Министерство образования Кировской области

Кировское областное государственное профессиональное образовательное бюджетное учреждение «Кировский авиационный техникум» (КОГПОБУ «Кировский авиационный техникум»)

КУРСОВОЙ ПРОЕКТ

ПМ.04 Обеспечение проектной деятельности

ИФОРМАЦИОННАЯ СИСТЕМА УЧЕТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЦИКЛОВОЙ КОМИССИИ

Пояснительная записка

КП.09.02.05.ИП41.05.01 ПЗ

Выполнил студент А.К. Казаков

Руководитель КП Э.Г. Сандова

Оценка защиты курсового проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата защиты курсового проекта «\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 4](#_Toc63962620)

[**1** **СИСТЕМНЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 6](#_Toc63962621)

[**2** **ТЕХНИЧЕСКИЕ СРЕДСТВА РАЗРАБОТКИ** 8](#_Toc63962622)

[**2.1База данных** 8](#_Toc63962623)

[**2.2Программное обеспечение** 9](#_Toc63962624)

[**2.3Сетевая топология** 10](#_Toc63962625)

[**2.4Операционные системы** 10](#_Toc63962626)

[**3** **ФУНКЦИОНАЛЬНАЯ И ОРГАНИЗАЦИОННАЯ СТРУКТУРА** 11](#_Toc63962627)

[**4** **ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОЙ БАЗЫ** 12](#_Toc63962628)

[**5** **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАЗРАБОТКИ** 16](#_Toc63962629)

[**6** **ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ** 17](#_Toc63962630)

[**ЗАКЛЮЧЕНИЕ** 18](#_Toc63962631)

[ПРИЛОЖЕНИЕ А 19](#_Toc63962632)

[ПРИЛОЖЕНИЕ Б 36](#_Toc63962633)

**ВВЕДЕНИЕ**

Информационная система (ИС) – система, предназначенная для хранения, поиска и обработки информации, и соответствующие организационные ресурсы, которые обеспечивают и распространяют информацию.

ИС предназначена для удовлетворения конкретных информационных потребностей в рамках определенной предметной области, при этом результатом функционирования ИС является информационная продукция – документы, информационные массивы, базы данных и информационные услуги.

Неотъемлемыми компонентами ИС являются данные, техническое и программное обеспечение, а также персонал и организационное обеспечение.

Все ИС делятся на следующие группы:

* Государственные (муниципальные) ИС – создаются с целью реализации полномочий государственных (муниципальных) органов, информационного обмена между ними, а также достижения иных поставленных целей.
* ИС персональных данных – совокупность содержащихся в базах данных персональных данных и обеспечивающих их обработку информационных технологий и технических средств.
* ИС общего пользования – это ИС, в которых содержится общедоступная информация.

В настоящее время информационные системы применяются практически во всех сферах деятельности от простого учета сотрудников до глобальных поисковых систем, позволяющих нам вести поиск разнообразной информации. Одно из таких направлений – образовательные организации.

Современная информационная система для образовательных учреждений представляет сбой комплекс различных программ для автоматизации основных процессов (делопроизводство, ведение личных дел преподавателей, сотрудников, обучаемых, составление расписания, поддержание актуального программного обеспечения).

Основная цель и задача таких ИС – обеспечение необходимой интерактивности и создание единого образовательного информационного пространства.

На курсовой проект было дано задание - разработать информационную систему учета программного обеспечения цикловой комиссии.

Предметно-цикловые комиссии – объединения преподавателей нескольких смежных учебных дисциплин по специальности. Они созданы для учебно-программного, учебно-методического и воспитательного обеспечения освоения учебных дисциплин по специальностям, а также для совершенствования методического и профессионального мастерства преподавателей. Их особым направлением является организация научно-исследовательской деятельности студентов.

Преподавателей цикловых и предметных комиссий в образовательном учреждении может быть много и иногда тяжело бывает не запутаться в том, какой преподаватель в каком кабинете находится и какие предметы он ведет.

