МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Российский государственный университет им. А.Н. Косыгина   
(Технологии. Дизайн. Искусство)»**

Институт ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ЦИФРОВЫЕ ТРАНСФОРМАЦИИ

**Отчет по лабораторной работе № 13**

**по дисциплине «Объектно-ориентированное программирование»**

**Тема: «Отработка задач с указателями this»**

Выполнил: Сидоров Д. С., группа ИТС-123

Проверил: к.т.н., доц. Семёнов А. А.

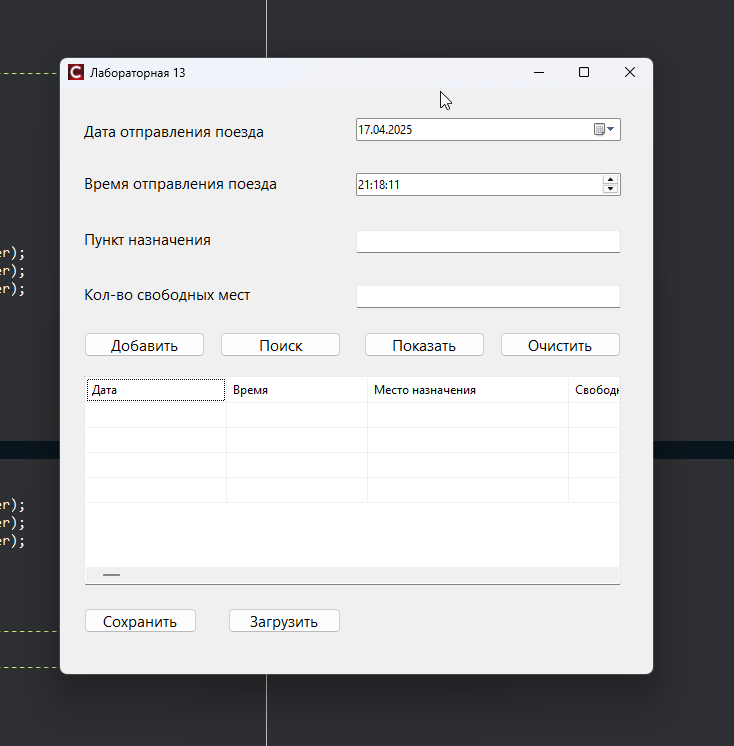
Москва, 2025г

**Отработка задач с указателями this**

Лабораторная работа посвящена изучению классов в ООП на C++.

Задание:

В железнодорожных кассах хранится информация о свободных местах в поездах на ближайшую неделю в следующем виде: дата выезда, пункт назначения, время отправления, число свободных мест. Поступает запрос на резервирование m мест до города N на k-й день недели с временем отправления поезда не позднее t часов вечера. Вывести время отправления или сообщение о невозможности выполнить заказ в полном объеме.

*Рисунок 1. Демонстрация работы готовой программы.*

**Листинг кода**

**Unit1.cpp**

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

#include <vcl.h>

#include <vector>

#pragma hdrstop

#include "Unit1.h"

#include "Unit2.h"

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

#pragma package(smart\_init)

#pragma resource "\*.dfm"

TForm1 \*Form1;

std::vector<train> trains;

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

void \_\_fastcall TForm1::Button2Click(TObject \*Sender)

{

bool good = false;

StringGrid1->RowCount = 1;

for (int i = 0; i < trains.size(); i++) {

String sdate = Form1->datetrain->Date;

if(sdate != trains[i].get\_date())

continue;

if (Form1->Edit1->Text != "" && Form1->Edit1->Text != trains[i].get\_point())

continue;

String stime = Form1->timetrain->Time;

if (stime < trains[i].get\_time())

continue;

if (StrToInt(Form1->Edit2->Text) > trains[i].get\_space())

continue;

good = true;

int row\_count = StringGrid1->RowCount;

StringGrid1->RowCount = row\_count + 1;

Form1->StringGrid1->Cells[0][row\_count] = trains[i].get\_date();

Form1->StringGrid1->Cells[1][row\_count] = trains[i].get\_time();

Form1->StringGrid1->Cells[2][row\_count] =

trains[i].get\_point();

Form1->StringGrid1->Cells[3][row\_count] =

IntToStr(trains[i].get\_space());

}

if (good == false)

Application->MessageBox(L"Подходящих поездов нет", L"Внимание",

MB\_OKCANCEL);

}

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

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

train tr(Form1->datetrain->Date, Form1->timetrain->Time, Form1->Edit1->Text, StrToInt(Form1->Edit2->Text));

trains.push\_back(tr);

StringGrid1->RowCount = 0;

int row\_count = StringGrid1->RowCount;

for (int i = 0; i < trains.size(); i++,row\_count++) {

StringGrid1->RowCount = row\_count + 1;

Form1->StringGrid1->Cells[0][row\_count] = trains[i].get\_date();

Form1->StringGrid1->Cells[1][row\_count] = trains[i].get\_time();

Form1->StringGrid1->Cells[2][row\_count] =

trains[i].get\_point();

Form1->StringGrid1->Cells[3][row\_count] =

IntToStr(trains[i].get\_space());

}

}

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

void \_\_fastcall TForm1::Button3Click(TObject \*Sender)

{

StringGrid1->RowCount = 1;

for (int i = 0; i < trains.size(); i++) {

int row\_count = StringGrid1->RowCount;

