**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра информатики**

отчет

**по лабораторным работам №1,2,3**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3488 |  | Шабалин Д.О. |
| Преподаватель |  | Армашев А.КА |

Санкт-Петербург

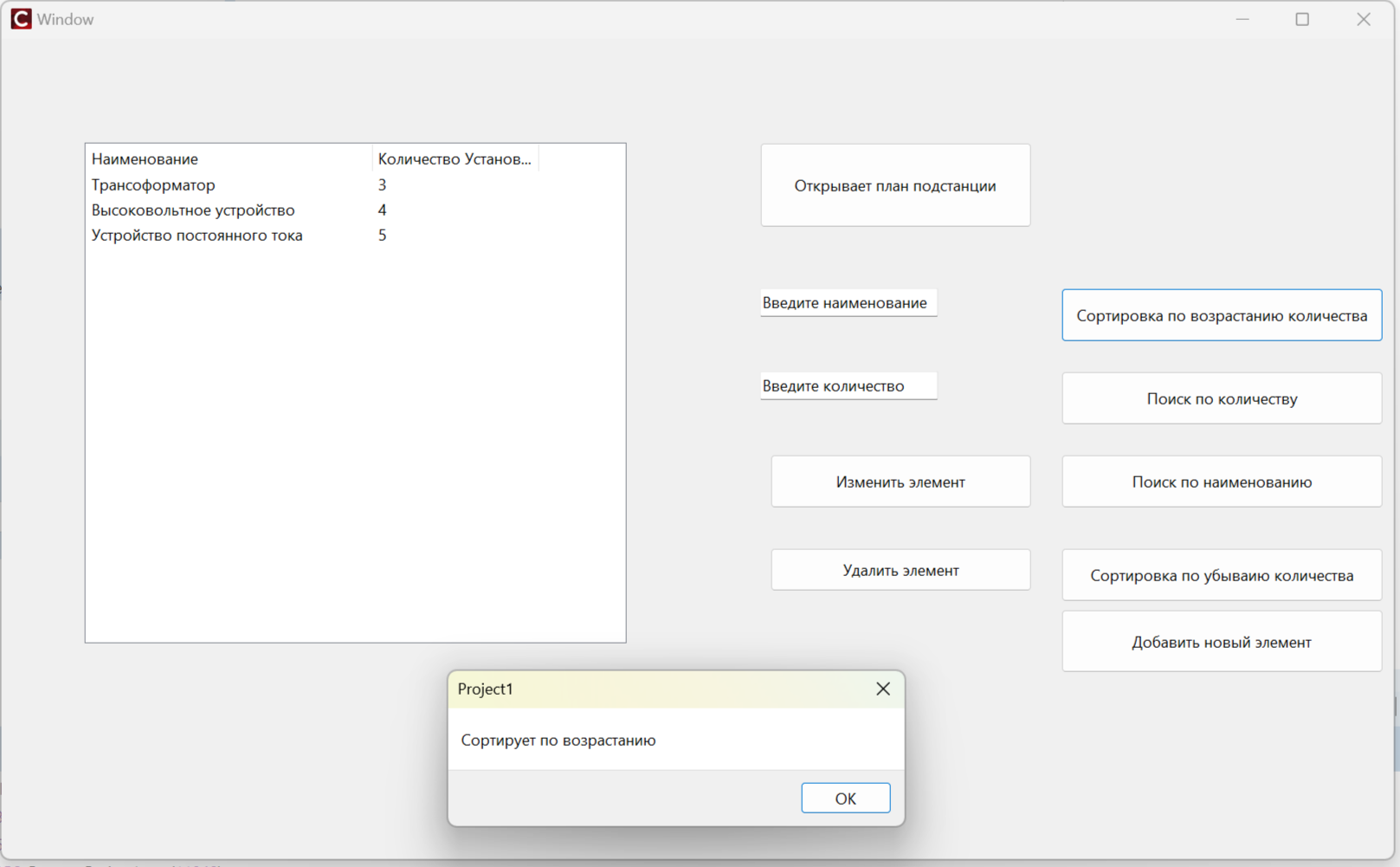
2024

|  |
| --- |
| Изменение элемента  Удалить элемент  План подстанции  Добавить элемент  Поиск по количеству  Сортировка  Подстанция |

