

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Пермский национальный исследовательский
политехнический университет»**

Электротехнический факультет
Кафедра «Информационные технологии и автоматизированные системы»
направление подготовки: 09.03.04 Программная инженерия

О Т Ч Е Т
по учебной практике

Выполнила студентка гр. РИС-19-16
Касимова А. Д.

(подпись)

Проверил:

ст. преподаватель Кузнецов Д. Б.

(оценка)

(подпись)

(дата)

Пермь 2021

Содержание

Введение	3
1 Анализ предметной области	4
1.2 Место прохождения практики	4
1.2 Постановка задачи	5
1.3 Инструменты разработки	8
2 Технология реализации	9
2.1 Работа с базой данных	9
2.2 Интерфейс приложения	9
Заключение	13
Список использованных источников	14
Приложение А. Код классов для работы с БД	15
Приложение Б. Код классов для работы с CheckedListBox	17
Приложение В. Код программы	21

Введение

Производственная практика предназначена для закрепления знаний и навыков, полученных за период обучения, а также для получения практических навыков и новых других знаний.

Соответственно, целью производственной практики является ознакомление с будущей специальностью, приобретения первичных профессиональных умений и навыков.

В рамках научно научно-производственной практики были поставлены следующие задачи:

1. анализ места прохождения практики, задачи;
2. реализация алгоритма, программы;

1 Анализ предметной области

1.2 Место прохождения практики

ООО “ДФС” (Диджитал Фьюче Системс) — российская компания, занимающаяся разработкой и интеграцией бизнес решений. Компания специализируется на:

- проектировании и разработке систем управления, систем поддержки принятия решений, информационно-аналитических и учетных систем;
- внедрении готовых бизнес-приложений и их техническая поддержка;
- управлении данными и глубокой аналитике;
- разработке специализированного программного обеспечения и уникальных заказных решений;
- аутсорсинг.

ДФС решает задачи государственной и корпоративной сфер, подробно указанные в таблице 1. [1]

Таблица 1 — Задачи решаемые компанией

Государственная сфера	Корпоративная сфера
<ul style="list-style-type: none">• Комплексная автоматизация деятельности государственных учреждений• Проектное управление• Управление закупками• Создание информационно-аналитических систем• Создание систем сбора и консолидации отчетности• Создание ситуационных центров	<ul style="list-style-type: none">• Мониторинг реализации стратегии и KPI• Оптимизация производственной деятельности• Бизнес-планирование и бюджетирование• Управление инвестиционными программами и проектами• Создание систем сбора и консолидации отчетности• Создание электронного офиса руководителя• Создание информационно-аналитических порталов

1.2 Постановка задачи

На момент прохождения практики одним из разрабатываемых проектов компании являлся информационный портал “ЭПОС. Аналитика и отчетность” — одна из подсистем Электронной Пермской Образовательной Системы. Из названия портала очевидно, что он предназначен для анализа информации, составления или просмотра некой отчетности. Главная страница сайта изображена на рисунке 1.

