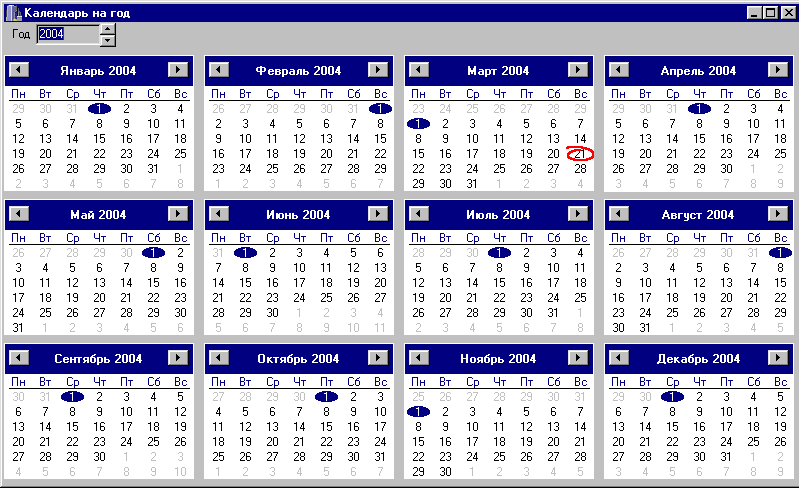


Рис. 13. Пример интерфейса приложения «Календарь на год».

**Вариант 14.** Игра «Сапер».

Составьте программу для реализации игры «Сапер», изображенной на рис. 14.

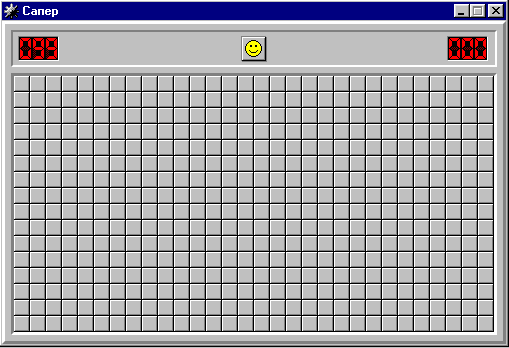


Рис. 14. Игра «Сапер».

**Вариант 15.** Графический редактор с карандашом и ластиком.

Составьте программу для реализации графического редактора, изображенного на рис. 15.

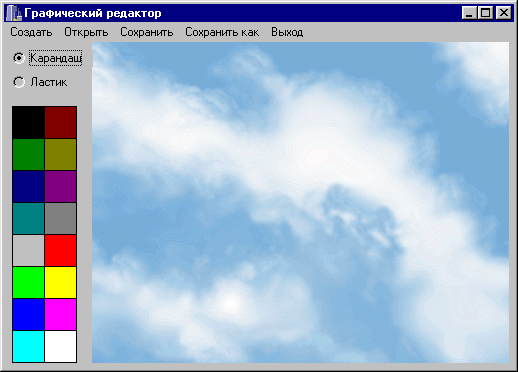


Рис. 15. Пример интерфейса приложения «Графический редактор».

**Вариант 16.** Папки и файлы в дереве.

Составьте программу для реализации древовидной структуры папок и файлов. Пример изображен на рис. 16.

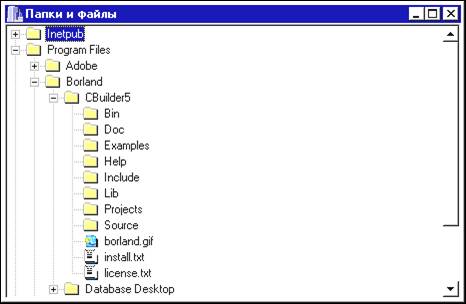


Рис. 16. Пример дерева папок и файлов.

Примечание. Для получения списка папок и файлов используйте функции findfirst и findnext.

**Вариант 17.** Графический редактор линий.

Составьте программу для реализации графического редактора линий, изображенного на рис. 17.

