



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«МИРЭА – Российский технологический университет»
РТУ МИРЭА**

Институт информационных технологий (ИТ)

Кафедра инструментального и прикладного программного обеспечения (ИиППО)

КУРСОВОЙ ПРОЕКТ (РАБОТА)

по дисциплине: Объектно-ориентированное программирование

по профилю: Информатизация организаций

направления профессиональной подготовки: ____09.03.03

Прикладная информатика (академический бакалавриат)

Тема: Разработка приложения “Личный календарь”

Студент (ф.и.о. полностью): Головки Алексей Николаевич

Группа: ____ИНБО-07-18____

Работа представлена к защите _____ (дата)_____/_____/_____

(подпись и ф.и.о. студента)

Руководитель: _____ (ф.и.о., должность, звание. уч. степень)

Работа допущена к защите _____ (дата)_____/_____/_____

(подпись ф.и.о. руководителя)

Оценка по итогам защиты: _____

_____/_____/_____

_____/_____/_____

(подписи, дата, ф.и.о., должность, звание, уч. степень двух преподавателей, принявших защиту)

М. МИРЭА. 2019г.

УДК 004.4

Головко А.Н . Необходимо разработать программу, для ведения записей в соответствии с планируемым списком дел./ Курсовая работа по дисциплине «Объектно-ориентированное программирование» профиля «Информатизация организаций» направления профессиональной подготовки бакалавриата 09.03.03. «Прикладная информатика» (2ой семестр) / руководитель асс Зорина Н.В / кафедра ИиППО Института ИТ РТУ

Целью работы является разработка программы для ведения записей в соответствии с планируемым списком дел.

В рамках работы были разработаны логика ,интерфейс и структура базы данных.В рамках работы также были осуществлены анализ предметной области , разработка и тестирование приложения.

Golovko A.N. It is necessary to develop a program for keeping records in accordance with the planned to-do list. / Course work in the subject-oriented programming discipline of the Informatization of Organizations profile of the undergraduate training 09.03.03. “Applied Informatics” (2nd semester) / Head of Ass. Zorina N.V. / Department of I & PPO, Institute of IT RTU

The aim of the work is to develop a program for keeping records in accordance with the planned to-do list.

As part of the work, the logic, interface and structure of the database were developed. As part of the work, the analysis of the subject area, development and testing of the application were also carried out.

М. МИРЭА. Ин-т ИТ. Каф. ИиППО. 2019г. Головко А.Н.



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

**Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения
(ИиППО)**

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

по дисциплине: Объектно-ориентированное программирование

по профилю: Информатизация организаций

направления профессиональной подготовки: 09.03.03

Прикладная информатика (академический бакалавриат)

Студент: Головко Алексей Николаевич

Группа: ИНБО-07-18

Срок представления к защите 20.05.2019 г.

Руководитель: Старший преподаватель Зорина Наталья Валентиновна

Тема: «Разработка приложения “Личный Календарь”»

Исходные данные:

- 1) Индивидуальное задание на курсовую работу.
- 2) Интегрированная среда разработки: Microsoft Visual Studio 2017.
- 3) Комплект разработчика приложений на языке C++, руководство по языку программирования C++, стандартные библиотеки классов C++.
- 4) Интернет-ресурсы и источники по теме курсовой работы.

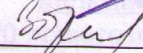
Перечень вопросов к разработке (основная задача), графические материалы (слайды):

- 1) Разработать техническое задание на программный продукт.
- 2) Изучить и научиться применять принципы поэтапной разработки и отладки программ средней сложности.

- 3) Проанализировать предметную область согласно заданию и разработать собственные классы, разработать реализацию классов, а также разработать алгоритмы, необходимые для функционирования программы.
- 4) Разработать интерфейс для программы.
- 5) Разработать тесты для проверки работоспособности программного продукта.
- 6) Разработать руководство по использованию разработанного программного продукта.
- 7) Перечень обязательных графических материалов: UML-диаграмма(ы), Листинги программы.

Руководителем произведён инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Заведующий кафедрой ИиППО:  /В.А. Мордвинов/, дата: _____

Задание на КР выдал  /ст. преп. Н.В. Зорина/

« ____ » _____ 2019 г.

Задание на КР получил  /А.Н.Головко/

« ____ » _____ 2019 г.

Содержание

Содержание	5
Список используемых сокращений	6
Введение	7
1.Анализ программных средств.....	8
2.Методология разработки приложения.....	9
3.Реализация приложения.....	12
4.Тестирование приложения	14
Вывод	15
Список источников	16
Приложение 1	17

Список используемых сокращений

«ОС» - операционная система.

«ПО» - программное обеспечение.

«ООП» - объектно-ориентированное программирование.

«ТЗ» - техническое задание

Введение

С течением времени у каждого из нас появляются определённые дела которые физически трудно запомнить обычному человеку . Поэтому люди начали использовать бумажный ежедневник для записи планов на ближайшее будущее . Однако в период развития информационных технологий он становится все менее удобным средством хранения информации . Есть несколько причин в связи с которыми бумажный ежедневник уходит в прошлое :

1. Редактирование информации выглядит не красиво и занимает лишнее пространство.

2. Бумажный ежедневник имеет свойство заканчиваться.

3. Человек реже забывает телефон/компьютер нежели что-то бумажное.

После создания новых способов хранения данных были созданы первые электронные версии ежедневника , которые в свою очередь, оказались практичнее бумажных . Я выбрал разработку программы ,для ведения записей в соответствии с планируемым списком дел , в качестве курсовой работы.

1. Анализ программных средств

Личный календарь - программа для записи вещей которые необходимо сделать .Идея такой программы не нова , так что существует большое количество разных программ для различных ОС.

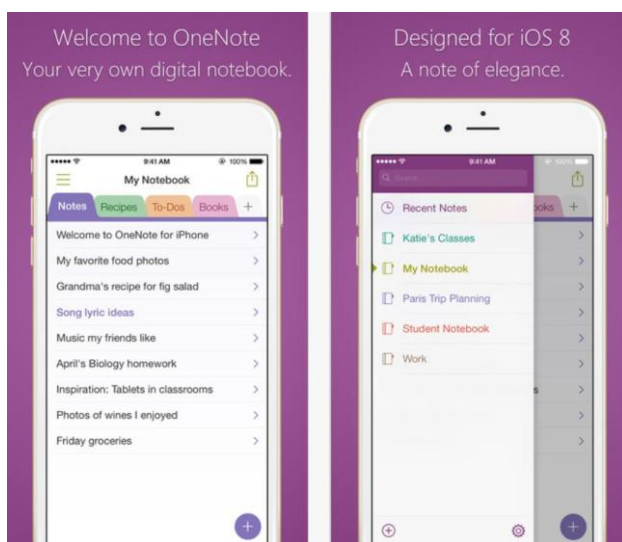
Рассмотрим некоторые из таких программ.