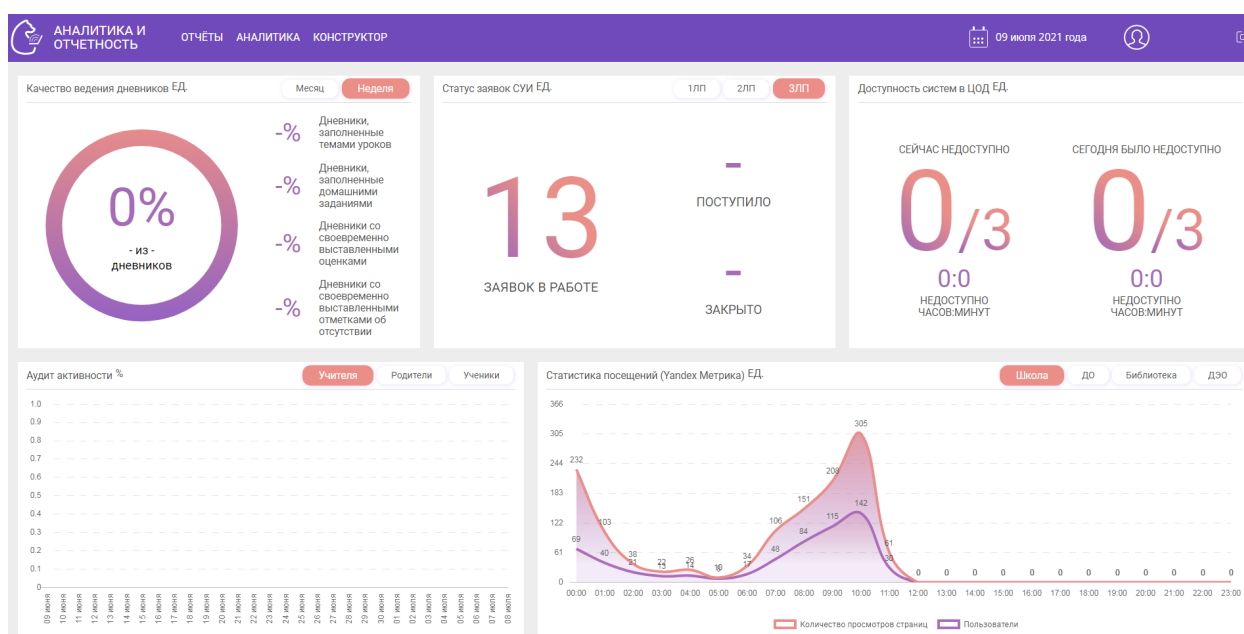


Рисунок 1 — Главная страница портала “ЭПОС. Аналитика и отчетность”

Одним из главным функционалом сайта является конструктор. Он нужен в том случае, если по какой-то причине пользователю недостаточно информации, которая присутствует в отчетах и есть необходимость формировать информацию в других разрезах, с использованием других показателей. Иными словами, настраивать таблицы под конкретные нужды пользователя.

Панель фильтрации конструктора состоит из нескольких блоков:

- Параметры – выбираются показатель/показатели, информация по которым будет отображаться в таблице;
- Столбцы – выбирается, что будет представлено по столбам;
- Строки – выбирается, что будет представлено по строкам;
- Фильтры – выбираются так называемые фиксированные измерения, по которым будет строиться таблица;

Доступные измерения – те измерения, которые можно выбрать в конкретном кубе. Если измерение требуется для работы, необходимо переместить его в какой-то из блоков выше. Если это измерение не нужно, пользователь просто оставляет его в этом блоке. Страница конструктора изображена на рисунке 2.

The screenshot displays a web-based data builder interface. On the left, a sidebar titled 'Панель фильтрации' (Filtering Panel) contains several sections: 'Параметры' (Parameters) with a dropdown menu; 'Столбцы' (Columns) with a 'Класс' (Class) dropdown; 'Строки' (Rows) with 'Территория' (Territory) and 'Школа' (School) dropdowns; 'Фильтры' (Filters); and 'Доступные измерения' (Available Measures) with dropdowns for 'Тип обучения' (Type of education), 'Учебные группы' (Classes), 'Предмет' (Subject), and 'Ученик' (Student). A 'Применить' (Apply) button is at the bottom of the sidebar. The main area on the right, titled 'Показатели качества' (Quality Indicators), currently shows 'нет данных' (no data). An 'Экспорт в Excel' (Export to Excel) button is located in the bottom right corner.

Рисунок 2 — Страница конструктора

Содержимое блоков формируется в зависимости от выбранного куба.

Все манипуляции с настройкой измерений в конструкторе выполняются посредством перетаскивания: пользователь захватывает нужный элемент и далее переносит его в нужный блок. [2]

Работу конструктора можно увидеть на примере. Если в область “Столбцы” перенести измерение “Календарь (Месяц)”, в “Строки” поместить измерения «Учитель» и «Предмет», а в область “Фильтры” поместить параметры “Территория” и “Школа” можно получить показатели успеваемости.

Параметры фильтрации

Параметры

отмечено (3)

Столбцы

Календарь (Месяц)

Сентябрь 2020 г.

Строки

Учитель

Иванов Иван Иванович

Предмет

отмечено (2)

Фильтры

Территория

Лысьвенский городской округ

Школа

МБОУ "СОШ № 6"

Доступные измерения

Тип обучения

Учебные группы

Применить

Рисунок 3 — Заполненные параметры фильтрации

Настроив параметры как на рисунке 3, получим результаты, изображенные на рисунке 4.