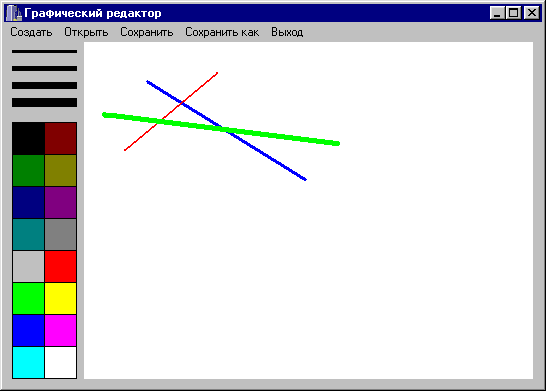


Рис.17. Пример интерфейса приложения «Графический редактор линий».

**Вариант 18.** Обход шахматной доски конем.

Составьте программу для обхода шахматной доски конем. Клеточки в которых конь уже был должны быть помечены крестиком, в них повторный ход запрещен (рис. 18).

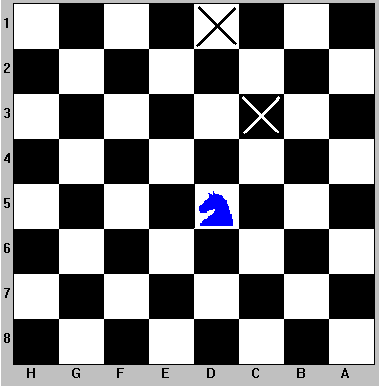


Рис. 18. Программа для обхода шахматной доски конем.

**Вариант 19.** Графический редактор многоугольных фигур.

Составьте программу для реализации графического редактора много­угольных фигур, изображенного на рис. 7.

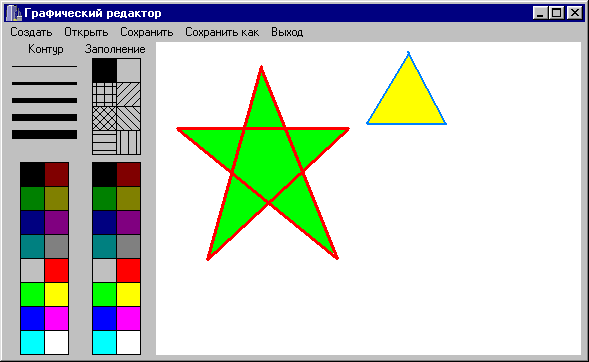


Рис.19. Пример интерфейса приложения «Графический редактор многоугольных фигур».

**Вариант 20.** Игра «Крестики-нолики».

Составьте программу для реализации игры «Крестики-нолики» на поле 3x3, изображенной на рис. 20.

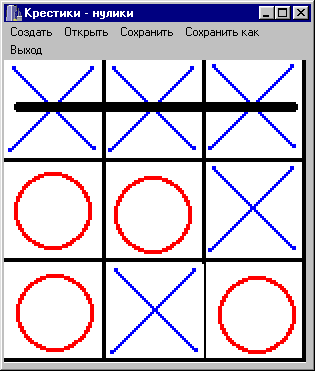


Рис. 20. Пример интерфейса игры «Крестики-нолики».

**Вариант 21.** Калькулятор.

Составьте программу для реализации калькулятора, изображенного на рис. 21.

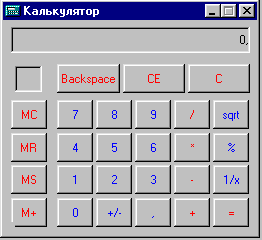


Рис.21. Пример интерфейса приложения «Калькулятор».

**Вариант 22.** Игра «Lines».

Составьте программу для реализации игры «Lines», изображенной на рис. 22.

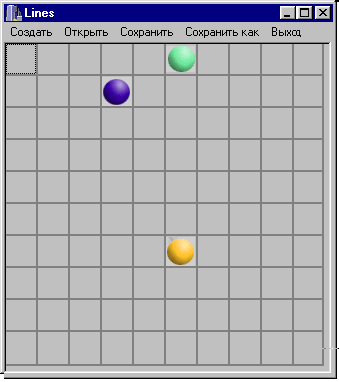


Рис. 22. Игра «Lines».