StringGrid1->RowCount = row\_count + 1;

Form1->StringGrid1->Cells[0][row\_count] = trains[i].get\_date();

Form1->StringGrid1->Cells[1][row\_count] = trains[i].get\_time();

Form1->StringGrid1->Cells[2][row\_count] =

trains[i].get\_point();

Form1->StringGrid1->Cells[3][row\_count] =

IntToStr(trains[i].get\_space());

}

}

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

void \_\_fastcall TForm1::Button4Click(TObject \*Sender)

{

trains.clear();

StringGrid1->RowCount = 1;

}

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

void \_\_fastcall TForm1::Button5Click(TObject \*Sender)

{

if (SaveDialog1->Execute()) {

TStringList\* list = new TStringList();

try {

list->WriteBOM = true; // Добавляем BOM для UTF-8

list->DefaultEncoding = TEncoding::UTF8;

for (int i = 0; i < StringGrid1->RowCount; i++) {

list->Add(StringGrid1->Rows[i]->DelimitedText);

}

list->SaveToFile(SaveDialog1->FileName, TEncoding::UTF8);

ShowMessage(L"Файл успешно сохранён!");

}

\_\_finally {

delete list;

}

}

}

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

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

: TForm(Owner)

{

Form1->StringGrid1->ColCount = 4;

Form1->StringGrid1->Cells[0][0] = L"Дата";

StringGrid1->ColWidths[0] = 140;

Form1->StringGrid1->Cells[1][0] = L"Время";

StringGrid1->ColWidths[1] = 140;

Form1->StringGrid1->Cells[2][0] = L"Место назначения";

StringGrid1->ColWidths[2] = 200;

Form1->StringGrid1->Cells[3][0] = L"Свободных мест";

StringGrid1->ColWidths[3] = 130;

SaveDialog1->Filter = L"Текстовые файлы (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

OpenDialog1->Filter = L"Текстовые файлы (\*.txt)|\*.txt|Все файлы (\*.\*)|\*.\*";

}

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

void \_\_fastcall TForm1::Button6Click(TObject \*Sender)

{

if (OpenDialog1->Execute()) {

TStringList \*fileLines = new TStringList();

try {

fileLines->LoadFromFile(OpenDialog1->FileName, TEncoding::UTF8);

// Очищаем таблицу и список поездов

StringGrid1->RowCount = 1;

trains.clear();

for (int i = 1; i < fileLines->Count; i++) {

StringGrid1->RowCount = StringGrid1->RowCount + 1;

int currentRow = StringGrid1->RowCount - 1;

StringGrid1->Rows[currentRow]->DelimitedText = fileLines->Strings[i];

train tr(

StringGrid1->Cells[0][currentRow],

StringGrid1->Cells[1][currentRow],

StringGrid1->Cells[2][currentRow],

StrToInt(StringGrid1->Cells[3][currentRow])

);

trains.push\_back(tr);

}

ShowMessage(L"Файл успешно загружен!");

}

\_\_finally {

delete fileLines;

}

}

}

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

**Unit1.h**

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

#ifndef Unit1H

#define Unit1H

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

#include <System.Classes.hpp>

#include <Vcl.Controls.hpp>

#include <Vcl.StdCtrls.hpp>

#include <Vcl.Forms.hpp>

#include <Vcl.ComCtrls.hpp>

#include <Vcl.Dialogs.hpp>

#include <Vcl.Grids.hpp>

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

class TForm1 : public TForm

{

\_\_published: // IDE-managed Components

TStringGrid \*StringGrid1;

TButton \*Button1;

TButton \*Button2;

TButton \*Button3;

TButton \*Button4;

TButton \*Button5;

void \_\_fastcall Button4Click(TObject \*Sender);

void \_\_fastcall Button5Click(TObject \*Sender);

void \_\_fastcall Button6Click(TObject \*Sender);

TButton \*Button6;

TOpenDialog \*OpenDialog1;

TSaveDialog \*SaveDialog1;

TLabel \*Label1;

TLabel \*Label2;

TLabel \*Label3;

TLabel \*Label4;

TDateTimePicker \*datetrain;

TDateTimePicker \*timetrain;

TEdit \*Edit1;

TEdit \*Edit2;

void \_\_fastcall Button1Click(TObject \*Sender);

void \_\_fastcall Button2Click(TObject \*Sender);

void \_\_fastcall Button3Click(TObject \*Sender);

private: // User declarations

public: // User declarations

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

};

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

extern PACKAGE TForm1 \*Form1;

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

#endif

**Вывод:** В ходе выполнения лабораторной работы была разработана программа, моделирующая работу информационной системы железнодорожных касс, с акцентом на применение указателя this в объектно-ориентированном программировании. Созданное приложение успешно реализует хранение и обработку данных о доступных железнодорожных рейсах, включая такие параметры как дата выезда, пункт назначения, время отправления и количество свободных мест. Ключевой особенностью реализации стало использование указателя this для доступа к полям объектов внутри методов класса, что обеспечило корректное взаимодействие с текущим экземпляром класса и наглядно продемонстрировало этот важный механизм ООП. Разработанная система успешно обрабатывает пользовательские запросы на резервирование билетов, учитывая все заданные параметры поиска, и выдает соответствующий результат либо сообщение о невозможности выполнения заказа. Практическое применение указателя this в данной работе позволило лучше понять его роль в реализации методов класса и организации доступа к полям объектов.