АНАЛИТИКА И ОТЧЕТНОСТЬ					2020-2021			еросрег_test.ero...	
ОТЧЕТЫ АНАЛИТИКА КОНСТРУКТОР									
Показатели качества									
ФНО	Продукт	Количество уроков		Количество уроков, для которых предоставлены материалы об изучаемой теме		Количество уроков, для которых предоставлены материалы о деятельности учащихся			
		Сентябрь 2020 г.		Сентябрь 2020 г.		Сентябрь 2020 г.			
Иванов Иван Иванович	Биология	32		32		45			
Иванов Иван Иванович	Химия	35		35		35			

Рисунок 4 — Результат фильтрации

Поскольку конструктор предоставляет возможность выбирать измерения в независимом порядке, пользователь может выбрать сначала школы, потом территорию или наоборот. Задача заключается в том, чтобы пользователь не смог сломать конструктор, а именно, не выбирать значения измерений которые не связаны с другими выбранными. Например, выбрав Пермский городской округ в качестве “Территории”, параметр “Школы” автоматически должен заполняться только пермскими школами, точно также, как если бы были выбраны школы Пермского городского округа, у пользователя не должно быть возможности выбрать другую территорию или классы, не принадлежащие этим школам.

Иными словами, задача заключается в разработке алгоритма двухсторонней фильтрации между параметрами.

1.3 Инструменты разработки

Поскольку задача заключается в разработке алгоритма, программная реализация может быть любой на выбор разработчика. Таким образом, в качестве инструментов для разработки были выбраны язык программирования C#, поскольку на нем реализовано множество полезных библиотек для программирования, в том числе Windows Forms, Npgsql и другие.

2 Технология реализации

2.1 Работа с базой данных

Для работы приложения с базой данных (БД) в проект была подключена библиотека Npgsql. Методы подключения и обработки запросов используют инструменты этой библиотеки и являются атрибутами класса `DataAccessHelper`. SQL запросы формируются с помощью методов класса `QueryGenerator`. Схема обоих классов представлена на рисунке 5.

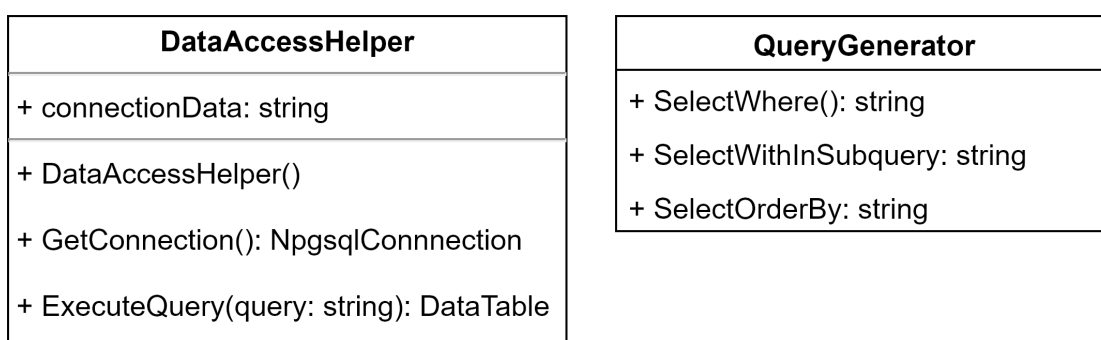


Рисунок 5 — Диаграмма классов

Код классов `DataAccessHelper` и `QueryGenerator` представлен в приложении А.

2.2 Интерфейс приложения

На сайте проекта измерения представляют из себя выпадающий список пунктов, которые при необходимости можно выбрать, поставив галочку, или убрать (рисунок 6).