1.1 «OneNote»Microsoft.

Данная программа считается одной из самых мощных в своём роде, из-за очень широкого набора функций:

- 1.голосовой ввод
- 2.общий доступ к документам
- 3.возможность рисовать
- 4.удобный поиск по тегам
5. широкие возможности форматирования текста

Интерфейс программы(карточка 1) достаточно сложен ,однако, удобен в использовании.

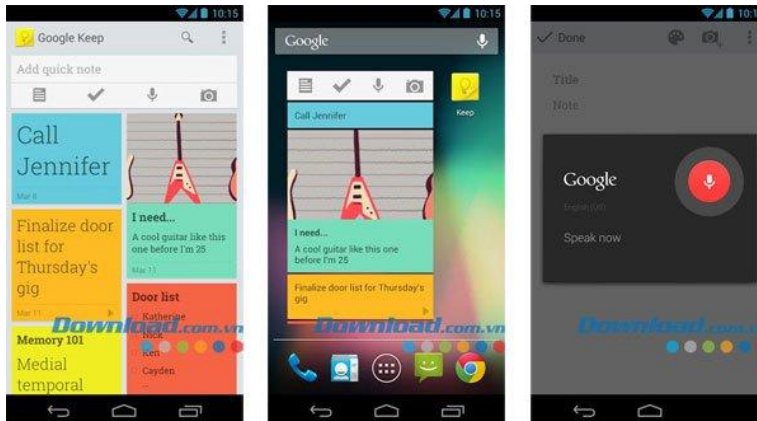


Карточка 1.Интерфейс программы «OneNote»Microsoft.

1.2 «Google Keep» GoogleInc

Блокнот для Android с простым и удобным интерфейсом(карточка 2), имеет ряд интересных функций:

- 1.сохранение любой информации из сети
- 2.поиск заметок и сортировка по цветам
- 3.наличие голосового ввода
- 4.напоминания можно запрограммировать на конкретное место и время



Карточка 2 .Интерфейс программы «Google Keep» GoogleInc

1.3 Выводы к первой главе

На основании рассмотренных выше программ , для реализации приложения «личный календарь» ставятся задачи:

- 1.спроектировать функции которыми должна обладать программа
2. Разработать хорошее ПО с простым интерфейсом
3. Произвести тестировании продукта

2. Методология разработки приложения

2.1 выбор языка программирования для разработки приложения

В качестве языка для разработки ПО был выбран язык C++ из семейства объекту-ориентированных языков , так как в данном семестре ведется изучение этого языка.

2.2 Выбор инструментов для разработки графического интерфейса приложения

2.2.1 Windows Forms

Windows Forms является интерфейсом программирования приложения отвечающим за графический интерфейс пользователя

2.3 Выбор инструментов для разработки функциональной системы приложения

2.3.1 Microsoft Visual Studio

Microsoft Visual Studio - продукт компании Microsoft который позволяет разрабатывать ПО

2.4 Разработка структуры приложения

Программа содержит следующие функции:

- добавление новой заметки
- удаление заметки
- поиск заметки в соответствии с выбранной датой

2.4.1 Базовая структурная схема работы окон программы

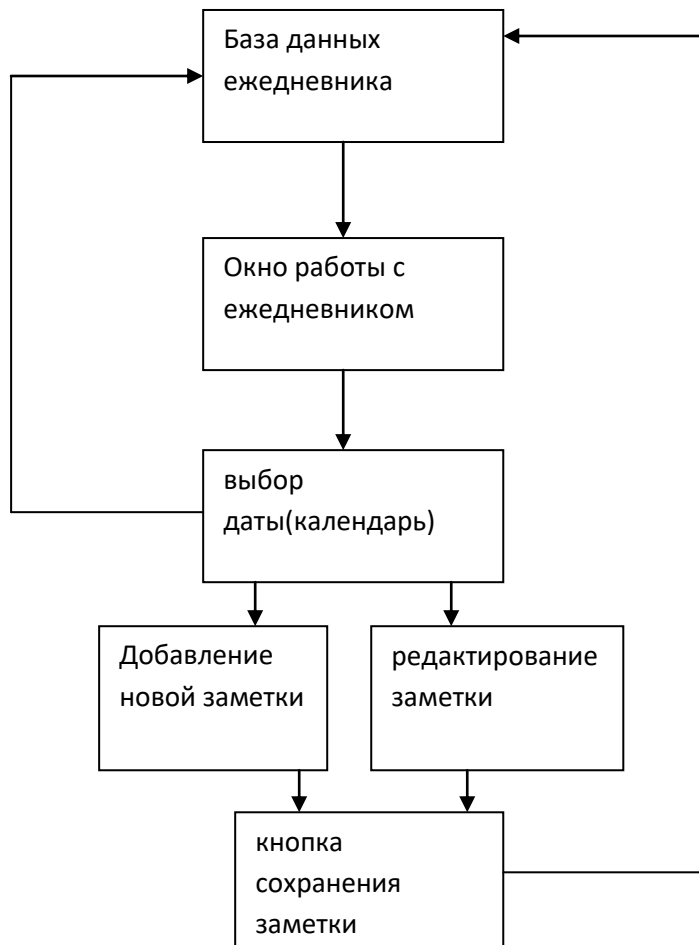


Схема 1. Базовая схема работы программы

После открытия приложения база данных ежедневника соединяется с основным окном работы программы , где и находятся основные функции программы . После вызова функции добавления/редактирования заметки или работы с календарём в базу данных отправляются дополнительные запросы.