На каждом шаге игры случайным образом в свободные клеточки добавляются три шарика случайного цвета (всего восемь цветов). Игрок может переставить один любой шарик в любую свободную клетку. Если при этом в ряд будет расположено пять или более шариков, то они сгорают и клеточки становятся свободными.

**Вариант 23.** Ежедневник.

Составьте программу для реализации ежедневника, изображенного на рис. 11.



Рис.23. Пример интерфейса приложения «Ежедневник».

**Вариант 24.** Игра «Реверси».

Составьте программу для реализации игры «Реверси». На поле 10x10 игроки поочередно ставят фишки, первый игрок – белые, второй игрок - черные. Если при постановке очередной фишки игрок окружил фишками своего цвета чужие фишки, то чужие фишки меняют цвет на свой (рис. 12). Игра за­канчивается, когда все поле заставлено фишками. Победителем является тот, чьих фишек больше.



Рис. 24. Игра «Реверси».

**Вариант 25.** Календарь на год.

Составьте программу для реализации календаря на год, изображенного на рис. 25.

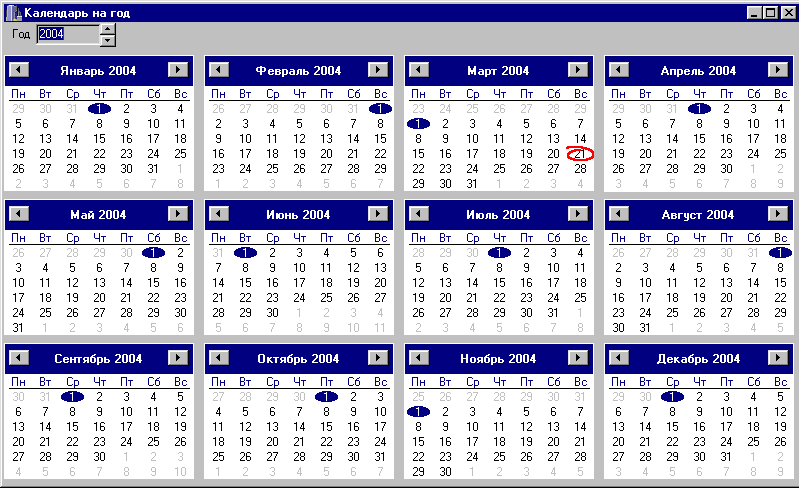


Рис. 25. Пример интерфейса приложения «Календарь на год».

**Вариант 26.** Электронная таблица.

Составьте программу для реализации электронной таблицы, изображенной на рис. 26.



Рис. 26. Пример интерфейса приложения «Электронная таблица».

**Вариант 27.** Часы.

Составьте программу для реализации часов со стрелками, изображен­ных на рис. 27.

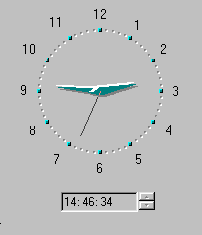


Рис.27. Пример интерфейса приложения «Часы».

**Образец выполнения задания**

**1 ПОСТАНОВКА ЗАДАЧИ**

Составьте программу для реализации часов со стрелками, изображенных на рис. 1.

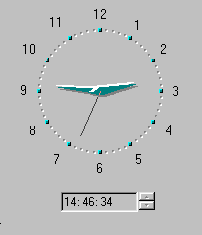


Рис.1. Пример интерфейса приложения «Часы».