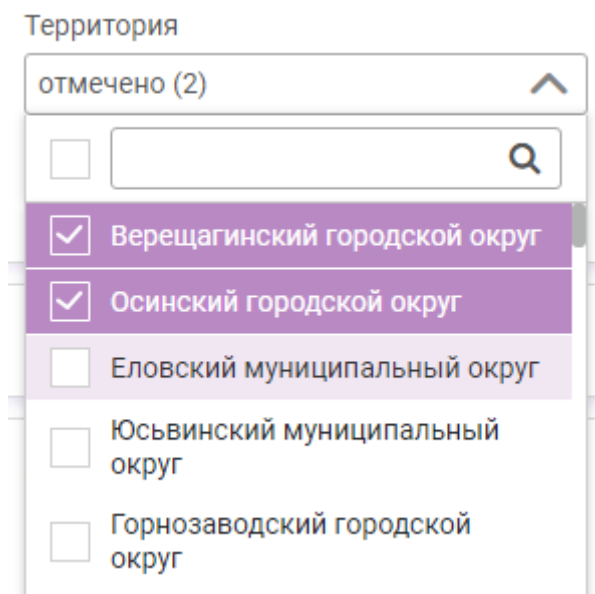


Рисунок 6 — Выпадающий контрольный список

В качестве аналога такому элементу в Windows Forms был выбран класс `checkedListBox`. Класс представляет из себя контейнер содержащий элементы, которые тоже можно выбирать поставив галочку (рисунок 7).

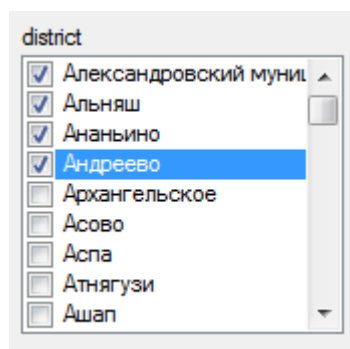


Рисунок 7 — Элемент `checkedListBox`

При выборке элементов одного списка, необходимо, чтобы менялось и содержимое других. Чтобы настроить именно такое поведение формы, на событие класса `checkedListBox` `ItemCheck` (происходит при изменении состояния флажка элемента [3]) нужно подписать методы, фильтрующие другие списки. Методы-подписчики хранятся в соответствующих списках

классам (District, Schools и ClassUnits), которые реализуют интерфейс IFilterState, определяя метод onCheckedListBox_CheckedItems. Алгоритм метода onCheckedListBox_CheckedItems класса District представлен на рисунке 7. Листинг метода представлен в приложении Б.