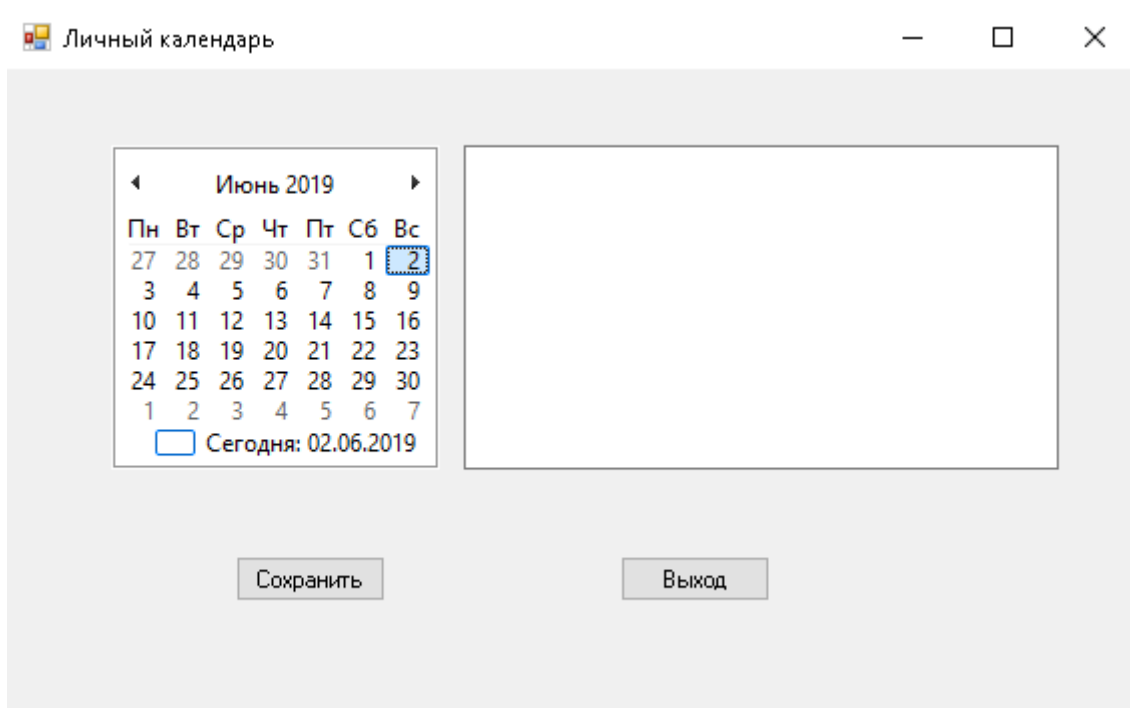
2.4.2 Разработка базы данных для приложения

В базе данных содержатся следующие сведения:

- информация о дате
- заметка

2.5 Интерфейс приложения

2.5.1 главное окно работы с личным календарём



Карточка 3. Главное окно работы с личным календарём

Главное окно работы приложения “личный календарь” содержит календарь , необходимый для выбора даты , пустое окно для добавления новой заметки, и в случае ,если заметка на эту дату уже есть ,то существует возможность редактирования или удаления заметки . Также есть кнопка “сохранить”для сохранения внесенных изменений в базу данных и кнопка “выход” для выхода из программы.

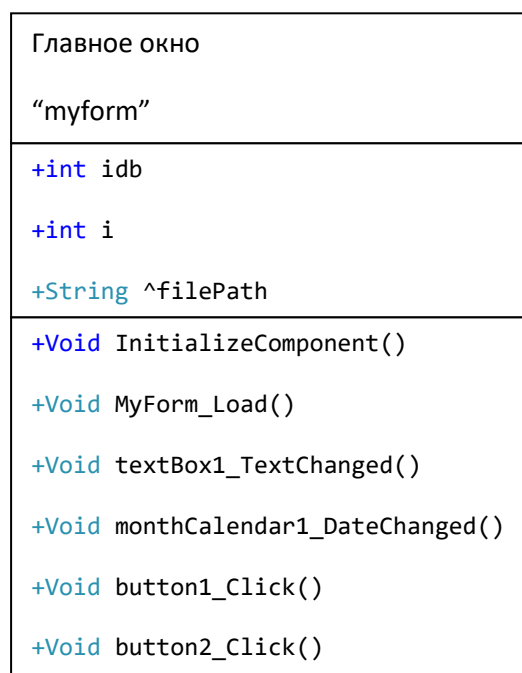
2.6 Выводы ко второй главе

Во второй главе:

1. были рассмотрены программные средства, которые потребуются для разработки ПО
2. была разработана модель базы данных Личного Календаря
3. был определён перечень базовых функций программы
4. была разработана базовая схема работы программы
5. было проведено ознакомление с интерфейсом программы

3. Реализация приложения

3.1 UML-диаграммы классов окон



3.2 Окно работы с ежедневником

Окно работы с ежедневником содержит ряд важных функций

1. Работа с xml файлом, чтение текста

```
private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e) {  
    monthCalendar1->SelectionStart.Today;  
    //чтение из xml  
    String ^filePath = "";  
    DataZametka->ReadXml("dataz.xml");  
}
```

```

        dataGridView1->DataSource = DataZametka;
        dataGridView1->DataMember = "authors";
        dataGridView1->CurrentCell = dataGridView1->Rows[0]->Cells[0]; // при создании
в начало
    }

```

2. функция для записи текста при изменениях в базу данных

```

private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {
    dataGridView1->CurrentRow->Cells[0]->Value = monthCalendar1-
    >SelectionStart.ToShortDateString();
    //записываем дату
    dataGridView1->CurrentRow->Cells[1]->Value = textBox1->Text; //записываем заметку
}

```

3. функция поиска записи для выбранной даты и вывода заметки в текстовое поле

```

private: System::Void monthCalendar1_DateChanged(System::Object^ sender,
System::Windows::Forms::DateRangeEventArgs^ e) {
    //выбор даты
    //поиск записи к выбранной дате
    int i = 0;
    int idr = 0; //номер строки
    //this->bindingSource1->AddNew(); //добавление строки в grid
    idr = dataGridView1->Rows->Count; //количество строк

    for (int i = 0; i < idr; i++) {
        if (dataGridView1->Rows[i]->Cells[0]->Value->ToString()-
        >Contains(monthCalendar1->SelectionStart.ToShortDateString()).Equals(true))
        //проверяем условия
        {
            dataGridView1->CurrentCell = dataGridView1->Rows[i]-
            >Cells[0]; //переход на нужную строку
            this->textBox1->Text = dataGridView1->Rows[i]->Cells[1]->Value-
            >ToString(); //добавляем заметку
            break;
        }
        else
        {
            //this->bindingNavigator1->AddNewItem;
            dataGridView1->CurrentCell = dataGridView1->Rows[dataGridView1-
            >Rows->Count - 1]->Cells[0]; //переход на последнюю
            dataGridView1->Rows[dataGridView1->Rows->Count - 1]->Cells[0]-
            >Value = monthCalendar1->SelectionStart.ToShortDateString(); //записываем дату
            this->textBox1->Text = dataGridView1->CurrentRow->Cells[1]-
            >Value->ToString();
        }
    };
}

```

4. Функция записи текущего состояния с учётом внесённых изменений

```

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    DataZametka->WriteXml("dataz.xml"); //запись файла
}

```

5. Функция закрытия приложения

```

private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    Application::Exit();
}

```


3.6 Выводы к третьей главе

В третьей главе были продемонстрированы uml-диаграммы классов окон и разработаны базовые функции приложения.