По убыванию

По возрастанию

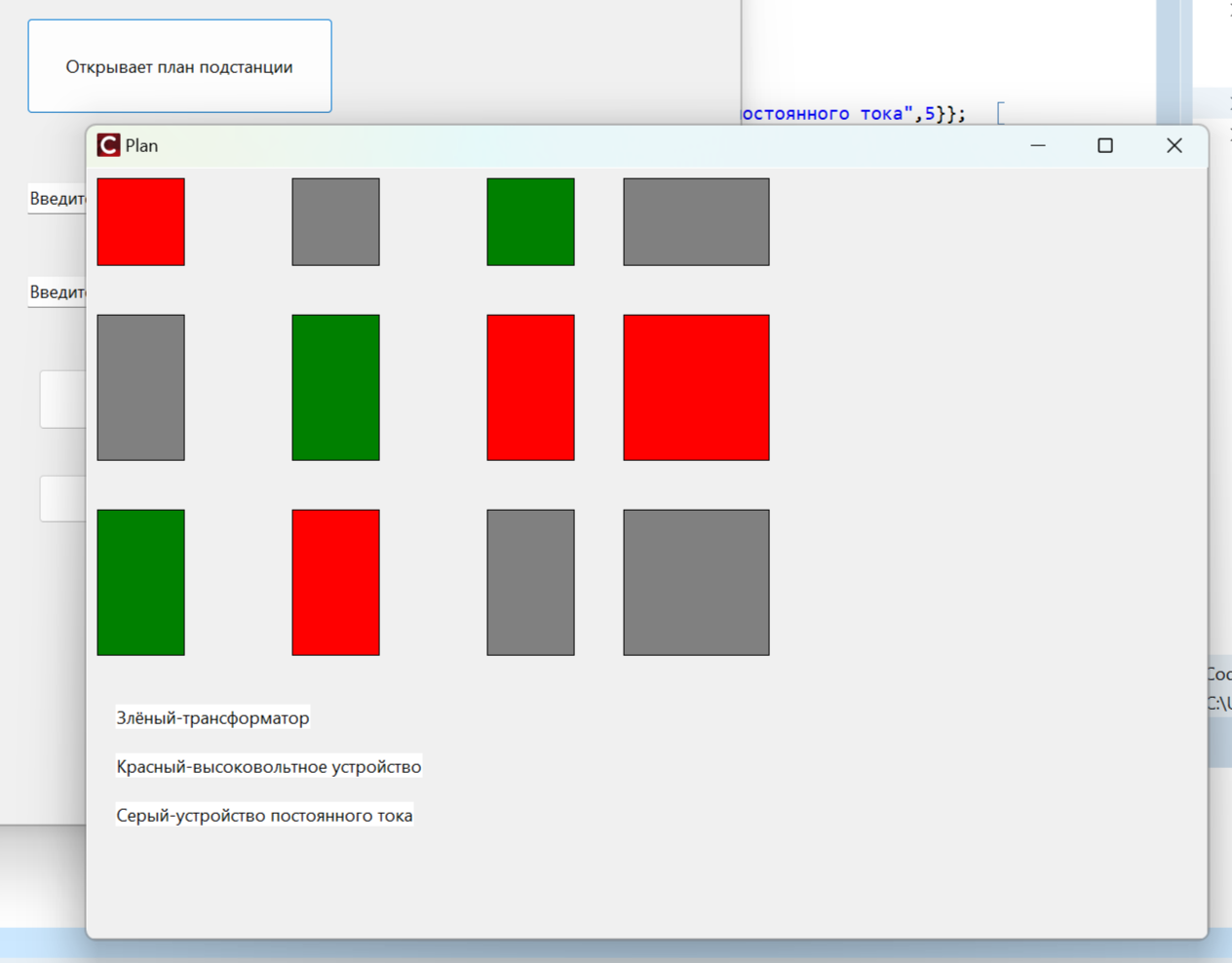
**Главная форма**



На главной форме находится таблица и кнопки для работы с ней, а также кнопка для вызова плана подстанции. При нажатии копки для работы таблицей появляется надпись, описывающая дальнейшую работу кнопки.