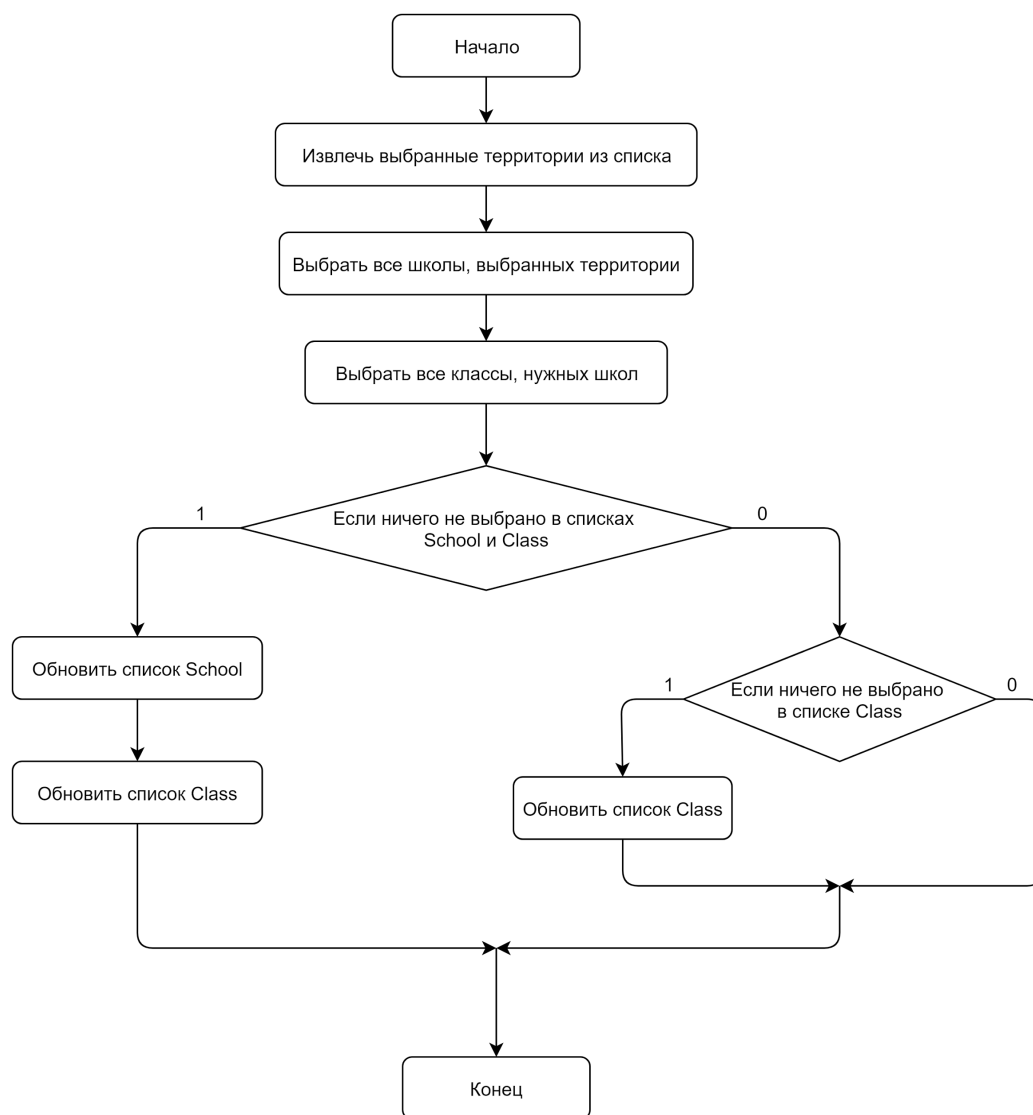


Рисунок 7 — Блок-схема метода onCheckedListBox_CheckedItems

Измерений на самом деле больше чем три (всего их восемь: территория, школа, тип обучения, учебные группы, предмет, ученик и учитель), поэтому был создан интерфейс, который реализовывали бы классы

измерений, переопределяя метод `onCheckedListBox_CheckedItems` по-своему. Интерфейс даёт возможность далее добавлять другие классы в проект.

Заключение

В ходе выполнения работы были реализованы все поставленные задачи, а именно был проведен анализ места прохождения практики, проанализирована задача и были выбраны инструменты разработки. Также был разработан алгоритм решения задачи и в последующем реализован программно.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Digital Future Systems* [Электронный ресурс]. — Режим доступа: <https://dfsystems.ru/> (дата обращения: 09.07.2021)
2. Инструкция по работе с подсистемой Аналитика и отчетность [Электронный ресурс]. — Режим доступа: <https://epos.permkrai.ru/wp-content/uploads/2020/10/instrukciya-po-rabote-s-podsistemoj-analitika-i-otchetnost..docx> (дата обращения: 09.07.2021)
3. *CheckedListBox.ItemCheck* Событие [Электронный ресурс]. — Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/api/system.windows.forms.checkedlistbox.itemcheck?view=net-5.0> (дата обращения: 10.07.2021)

Приложение А. Код классов для работы с БД

Листинг А1 — Класс DataAccessHelper

```
using System;
using Npgsql;
using System.Windows.Forms;
using System.Data;

namespace EScoolTestProject
{
    public class DataAccessHelper
    {
        public string connectionData;

        public DataAccessHelper(string server, string port, string db, string
userId, string password) {
            connectionData = $"Server={server};Port={port};Database={db};User
Id = {userId};Password = {password}";
        }

        public NpgsqlConnection GetConnection()
        {
            NpgsqlConnection conn = new NpgsqlConnection(connectionData);
            return conn;
        }

        public DataTable ExecuteQuery(string query) {
            NpgsqlConnection connection = GetConnection();
            connection.Open();
            NpgsqlCommand command = new NpgsqlCommand(query, connection);
            NpgsqlDataReader datareader = null;
            try
            {
                datareader = command.ExecuteReader();
            }
            catch (Exception exect)
            {
                MessageBox.Show($"при выполнении запроса \"{ query}\" возникла
ошибка: {exect.Message}");
            }
        }
    }
}
```

```

    }
    DataTable dt = new DataTable();
    dt.Load(datareader);
    command.Dispose();
    connection.Close();
    return dt;
}
}
}