4.Тестирование приложения

В ходе разработки ПО были проведены различные тесты при которых находились и исправлялись некоторые ошибки приложения.

Тестирования производились во время всей разработки приложения, различными пользователями на операционной системе Windows 10. Тестовые запуски показали, что приложение:

- Работоспособно.
- Понятно и удобно , благодаря простому интерфейсу программы.
- Функционально. Приложение выполнено полностью в соответствии с ТЗ

4.1Выводы к четвертой главе

В ходе выполнения курсовой работы было разработано полностью соответствующее техническому заданию эффективное, понятное приложение для записи и редактирования заметок в соответствии с датой.

Вывод

По итогам разработки приложения были выполнены следующие задачи

- Была выбрана программная среда
- Была спроектирована логика и интерфейс приложения.
- Для приложения был разработан интуитивно понятный интерфейс, не вызывающий трудностей в работе.
- Все функции приложения, указанные в техническом задании, были реализованы, и работают без ошибок.
- Были произведены тесты, для оценки стабильности и работоспособности приложения.
- Поставленная цель курсовой работы была выполнена.

В результате проведенной работы были изучены и приобретены следующие компетенции:

ОПК-1 - владением основными концепциями, принципами, теориями и фактами, связанными с информатикой;

ПК-14 - готовностью обосновать принимаемые проектные решения, осуществлять постановку и выполнение экспериментов по проверке их корректности и эффективности;

ПК-20 - способностью оценивать временную и емкостную сложность программного обеспечения.

Список источников

1.официальный сайт Microsoft

<https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms?view=netcore-3.0>

2. Язык C++[С.В.Глушаков, А.В.Коваль, С.В.Смирнов](книжное издание)

Обращение к указанным источникам происходило вплоть до финальной стадии разработки приложения.

Приложение 1

Исходный код программного обеспечения

1.1MyForm.cpp

```
#include "MyForm.h"
#include <iostream>
#include <fstream>
using namespace System;
//using namespace std;
using namespace System::Windows::Forms;
//extern const int len = 30;
//extern const int strings = 5;
//extern const char ch = '\n';
//extern char mass[len][strings];
//extern std::fstream fs("data.txt");
[STAThread]
//std::fstream fp("data.txt");

void main(array<String^>^ args) {
    //setlocale(LC_ALL, «rus»);
    Application::EnableVisualStyles();
    Application::SetCompatibleTextRenderingDefault(false);
    Project3::MyForm form;
    Application::Run(%form);
}
```

2.2MyForm.H

```
namespace Project3 {

    using namespace System;

    using namespace System::ComponentModel;

    using namespace System::Collections;

    using namespace System::Windows::Forms;

    using namespace System::Data;

    using namespace System::Drawing;

    /// <summary>

    /// Сводка для MyForm

    /// </summary>
```

```

public ref class MyForm : public System::Windows::Forms::Form
{
public:
    MyForm(void)
    {
        InitializeComponent();

        //
        //TODO: добавьте код конструктора
        //
    }

protected:
    /// <summary>
    /// Освободить все используемые ресурсы.
    /// </summary>
    ~MyForm()
    {
        if (components)
        {
            delete components;
        }
    }

    // календарь для выбора даты
    private: System::Windows::Forms::MonthCalendar^ monthCalendar1;

    //кнопка
    private: System::Windows::Forms::Button^ button1;

    //кнопка
    private: System::Windows::Forms::Button^ button2;

    //
    private: System::Windows::Forms::SaveFileDialog^ saveFileDialog1;

    //

```



```

        private: System::Windows::Forms::TextBox^    textBox1;

//

        private: System::Windows::Forms::DataGridView^    dataGridView1;

//

        private: System::Data::DataSet^    DataZametka;

//

        private: System::Data::DataTable^    dataTable1;

//

        private: System::Data::DataColumn^    dataColumn1;

//

        private: System::Data::DataColumn^    dataColumn2;

//

        private: System::Windows::Forms::DataGridViewTextBoxColumn^
audateDataGridViewTextBoxColumn;

        private: System::Windows::Forms::DataGridViewTextBoxColumn^
au_zametka;

        private: System::Windows::Forms::BindingNavigator^    bindingNavigator1;

        private: System::Windows::Forms::ToolStripButton^
bindingNavigatorAddNewItem;

        private: System::Windows::Forms::ToolStripLabel^
bindingNavigatorCountItem;

        private: System::Windows::Forms::ToolStripButton^
bindingNavigatorDeleteItem;

        private: System::Windows::Forms::ToolStripButton^
bindingNavigatorMoveFirstItem;

        private: System::Windows::Forms::ToolStripButton^
bindingNavigatorMovePreviousItem;

        private: System::Windows::Forms::ToolStripSeparator^
bindingNavigatorSeparator;

        private: System::Windows::Forms::ToolStripTextBox^
bindingNavigatorPositionItem;

        private: System::Windows::Forms::ToolStripSeparator^
bindingNavigatorSeparator1;

        private: System::Windows::Forms::ToolStripButton^
bindingNavigatorMoveNextItem;