**Второстепенная форма**

Выводит план подстанции с легендой



**Код главной формы:**

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

#include "Unit1.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TWindow \*Window;

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

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

: TForm(Owner)

{

}

int count=3;

struct devices {char name[100];int quantity;};

struct devices cev[]={{"Трансоформатор",3},{"Высоковольтное устройство",4},{"Устройство постоянного тока",5}};

void TWindow::DevicesView()

{

spisok->Items->BeginUpdate();

spisok->Items->Clear();

TListColumn \*NewColumn=spisok ->Columns->Add();

NewColumn->Caption = "Наименование";

NewColumn=spisok ->Columns->Add();

NewColumn->Caption = "Количество Установленных объектов";

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

{

TListItem\* item = spisok->Items->Add();

item->Caption = cev[i].name;

item->SubItems->Add(cev[i].quantity);

item->ImageIndex = 1;

item->Data = (void\*)i;

}

spisok->Items->EndUpdate();

}

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

void \_\_fastcall TWindow::tabel(TObject \*Sender)

{

DevicesView();

}

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

void \_\_fastcall TWindow::onplan(TObject \*Sender)

{

Plan->Visible=true;

}

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

void \_\_fastcall TWindow::poiskcol(TObject \*Sender)

{

ShowMessage("Ищет по количеству");

}

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

void \_\_fastcall TWindow::izm(TObject \*Sender)

{

ShowMessage("Изменяет элемент");

}

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

void \_\_fastcall TWindow::del(TObject \*Sender)

{

ShowMessage("Удаляет элемент");

}

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

void \_\_fastcall TWindow::sortvozr(TObject \*Sender)

{

ShowMessage("Сортирует по возрастанию");

}

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

void \_\_fastcall TWindow::sorubav(TObject \*Sender)

{

ShowMessage("Сортирует по убыванию");

}

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

void \_\_fastcall TWindow::poiskname(TObject \*Sender)

{

ShowMessage("Ищет по наименованию");

}

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

void \_\_fastcall TWindow::dobav(TObject \*Sender)

{

ShowMessage("Добавить новый элмент");

}

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

**Код второстепенной формы**

#include <vcl.h>

#pragma hdrstop

#include "Unit2.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TPlan \*Plan;

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

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

: TForm(Owner)

{

}

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

void \_\_fastcall TPlan::ris(TObject \*Sender)

{

TCanvas\* cs = paint->Canvas;

cs->Pen->Color = clBlack;

cs->Pen->Style = psSolid;

cs->Brush->Color = clRed;

cs->Rectangle(10, 10, 100, 100);

cs->Brush->Color = clGray;

cs->Rectangle(10, 150, 100, 300);

cs->Brush->Color = clGreen;

cs->Rectangle(10, 350, 100, 500);;

cs->Brush->Color = clRed;

cs->Rectangle(210, 350, 300, 500);

cs->Brush->Color = clGray;

cs->Rectangle(210, 10, 300, 100);

cs->Brush->Color = clGreen;

cs->Rectangle(210, 150, 300, 300);

cs->Brush->Color = clRed;

cs->Rectangle(410, 150, 500, 300);

cs->Brush->Color = clGray;

cs->Rectangle(410, 350, 500, 500);

cs->Brush->Color = clGreen;

cs->Rectangle(410, 10, 500, 100);

cs->Brush->Color = clRed;

cs->Rectangle(550, 150, 700, 300);

cs->Brush->Color = clGray;

cs->Rectangle(550, 350, 700, 500);

cs->Brush->Color = clGray;

cs->Rectangle(550, 10, 700, 100);

cs->Brush->Color = clWhite;

cs->TextOut(30, 550, "Злёный-трансформатор");

cs->TextOut(30, 600, "Красный-высоковольтное устройство");

cs->TextOut(30, 650, "Серый-устройство постоянного тока");

}

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