```

Листинг A2 — Класс QueryGenerator

```

namespace EScoolTestProject
{
    public class QueryGenerator
    {
        public static string SelectWhere(string columns, string table, string
conditionValue, string conditionTuple) {
            return $"SELECT {columns} FROM {table} WHERE {conditionValue} IN
{conditionTuple}";
        }
        public static string SelectWithInSubquery(string columns, string
table, string conditionValue, string subquery) {
            return $"SELECT {columns} FROM {table} WHERE {conditionValue} IN
({subquery})";
        }
        public static string SelectOrderBy(string columns, string table,
string parameter, string asc_desc="") {
            return $"SELECT {columns} FROM {table} ORDER BY {parameter}
{asc_desc}";
        }
    }
}

```


Приложение Б. Код классов для работы с CheckedListBox

Листинг Б1 — Код интерфейса IFilterState

```
using System;
namespace EScoolTestProject
{
    public interface IFilterState
    {
        void onCheckedListBox_CheckedItems(object sender, EventArgs e);
    }
}
```

Листинг Б2 — Код класса School

```
using System;
using System.Windows.Forms;
using System.Data;

namespace EScoolTestProject
{
    class School : IFilterState
    {
        public Form1 form1;
        public CheckedListBox checkedListBox1;
        public CheckedListBox checkedListBox2;
        public CheckedListBox checkedListBox3;

        public School() { }

        public void onCheckedListBox_CheckedItems(object sender, EventArgs e)
        {
            form1.BeginInvoke((MethodInvoker)delegate
            {
                var selectedSchools = checkedListBox2.CheckedItems;

                string command_checked_districts =
                    QueryGenerator.SelectWhere("district_id", "report.mv_schools", "school_id",
                    CBLHelper.GetCheckedItems(checkedListBox2, "school_id"));
                command_checked_districts =
                    QueryGenerator.SelectWithInSubquery("name, id", "report.district", "id",
                    command_checked_districts);

                string command_checked_classes =
                    QueryGenerator.SelectWhere("name, class_unit_id", "report.mv_class_units",
                    "school_id", CBLHelper.GetCheckedItems(checkedListBox2, "school_id"));

                var collection = checkedListBox1.CheckedItems;
                var collection1 = checkedListBox3.CheckedItems;

                DataAccessHelper dah = new DataAccessHelper("192.168.50.40",
                    "5432", "epos_report", "postgres", "postgres");
                if (collection.Count == 0 && collection1.Count == 0)
                {

```

[illegible]

Листинг Б3 — Код класса District

```

using System;
using System.Data;
using System.Windows.Forms;

namespace EScoolTestProject
{
    class District : IFilterState
    {
        public Form1 form1;
        public CheckedListBox checkedListBox1;
        public CheckedListBox checkedListBox2;
        public CheckedListBox checkedListBox3;

        public District() { }
        public void onCheckedListBox_CheckedItems(object sender, EventArgs e)
        {
            form1.BeginInvoke((MethodInvoker)delegate
            {
                string command_checked_schools =
QueryGenerator.SelectWhere("short_name, school_id", "report.mv_schools",
"district_id", CBLHelper.GetCheckedItems(checkedListBox1, "id"));

                string subquery = QueryGenerator.SelectWhere("short_name",
"report.mv_schools", "district_id",
CBLHelper.GetCheckedItems(checkedListBox1, "id"));
                string command_checked_classes =
QueryGenerator.SelectWithInSubquery("school_id", "report.mv_schools",
"short_name", subquery);
                command_checked_classes =
QueryGenerator.SelectWithInSubquery("name, class_unit_id",
"report.mv_class_units", "school_id", command_checked_classes);

                DataAccessHelper dah = new DataAccessHelper("192.168.50.40",
"5432", "epos report", "postgres", "postgres");
            });
        }
    }
}

```

```

        var collection = checkedListBox2.CheckedItems;
        var collection1 = checkedListBox3.CheckedItems;

        if (collection.Count == 0 && collection1.Count == 0)
        {
            DataTable dt = dah.ExecuteQuery(command_checked_schools);
            DataTable dt1 =
dah.ExecuteQuery(command_checked_classes);

            CBLHelper.FillCheckBoxList(dt, checkedListBox2,
"short_name", "school_id");
            CBLHelper.FillCheckBoxList(dt1, checkedListBox3, "name",
"class_unit_id");
        }
    });
}
}
}