```

```

        private: System::Windows::Forms::ToolStripButton^
bindingNavigatorMoveLastItem;

        private: System::Windows::Forms::ToolStripSeparator^
bindingNavigatorSeparator2;

        private: System::Windows::Forms::BindingSource^ bindingSource1;

        private: System::ComponentModel::IContainer^ components;

        protected:

        private:

            /// <summary>
            /// Обязательная переменная конструктора.
            /// </summary>

#pragma region Windows Form Designer generated code

            /// <summary>
            /// Требуемый метод для поддержки конструктора – не изменяйте
            /// содержимое этого метода с помощью редактора кода.
            /// </summary>

            void InitializeComponent(void)
            {

                this->components = (gcnew
System::ComponentModel::Container());

                System::ComponentModel::ComponentResourceManager^
resources = (gcnew System::ComponentModel::ComponentResourceManager(MyForm::typeid));

                this->monthCalendar1 = (gcnew
System::Windows::Forms::MonthCalendar());

                this->button1 = (gcnew System::Windows::Forms::Button());

                this->button2 = (gcnew System::Windows::Forms::Button());

                this->saveFileDialog1 = (gcnew
System::Windows::Forms::SaveFileDialog());

                this->textBox1 = (gcnew System::Windows::Forms::TextBox());

                this->dataGridView1 = (gcnew
System::Windows::Forms::DataGridView());

```

```

        this->audateDataGridViewTextBoxColumn = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());

        this->au_zametka = (gcnew
System::Windows::Forms::DataGridViewTextBoxColumn());

        this->DataZametka = (gcnew System::Data::DataSet());

        this->dataTable1 = (gcnew System::Data::DataTable());

        this->dataColumn1 = (gcnew System::Data::DataColumn());

        this->dataColumn2 = (gcnew System::Data::DataColumn());

        this->bindingNavigator1 = (gcnew
System::Windows::Forms::BindingNavigator(this->components));

        this->bindingNavigatorAddNewItem = (gcnew
System::Windows::Forms::ToolStripButton());

        this->bindingSource1 = (gcnew
System::Windows::Forms::BindingSource(this->components));

        this->bindingNavigatorCountItem = (gcnew
System::Windows::Forms::ToolStripLabel());

        this->bindingNavigatorDeleteItem = (gcnew
System::Windows::Forms::ToolStripButton());

        this->bindingNavigatorMoveFirstItem = (gcnew
System::Windows::Forms::ToolStripButton());

        this->bindingNavigatorMovePreviousItem = (gcnew
System::Windows::Forms::ToolStripButton());

        this->bindingNavigatorSeparator = (gcnew
System::Windows::Forms::ToolStripSeparator());

        this->bindingNavigatorPositionItem = (gcnew
System::Windows::Forms::ToolStripTextBox());

        this->bindingNavigatorSeparator1 = (gcnew
System::Windows::Forms::ToolStripSeparator());

        this->bindingNavigatorMoveNextItem = (gcnew
System::Windows::Forms::ToolStripButton());

        this->bindingNavigatorMoveLastItem = (gcnew
System::Windows::Forms::ToolStripButton());

        this->bindingNavigatorSeparator2 = (gcnew
System::Windows::Forms::ToolStripSeparator());
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>dataGridView1))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>DataZametka))->BeginInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>dataTable1))->BeginInit();

```

```

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this->bindingNavigator1))->BeginInit();

        this->bindingNavigator1->SuspendLayout();

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this->bindingSource1))->BeginInit();

        this->SuspendLayout();

        //

        // monthCalendar1

        //

        this->monthCalendar1->Location = System::Drawing::Point(53, 38);

        this->monthCalendar1->Name = L"monthCalendar1";

        this->monthCalendar1->TabIndex = 0;

        this->monthCalendar1->DateChanged += gcnew
System::Windows::Forms::DateRangeEventHandler(this,
&MyForm::monthCalendar1_DateChanged);

        this->monthCalendar1->DateSelected += gcnew
System::Windows::Forms::DateRangeEventHandler(this,
&MyForm::monthCalendar1_DateSelected);

        this->monthCalendar1->Enter += gcnew
System::EventHandler(this, &MyForm::monthCalendar1_Enter);

        //

        // button1

        //

        this->button1->Location = System::Drawing::Point(115, 243);

        this->button1->Name = L"button1";

        this->button1->Size = System::Drawing::Size(75, 23);

        this->button1->TabIndex = 1;

        this->button1->Text = L"Сохранить";

        this->button1->UseVisualStyleBackColor = true;

        this->button1->Click += gcnew System::EventHandler(this,
&MyForm::button1_Click);

        //

        // button2

        //

```

```

        this->button2->Location = System::Drawing::Point(307, 243);
        this->button2->Name = L"button2";
        this->button2->Size = System::Drawing::Size(75, 23);
        this->button2->TabIndex = 2;
        this->button2->Text = L"Выход";
        this->button2->UseVisualStyleBackColor = true;
        this->button2->Click += gcnew System::EventHandler(this,
&MyForm::button2_Click);
        //
        // textBox1
        //
        this->textBox1->Location = System::Drawing::Point(229, 38);
        this->textBox1->Multiline = true;
        this->textBox1->Name = L"textBox1";
        this->textBox1->Size = System::Drawing::Size(297, 162);
        this->textBox1->TabIndex = 5;
        this->textBox1->TextChanged += gcnew
System::EventHandler(this, &MyForm::textBox1_TextChanged);
        //
        // dataGridView1
        //
        this->dataGridView1->AutoGenerateColumns = false;
        this->dataGridView1->ColumnHeadersHeightSizeMode =
System::Windows::Forms::DataGridViewColumnHeadersHeightSizeMode::AutoSize;
        this->dataGridView1->Columns->AddRange(gcnew cli::array<
System::Windows::Forms::DataGridViewColumn^ >(2) {
            this->audateDataGridViewTextBoxColumn,
            this->au_zametka
        });
        this->dataGridView1->DataMember = L"authors";
        this->dataGridView1->DataSource = this->DataZametka;
        this->dataGridView1->Location = System::Drawing::Point(12,
202);

```



```

        this->dataGridView1->Name = L"dataGridView1";

        this->dataGridView1->Size = System::Drawing::Size(242,
119);

        this->dataGridView1->TabIndex = 6;

        this->dataGridView1->Visible = false;

        this->dataGridView1->CellContentClick += gcnew
System::Windows::Forms::DataGridViewCellEventHandler(this,
&MyForm::dataGridView1_CellContentClick);

        //
        // audateDataGridViewTextBoxColumn
        //
        this->audateDataGridViewTextBoxColumn->DataPropertyName =
L"au_date";

        this->audateDataGridViewTextBoxColumn->HeaderText =
L"au_date";

        this->audateDataGridViewTextBoxColumn->Name =
L"audateDataGridViewTextBoxColumn";

        //
        // au_zametka
        //
        this->au_zametka->DataPropertyName = L"au_zametka";
        this->au_zametka->HeaderText = L"au_zametka";
        this->au_zametka->Name = L"au_zametka";

        //
        // DataZametka
        //
        this->DataZametka->DataSetName = L"DataZametka";

        this->DataZametka->Tables->AddRange(gcnew cli::array<
System::Data::DataTable^>(1) { this->dataTable1 });

        //
        // dataTable1
        //

        this->dataTable1->CaseSensitive = false;