**1.1 Исходные данные (входные данные)**

R – радиус циферблата, может быть вещественной переменной(типа double);

Xc, Yc – координаты центра циферблата, должны быть целочисленными переменными (типа int);

Ds – длина секундной стрелки, может быть вещественной переменной (типа double). Можно этой величине задать значение: Ds = R-3, чтобы стрелка не выходила за границы циферблата;

Dm – длина минутной стрелки, может быть вещественной переменной (типа double). Так как эта стрелка должна быть покороче секундной, то можно задать ей значение: Dm = 2\*Ds/3;

Dch – длина часовой стрелки, может быть вещественной переменной (типа double). Так как эта стрелка должна быть короче минутной, то можно задать ей такое значение: Dch = Ds/2;

ts – секунды текущего времени (типа double);

tm – минуты текущего времени (типа double);

tch – часы текущего времени (типа double).

**1.2 Перечень результатов (выходные данные)**

Xs, Ys – координаты радиус-вектора секундной стрелки, должны быть целочисленными переменными (типа int);

Xm, Ym – координаты радиус-вектора минутной стрелки, должны быть целочисленными переменными (типа int);

Xch, Ych – координаты радиус-вектора часовой стрелки, должны быть целочисленными переменными (типа int).

**1.3 Средства разработки**

Задача будет решена с помощью алгоритма, реализованного в среде Borland Developer Studio 2006 на языке программирования С++.

**2 ПРОЕКТИРОВАНИЕ**

**2.1 Разработка алгоритма**

Для того чтобы сымитировать движение стрелок по циферблату, можно воспользоваться компонентом Timer. По таймеру раз в секунду должен запускаться алгоритм отрисовки секундной, минутной и часовой стрелок. Этот алгоритм определим как основной, и программно реализуем в виде функции MainProg, котjрая будет вызываться в обработчике событие OnTimer компонента Timer. В рамках основного алгоритма сначала перед отрисовкой стрелок должны вычисляться координаты радиус-векторов этих стрелок. Так как для каждой из стрелок алгоритм вычисления координат радиус-векторов будет аналогичен, целесообразно его выделить в отдельные алгоритмы (подпрограммы), которые будут программно реализован в виде функций XStrelki и YStrelki, вычисляющих координаты X и Y соответственно. Для этих алгоритмов определим входные данные:

Параметры функции XStrelki:

Xo – координата X центра циферблата (типа int);

Параметры функции YStrelki:

Yo – координата Y центра циферблата (типа int);

Следующие параметры будут одинаковыми для обеих функций XStrelki и YStrelki:

D – длина стрелки, должна быть такого же типа как и переменные Ds, Dm и Dch, то есть вещественной (типа double);

Um – угол в минутах (типа double). Так как для минутной и секундной стрелок количество делений на циферблате совпадает (60) то для секундной стрелки Um будет задаваться как количество секунд без дополнительных преобразований. А вот для часовой стрелки на циферблате 12 делений, поэтому для часовой стрелки Um будет задаваться как количество часов умноженное на 5.

Выходные данные, то есть результаты возвращаемые функцией XStrelki:

X – координата X радиус-вектора стрелки (типа int).

X=Xo+D\*sin(Um\*π/30)

Yo

D\*cos(Um\*π/30)

D\*sin(Um\*π/30)

**X**

**Y**

Um\*π/30

Xo

(Xo-R)

Y = Yo – D\*cos(Um\*π/30)

(Yo+R)

**D**

Выходные данные, то есть результаты возвращаемые функцией YStrelki:

Y – координата Y радиус-вектора стрелки (типа int).

Алгоритм подпрограммы XStrelki:

1) Вычисляем X по формуле: X = Xo + D\*cos(Um\*π/30);

2) Возвращение из функции X.

Алгоритм подпрограммы YStrelki:

1) Вычисляем Y по формуле: Y = Yo - D\*sin(Um\*π/30);

2) Возвращение из функции Y.

Основной алгоритм отрисовки стрелок:

1) Определяем текущее время и переменным присваиваем соответствующие значения:

1.1) Переменной ts присваиваем секунды текущего системного времени;

1.2) Переменной tm присваиваем минуты текущего системного времени;

1.3) Переменной tch присваиваем часы текущего системного времени.