```

Листинг Б4 — Код класса ClassUnit

```

using System;
using System.Windows.Forms;
using System.Data;

namespace EScoolTestProject
{
    class ClassUnit : IFilterState
    {
        public Form1 form1;
        public CheckedListBox checkedListBox1;
        public CheckedListBox checkedListBox2;
        public CheckedListBox checkedListBox3;

        public ClassUnit() { }

        public void onCheckedListBox_CheckedItems(object sender, EventArgs e)
        {
            form1.BeginInvoke((MethodInvoker)delegate
            {
                var selectedClasses = checkedListBox3.CheckedIndices;

                string command_checked_schools =
QueryGenerator.SelectWhere("school_id", "report.mv_class_units",
"class_unit_id", CBLHelper.GetCheckedItems(checkedListBox3,
"class_unit_id"));
                command_checked_schools =
QueryGenerator.SelectWithInSubquery("short_name, school_id",
"report.mv_schools", "school_id", command_checked_schools);

                string command_checked_districts =
QueryGenerator.SelectWhere("school_id", "report.mv_class_units",
"class_unit_id", CBLHelper.GetCheckedItems(checkedListBox3,
"class_unit_id"));
                command_checked_districts =
QueryGenerator.SelectWithInSubquery("district_id", "report.mv_schools",
"school_id", command_checked_districts);
                command_checked_districts =
QueryGenerator.SelectWithInSubquery("name, id", "report.district", "id",

```

```

command_checked_districts);

        var collection = checkedListBox2.CheckedItems;
        var collection1 = checkedListBox1.CheckedItems;

        DataAccessHelper dah = new DataAccessHelper("192.168.50.40",
"5432", "epos_report", "postgres", "postgres");
        if (collection.Count == 0 && collection1.Count == 0)
        {
            DataTable dt = dah.ExecuteQuery(command_checked_schools);
            DataTable dt1 =
dah.ExecuteQuery(command_checked_districts);
            CBLHelper.FillCheckBoxList(dt, checkedListBox2,
"short_name", "school_id");
            CBLHelper.FillCheckBoxList(dt1, checkedListBox1, "name",
"id");
        }
    });
}
}
}
}

```

Приложение В. Код программы

Листинг В1 — Листинга файла Form1.cs

```
using System;
using System.Data;
using System.Windows.Forms;

namespace EScoolTestProject
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            private void Form1_Load(object sender, EventArgs e)
            {
                DataAccessHelper helper = new DataAccessHelper("192.168.50.40",
"5432", "epos_report", "postgres", "postgres");

                DataTable dt =
helper.ExecuteQuery(QueryGenerator.SelectOrderBy("name, id",
"report.district", "name", "ASC"));
                CBLHelper.FillCheckBoxList(dt, checkedListBox1, "name", "id");

                dt =
helper.ExecuteQuery(QueryGenerator.SelectOrderBy("short_name, school_id",
"report.mv_schools", "short_name", "ASC"));
                CBLHelper.FillCheckBoxList(dt, checkedListBox2, "short_name",
"school_id");

                dt = helper.ExecuteQuery(QueryGenerator.SelectOrderBy("name,
class_unit_id", "report.mv_class_units", "class_unit_id"));
                CBLHelper.FillCheckBoxList(dt, checkedListBox3, "name",
"class_unit_id");

                District district = new District();
                district.form1 = this;
                district.checkedListBox1 = checkedListBox1;
                district.checkedListBox2 = checkedListBox2;
                district.checkedListBox3 = checkedListBox3;

                School school = new School();
                school.form1 = this;
                school.checkedListBox1 = checkedListBox1;
                school.checkedListBox2 = checkedListBox2;
                school.checkedListBox3 = checkedListBox3;

                ClassUnit cu = new ClassUnit();
                cu.form1 = this;
                cu.checkedListBox1 = checkedListBox1;
                cu.checkedListBox2 = checkedListBox2;
                cu.checkedListBox3 = checkedListBox3;

                checkedListBox1.ItemCheck +=
district.onCheckedListBox_CheckedItems;
                checkedListBox2.ItemCheck +=
```

```
school.onCheckedListBox_CheckedItems;  
    checkedListBox3.ItemCheck += cu.onCheckedListBox_CheckedItems;  
    }  
}  
}
```