```

```

        this->dataTable1->Columns->AddRange(gcnew cli::array<
System::Data::DataColumn^ >(2) { this->dataColumn1, this->dataColumn2 });

        this->dataTable1->TableName = L"authors";

        //
        // dataColumn1
        //

        this->dataColumn1->ColumnName = L"au_date";

        //
        // dataColumn2
        //

        this->dataColumn2->ColumnName = L"au_zametka";

        //
        // bindingNavigator1
        //

        this->bindingNavigator1->AddNewItem = this-
>bindingNavigatorAddNewItem;

        this->bindingNavigator1->BindingSource = this-
>bindingSource1;

        this->bindingNavigator1->CountItem = this-
>bindingNavigatorCountItem;

        this->bindingNavigator1->DeleteItem = this-
>bindingNavigatorDeleteItem;

        this->bindingNavigator1->Items->AddRange(gcnew cli::array<
System::Windows::Forms::ToolStripItem^ >(11) {

            this->bindingNavigatorMoveFirstItem,

                this->bindingNavigatorMovePreviousItem, this-
>bindingNavigatorSeparator, this->bindingNavigatorPositionItem, this-
>bindingNavigatorCountItem,

                    this->bindingNavigatorSeparator1, this-
>bindingNavigatorMoveNextItem, this->bindingNavigatorMoveLastItem, this-
>bindingNavigatorSeparator2,

                        this->bindingNavigatorAddNewItem, this-
>bindingNavigatorDeleteItem

                });

        this->bindingNavigator1->Location =
System::Drawing::Point(0, 0);

```

```

        this->bindingNavigator1->MoveFirstItem = this-
>bindingNavigatorMoveFirstItem;

        this->bindingNavigator1->MoveLastItem = this-
>bindingNavigatorMoveLastItem;

        this->bindingNavigator1->MoveNextItem = this-
>bindingNavigatorMoveNextItem;

        this->bindingNavigator1->MovePreviousItem = this-
>bindingNavigatorMovePreviousItem;

        this->bindingNavigator1->Name = L"bindingNavigator1";

        this->bindingNavigator1->PositionItem = this-
>bindingNavigatorPositionItem;

        this->bindingNavigator1->Size = System::Drawing::Size(567,
25);

        this->bindingNavigator1->TabIndex = 7;

        this->bindingNavigator1->Text = L"bindingNavigator1";

        this->bindingNavigator1->Visible = false;

        //

        // bindingNavigatorAddNewItem

        //

        this->bindingNavigatorAddNewItem->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;

        this->bindingNavigatorAddNewItem->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"bindingNavigatorAddNewItem.Image"))));

        this->bindingNavigatorAddNewItem->Name =
L"bindingNavigatorAddNewItem";

        this->bindingNavigatorAddNewItem-
>RightToLeftAutoMirrorImage = true;

        this->bindingNavigatorAddNewItem->Size =
System::Drawing::Size(23, 22);

        this->bindingNavigatorAddNewItem->Text = L"Добавить";

        //

        // bindingSource1

        //

        this->bindingSource1->DataMember = L"authors";

        this->bindingSource1->DataSource = this->DataZametka;

```

```

        this->bindingSource1->DataSourceChanged += gcnew
System::EventHandler(this, &MyForm::bindingSource1_DataSourceChanged);

        //

        // bindingNavigatorCountItem

        //

        this->bindingNavigatorCountItem->Name =
L"bindingNavigatorCountItem";

        this->bindingNavigatorCountItem->Size =
System::Drawing::Size(43, 22);

        this->bindingNavigatorCountItem->Text = L"для {0}";
        this->bindingNavigatorCountItem->ToolTipText = L"Общее
число элементов";

        //

        // bindingNavigatorDeleteItem

        //

        this->bindingNavigatorDeleteItem->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;

        //

        this->bindingNavigatorDeleteItem->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"bindingNavigatorDeleteItem.Image")));

        //

        this->bindingNavigatorDeleteItem->Name =
L"bindingNavigatorDeleteItem";

        this->bindingNavigatorDeleteItem->
RightToLeftAutoMirrorImage = true;

        this->bindingNavigatorDeleteItem->Size =
System::Drawing::Size(23, 22);

        this->bindingNavigatorDeleteItem->Text = L"Удалить";

        //

        // bindingNavigatorMoveFirstItem

        //

        this->bindingNavigatorMoveFirstItem->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;

```

```

        this->bindingNavigatorMoveFirstItem->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"bindingNavigatorMoveFirstItem.Image"))));

        this->bindingNavigatorMoveFirstItem->Name =
L"bindingNavigatorMoveFirstItem";

        this->bindingNavigatorMoveFirstItem-
>RightToLeftAutoMirrorImage = true;

        this->bindingNavigatorMoveFirstItem->Size =
System::Drawing::Size(23, 22);

        this->bindingNavigatorMoveFirstItem->Text = L"Переместить в
начало";

        //

        // bindingNavigatorMovePreviousItem

        //

        this->bindingNavigatorMovePreviousItem->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;

        this->bindingNavigatorMovePreviousItem->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"bindingNavigatorMovePreviousItem.Image"))));

        this->bindingNavigatorMovePreviousItem->Name =
L"bindingNavigatorMovePreviousItem";

        this->bindingNavigatorMovePreviousItem-
>RightToLeftAutoMirrorImage = true;

        this->bindingNavigatorMovePreviousItem->Size =
System::Drawing::Size(23, 22);

        this->bindingNavigatorMovePreviousItem->Text =
L"Переместить назад";

        //

        // bindingNavigatorSeparator

        //

        this->bindingNavigatorSeparator->Name =
L"bindingNavigatorSeparator";

        this->bindingNavigatorSeparator->Size =
System::Drawing::Size(6, 25);

        //

        // bindingNavigatorPositionItem

        //