2) Вычисление координат радиус-векторов секундной, минутной и часовой стрелок:

2.1) Xs =XStrelki(Xc, Ds, ts); Ys =YStrelki(Yc, Ds, ts);

2.2) Xm=XStrelki(Xc, Dm, tm); Ym =YStrelki(Yc, Dm, tm);

2.3) Xch=XStrelki(Xc, Dch, tch\*5); Ych=YStrelki(Yc, Dch, tch\*5).

3) Очистка предыдущего положения стрелок.

4) Рисование всех стрелок:

4.1) Рисование линии по точкам (Xc,Yc) и (Xs,Ys) (секундная стрелка);

4.2) Рисование линии по точкам (Xc,Yc) и (Xm,Ym) (минутная стрелка);

4.3) Рисование линии по точкам (Xc,Yc) и (Xch,Ych) (часовая стрелка).

**2.2 Блок-схема алгоритма**

XStrelki(Xo, D, Um)

Возврат X

X = Xo + D\*sin(Um\*π/30)

YStrelki(Yo, D, Um)

Возврат Y

Y = Yo - D\*cos(Um\*π/30)

Начало

ts = секунды текущего времени

Xs = XStrelki(Xc, Ds, ts)

tm = минуты текущего времени

tср = часы текущего времени

Ys = YStrelki(Yc, Ds, ts)

Xm = XStrelki(Xc, Dm, tm)

Ym =YStrelki(Yc, Dm, tm)

Xch = XStrelki(Xc, Dch, tch\*5)

Ych = YStrelki(Yc, Dch, tch\*5)

Рисование линии (Xc,Yc), (Xs,Ys)

Рисование линии (Xc,Yc), (Xm,Ym)

Рисование линии (Xc,Yc), (Xch,Ych)

Конец

Очистка предыдущего положения стрелок.

**3 РЕАЛИЗАЦИЯ**

Для реализации приложения был использован языка программирования C++. Код получившейся программы представлен ниже:

**Код в файле Unit1.h:**

//---------------------------------------------------------------------------

#ifndef Unit1H

#define Unit1H

//---------------------------------------------------------------------------

#include <Classes.hpp>

#include <Controls.hpp>

#include <StdCtrls.hpp>

#include <Forms.hpp>

#include <ExtCtrls.hpp>

#include <GIFImg.hpp>

#include <Graphics.hpp>

#include <ComCtrls.hpp>

//---------------------------------------------------------------------------

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TPaintBox \*PaintBox1;

TTimer \*Timer1;

TImage \*Image1;

TDateTimePicker \*DateTimePicker1;

void \_\_fastcall Timer1Timer(TObject \*Sender);

void \_\_fastcall PaintBox1Paint(TObject \*Sender);

private: // User declarations

public: // User declarations

\_\_fastcall TForm1(TComponent\* Owner);

};

//---------------------------------------------------------------------------

extern PACKAGE TForm1 \*Form1;

//---------------------------------------------------------------------------

#endif

**Код в файле Unit1.cpp:**

//---------------------------------------------------------------------------

#include <vcl.h>

#include <math.h>

#pragma hdrstop

#include "Unit1.h"

//---------------------------------------------------------------------------

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

int XStrelki(int Xo, double D, double Um);

int YStrelki(int Yo, double D, double Um);

void MainProg();

const double R = 208, Pi=3.141592654;

const int Xc = 294, Yc = 289;

const double Ds = R-3, Dm = 3\*Ds/4, Dch = Ds/2;

int Xs, Ys, Xm, Ym, Xch, Ych;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

MainProg();

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Timer1Timer(TObject \*Sender)

{

MainProg();

}

//---------------------------------------------------------------------------

int XStrelki(int Xo, double D, double Um)

{

int X = Xo + (int)(D\*sin(Um\*Pi/30));

return X;

}

int YStrelki(int Yo, double D, double Um)

{

int Y = Yo - (int)(D\*cos(Um\*Pi/30));

return Y;

}

void MainProg()