```



```

        this->bindingNavigatorPositionItem->AccessibleName =
L"Положение";

        this->bindingNavigatorPositionItem->AutoSize = false;

        this->bindingNavigatorPositionItem->Name =
L"bindingNavigatorPositionItem";

        this->bindingNavigatorPositionItem->Size =
System::Drawing::Size(50, 23);

        this->bindingNavigatorPositionItem->Text = L"0";

        this->bindingNavigatorPositionItem->ToolTipText = L"Текущее
положение";

        //

        // bindingNavigatorSeparator1

        //

        this->bindingNavigatorSeparator1->Name =
L"bindingNavigatorSeparator1";

        this->bindingNavigatorSeparator1->Size =
System::Drawing::Size(6, 25);

        //

        // bindingNavigatorMoveNextItem

        //

        this->bindingNavigatorMoveNextItem->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;

        this->bindingNavigatorMoveNextItem->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"bindingNavigatorMoveNextItem.Image"))));

        this->bindingNavigatorMoveNextItem->Name =
L"bindingNavigatorMoveNextItem";

        this->bindingNavigatorMoveNextItem-
>RightToLeftAutoMirrorImage = true;

        this->bindingNavigatorMoveNextItem->Size =
System::Drawing::Size(23, 22);

        this->bindingNavigatorMoveNextItem->Text = L"Переместить
вперед";

        //

        // bindingNavigatorMoveLastItem

        //

```

```

        this->bindingNavigatorMoveLastItem->DisplayStyle =
System::Windows::Forms::ToolStripItemDisplayStyle::Image;

        this->bindingNavigatorMoveLastItem->Image =
(cli::safe_cast<System::Drawing::Image^>(resources-
>GetObject(L"bindingNavigatorMoveLastItem.Image"))));

        this->bindingNavigatorMoveLastItem->Name =
L"bindingNavigatorMoveLastItem";

        this->bindingNavigatorMoveLastItem-
>RightToLeftAutoMirrorImage = true;

        this->bindingNavigatorMoveLastItem->Size =
System::Drawing::Size(23, 22);

        this->bindingNavigatorMoveLastItem->Text = L"Переместить в
конец";

        //
        // bindingNavigatorSeparator2
        //

        this->bindingNavigatorSeparator2->Name =
L"bindingNavigatorSeparator2";

        this->bindingNavigatorSeparator2->Size =
System::Drawing::Size(6, 25);

        //
        // MyForm
        //

        this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

        this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;

        this->ClientSize = System::Drawing::Size(567, 321);

        this->Controls->Add(this->bindingNavigator1);

        this->Controls->Add(this->dataGridView1);

        this->Controls->Add(this->textBox1);

        this->Controls->Add(this->button2);

        this->Controls->Add(this->button1);

        this->Controls->Add(this->monthCalendar1);

        this->FormBorderStyle =
System::Windows::Forms::FormBorderStyle::FixedSingle;

```

```

        this->Name = L"MyForm";

        this->Text = L"Личный календарь";

        this->Load += gcnew System::EventHandler(this,
&MyForm::MyForm_Load);

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>dataGridView1))->EndInit();

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>DataZametka))->EndInit();

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>dataTable1))->EndInit();
        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>bindingNavigator1))->EndInit();

        this->bindingNavigator1->ResumeLayout(false);

        this->bindingNavigator1->PerformLayout();

        (cli::safe_cast<System::ComponentModel::ISupportInitialize^>(this-
>bindingSource1))->EndInit();

        this->ResumeLayout(false);

        this->PerformLayout();

    }

#pragma endregion

private: System::Void MyForm_Load(System::Object^ sender, System::EventArgs^ e) {

    monthCalendar1->SelectionStart.Today;

    //чтение из xml

    String ^filePath = "";

    DataZametka->ReadXml("dataz.xml");

    dataGridView1->DataSource = DataZametka;

    dataGridView1->DataMember = "authors";

    dataGridView1->CurrentCell = dataGridView1->Rows[0]->Cells[0];// при
создании в начало

}

private: System::Void textBox1_TextChanged(System::Object^ sender,
System::EventArgs^ e) {

```

```

        dataGridView1->CurrentRow->Cells[0]->Value = monthCalendar1-
>SelectionStart.ToShortDateString();//записываем дату

        dataGridView1->CurrentRow->Cells[1]->Value = textBox1-
>Text;//записываем заметку
    }

private: System::Void monthCalendar1_DateChanged(System::Object^ sender,
System::Windows::Forms::DateRangeEventArgs^ e) {

    //выбор даты

    //поиск записи к выбранной дате

    int i = 0;

    int idr = 0; //номер строки

    //this->bindingSource1->AddNew(); //добавление строки в grid

    idr = dataGridView1->Rows->Count;//количество строк

    for (int i = 0; i < idr; i++) {

        if (dataGridView1->Rows[i]->Cells[0]->Value->ToString()-
>Contains(monthCalendar1->SelectionStart.ToShortDateString()).Equals(true))
        //проверяем условия

        {

            dataGridView1->CurrentCell = dataGridView1->Rows[i]-
>Cells[0];//переход на нужную строку

            this->textBox1->Text = dataGridView1->Rows[i]->Cells[1]-
>Value->ToString();//добавляем заметку

            break;

        }

        else

        {

            //this->bindingNavigator1->AddNewItem;

            dataGridView1->CurrentCell = dataGridView1-
>Rows[dataGridView1->Rows->Count - 1]->Cells[0];//переход на последнюю

            dataGridView1->Rows[dataGridView1->Rows->Count - 1]-
>Cells[0]->Value = monthCalendar1->SelectionStart.ToShortDateString();//записываем
дату

            this->textBox1->Text = dataGridView1->CurrentRow->Cells[1]-
>Value->ToString();

```

```

        }

};

//перевод курсора на выбранную строку
//отображение заметки выбраной даты или создание пустой строки
}

//кнопка сохранения изменений
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e) {
    DataZametka->WriteXml("dataz.xml");//запись файла
}

//кнопка выхода из программы
private: System::Void button2_Click(System::Object^ sender, System::EventArgs^ e) {
    Application::Exit() ;
}

};

}

```

ОТЗЫВ РУКОВОДИТЕЛЯ КУРСОВОЙ РАБОТЫ (КР/КП)

ПО КАФЕДРЕ ИиППО ИНСТИТУТА ИТ МИРЭА

Настоящий отзыв составлен руководителем КР/КП студента _____ (ф.и.о., группа, шифр) по итогам выполнения им на кафедре ИиППО института ИТ _____ (вид КР) по дисциплине _____ в _____ семестре _____ курса обучения в _____ (академическом бакалавриате, прикладном бакалавриате, специалитете, магистратуре) на основании ФГОС ВО и действующих в РТУ МИРЭА нормативных актов: «Инструкция по организации и проведению курсового проектирования (СМКО МИРЭА 7.5.1/04.И.05)»; «Временное положение о текущем контроле успеваемости и промежуточной аттестации по образовательным программам высшего образования – программам бакалавриата, программам специалитета и программам магистратуры (Положение. СМКО МИРЭА 7.5.1/03.П.08)»; «Рекомендации по оформлению письменных работ обучающимися (СМКО МИРЭА 7.5.1/03.П.69)»; а также Рабочей программы поименованной дисциплины.

Руководителем КР/КП отмечается, что задание на курсовую работу выдано в первые две недели семестра. В это же время со студентом проведено собеседование по вопросам организации, учебно-трудовой дисциплины, графика работ и взаимодействия с руководителем, поставлены, обеспечены информационными ресурсами и обсуждены цель, основная задача, содержание, технологии, информационно-методическое обеспечение, формы и сущности отчётности и её защиты под дифференцированную аттестацию. Проведён инструктаж по соблюдению требований охраны труда, техники безопасности (ТБ), а также правилам внутреннего трудового (учебного) распорядка (ПВР).

За студентом закреплён тьютор-стажёр из числа актива подразделения СНК при кафедре ИиППО, выход в кафедральный тематический мини-портал Кабинета дипломного проектирования и магистерской подготовки при кафедре ИиППО (далее: «Кабинет»); в портале создана соответствующая директория курсовой работы, в распоряжение проектанта предоставлены стартовые информационные консисты тематических библиотек портала, выданы методические указания и типовые шаблоны документального оформления пояснительной записки (ПЗ), относящиеся к конкретной дисциплине – в совокупности позволяющие эффективно использовать средства Онтонет управления образовательным макромедиа контентом, а также всемирных технологий WorkBooks. Студент ознакомлен с ТЗ, задачами, технологическими особенностями, применяемым ПО, сущностью этапа выполнения кафедральных НИР, грантов, текущих научно-информационных-технологических работ по развитию информационно-технологической базы кафедры.

Обозначено и реализовано обретение обучающимся профессиональных компетенций, а именно _____.

(При этом согласно пожеланиям обучающихся предусмотрено групповое/бригадное (до 3 человек)) выполнение КП / КР, причём в задании и ПЗ групповой реализации дифференцирован конкретный вклад каждого участника; каждым проектантом представлена индивидуальная ПЗ).*

Отчётные материалы студента в целом отвечают заданной теме, поставленной задаче, требованиям по оформлению. Предложенные решения обладают полнотой

и качественной глубиной раскрытия изучаемых вопросов, демонстрируя признаки новизны, креативности и меры самостоятельности, а также проявление проектантом дисциплинированности, элементов профессиональной этики, умения работать в коллективе.

Качество, полнота и технико-эстетический уровень оформления ПЗ удовлетворяют типовым требованиям. Проверенная руководителем на работоспособность, соответствие и незашумлённость электронная версия ПЗ в форматах Word и Pdf удовлетворительна.

(Работа в качестве НИРС направляется руководителем для участия в конкурсе НИРС МИРЭА, для участия в конференциях, на опубликование или на депонирование).*

По итогам курсовой работы, представления ПЗ руководитель согласно перечисленным выше нормативам подводит следующие итоги:

1. Качественная сторона выполненных работ - _____ (отл., хор., удовл., неудовл.).
2. Количественная сторона (характеризующая охват, объём, детализацию и т.п.) выполненных работ _____ (отл., хор., удовл., неудовл.).
3. Содержание планово-отчётной документации проектанта _____ (отл., хор., удовл., неудовл.).
4. Качество оформления документации на бумажном носителе и в электронной форме (в том числе, соблюдение требований стандартов и упомянутых выше нормативов) _____ (отл., хор., удовл., неудовл.).
5. Учебно-производственная дисциплина, включая соблюдение установленных сроков выполнения, отчётности и аттестаций, а также норм учебно-профессиональной этики _____ (отл., хор., удовл., неудовл.).

Итоговая оценка, рекомендованная руководителем: _____ (отл., хор., удовл., неудовл.).

Таким образом, курсовая работа успешно и в срок завершено. Курсовая работа рекомендована к защите.

Руководитель: _____ (должность, ф.и.о., дата)

Прим.: «Кабинет» каф. ИиППО м-лы КП / КР на хранение принял»

Подпись отв. лица. _____ Должность. Дата.

Прим.: - * при наличии; отзыв составляется на одном двухстороннем листе, при отсутствии удаляется.

Отчет о проверке на заимствования №1



Автор: lesjka31_01@mail.ru / ID: 6812847

Проверяющий: lesjka31_01@mail.ru / ID: 6812847

Отчет предоставлен сервисом «Антиплагиат» - <http://users.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

№ документа: 1

Последний готовый отчет (pdf)

Начало загрузки: 02.06.2019 20:28:32

Начало проверки: 02.06.2019 20:28:36

Длительность загрузки: 00:00:02

Длительность проверки: 00:00:02

Имя исходного файла: курсовая3132313

Комментарии: не указано

Размер текста: 1008 кБ

Модули поиска: Модуль поиска Интернет

Символов в тексте: 29587

Слов в тексте: 2866

Число предположений: 68

ЗАИМСТВОВАНИЯ

ЦИТИРОВАНИЯ

ОРИГИНАЛЬНОСТЬ

54,78%

0%

45,22%



Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.

Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использованием корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.

Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.

Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.

Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которым шла проверка, по отношению к общему объему документа.

Заимствования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.

Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

№	Доля в отчете	Источник	Ссылка	Актуален на	Модуль поиска
101	42,18%	MyForm7	http://studfiles.ru	27 Июл 2016	Модуль поиска Интернет
102	0%	MyForm5	http://studfiles.ru	27 Июл 2016	Модуль поиска Интернет
103	9,51%	Задание двумерного массива - С++/С# - Киберфорум	http://cyberforum.ru	23 Сен 2018	Модуль поиска Интернет