{

Form1->DateTimePicker1->DateTime=Now();

double ts = Form1->DateTimePicker1->Time.FormatString("s").ToFloat(),

tm = Form1->DateTimePicker1->Time.FormatString("n").ToFloat(),

tch = Form1->DateTimePicker1->Time.FormatString("h").ToFloat()+

(tm\*60+ts)/3600;

Xs = XStrelki(Xc, Ds, ts);

Ys = YStrelki(Yc, Ds, ts);

Xm = XStrelki(Xc, Dm, tm);

Ym = YStrelki(Yc, Dm, tm);

Xch = XStrelki(Xc, Dch, tch\*5);

Ych = YStrelki(Yc, Dch, tch\*5);

Form1->PaintBox1->Repaint();

Form1->PaintBox1->Canvas->Pen->Width = 2;

Form1->PaintBox1->Canvas->MoveTo(Xc, Yc);

Form1->PaintBox1->Canvas->LineTo(Xs, Ys);

Form1->PaintBox1->Canvas->Pen->Width = 4;

Form1->PaintBox1->Canvas->MoveTo(Xc, Yc);

Form1->PaintBox1->Canvas->LineTo(Xm, Ym);

Form1->PaintBox1->Canvas->Pen->Width = 6;

Form1->PaintBox1->Canvas->MoveTo(Xc, Yc);

Form1->PaintBox1->Canvas->LineTo(Xch, Ych);

}

//---------------------------------------------------------------------------

**4 ТЕСТИРОВАНИЕ**

**4.1 Тестирование программы**

Скриншоты, демонстрирующий работу программы представлены ниже:

**ЗАКЛЮЧЕНИЕ**

В ходе выполнения задания был разработаны алгоритмы, реализующие виртуальную имитацию часов. Затем эти алгоритмы были программно реализован средствами среды Borland Developer Studio 2006 на языке программирования С++.

**Учебно-методическое и информационное обеспечение дисциплины**

Основная литература:

1. Павловская, Т.А. С/С ++. Программирование на языке высокого уровня: учебник / Т.А.Павловская.– СПб.: Питер, 2010.– 461с.: ил.
2. Подбельский, В.В. Язык СИ++.: учебное пособие /В.В. Подбельский .–5-е изд.–М.: Финансы и статистика, 2006. – 260с.: ил.
3. Подбельский, В.В. Программирование на языке СИ: учеб. пособие / В.В.Подбельский, С.С.Фомин. – 2-е изд., доп. – М.: Финансы и статистика, 2002.- 600с.: ил.

Дополнительная литература:

1. Аляев, Ю.А. Алгоритмизация и языки программирования Pascal, C++, Visual Basic: учебно- справочное пособие / Ю.А.Аляев, О.А.Козлов.- М.: Финансы и статистика, 2004.- 320с.
2. Иванова Г.С. Основы программирования: Учебник для вузов. – 2-е изд., перераб. и доп. – М.: Изд-во МГТУ им. Н.Э. Баумана, 2002. – 416 с.
3. Культин, Н.Б. С.С ++ в задачах и примерах / Н.Б. Культин.- СПб.: БХВ\_Петербург, 2003. -288с.: ил.
4. Немнюгин, С.А. Turbo Paskal: практикум / С.А.Немнюгин.- СПб.: Питер, 2000.-256с.: ил.
5. Павловская, Т.А. Паскаль. Программирование на языке высокого уровня: учебник для вузов / Т.А.Павловская.– СПб.: Питер, 2008.– 393 с.: ил.
6. Немнюгин, С.А. Turbo Paskal. Программирование на языке высокого уровня: учебник для вузов / С.А. Немнюгин. - 2-е изд.- СПб.:Питер, 2005.-544с.: ил.

Программное обеспечение

1. Embarcadero RAD Studio 2006
2. Microsoft Visual Studio Express
3. Free Pascal

Базы данных, информационно-справочные и поисковые системы:

1. Электронный ресурс Национального открытого университета ИНТУИТ www.intuit.ru