Также очень сложно среди большого числа рабочих станций, на которых установлено по несколько программ проконтролировать, где какое программное обеспечение установлено, по какой лицензии и следить за своевременным обновлением лицензии или удалять программное обеспечение, для которого нет обновления лицензии.

Для решения всех этих проблем и создается информационная система учета программного обеспечения цикловой комиссии.

1. **СИСТЕМНЫЙ АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

При выполнении задания на разработку информационной системы учета программного обеспечения цикловой комиссии спроектировать и разработать базу данных преподавателей, кабинетов, ПО и дисциплин.

Она должна обеспечивать поиск, добавление, изменение и удаление, составлять статистику обновления лицензионного программного обеспечения, предупреждать о скором завершении действия лицензии ПО.

При успешном входе в систему пользователи должны видеть всех преподавателей цикловой комиссии, закрепленный за ними кабинет. Должны иметь возможность искать преподавателей по номеру кабинета и, наоборот, номер кабинета по преподавателю, искать преподавателей по дисциплинам, которые он ведет.

Также у пользователей должна быть возможность просматривать установленное ПО, дату установки, срок действия лицензии и где установлено. У них должна быть возможность вести поиск кабинета и номера ПК по названию предмета, находить ПО с определенным типом лицензии, смотреть статистику обновления ПО, искать, к каким дисциплинам относится.

Можно определить три подсистемы:

* Преподаватели
* Программное обеспечение
* Сотрудники

Подсистема «преподаватели» должна содержать в себе информацию о преподавателях (фамилия, имя, отчество, телефон, кабинет, преподаваемые дисциплины), должна быть возможность поиска либо преподавателей по номеру кабинета или по дисциплине, либо наоборот. Идентификация преподавателей идет по номеру сотрудника, номера кабинетов не повторяются. Один преподаватель может вести несколько дисциплин, также одну дисциплину могут вести несколько преподавателей.

Информация о преподавателях должна быть доступна для экспорта с возможностью группировки по дисциплинам или кабинетам.

Добавлять, изменять и удалять информацию смогут только сотрудники с правами администратора.

Подсистема «программное обеспечение» должна содержать в себе информацию об установленном программном обеспечении, типе лицензии, дате установки и дате окончания лицензии. Также у каждого ПО должно быть указано место установки (может быть как одна, так и несколько рабочих станций), при приближении даты окончания действия лицензии должен появиться индикатор и уведомление о необходимости обновления лицензии.

Программное обеспечение должно быть связано с конкретной дисциплиной или дисциплинами.

В данной подсистеме должен быть поиск по наименованию ПО, по типу лицензии, по дисциплинам находить программное обеспечение.

Также в этом подразделе должна вестись статистика обновления программного обеспечения.

Должен быть доступен экспорт информации о программном обеспечении с возможностью группировки по типу лицензии, номерам рабочих станций и дисциплинам.

Добавлять, изменять и удалять информацию смогут только сотрудники с правами администратора.

Подсистема «Сотрудники». В ней должна быть информация обо всех зарегистрированных в системе сотрудниках, указан их уровень прав, указан логин и зашифрованный пароль для входа в систему. К данной подсистеме доступ должен быть только у сотрудников с правами администратора системы, которые смогут изменять, добавлять и удалять информацию.

Пользователями будут:

* Сотрудники с правами администратора системы
* Преподаватели

Администратор может выполнять все операции над системой.

Преподаватели могут просматривать информацию, искать требуемые данные.

Требования к системе:

Информационная система учета программного обеспечения создается для бюджетных образовательных организаций. Поэтому система должна быть экономичной.

Так как информационные системы не могут быть одновременно и производительными с высоким быстродействием, и экономичными, поэтому быстродействие будет снижено для достижения максимальной экономичности.

Так как система в основном ориентирована на учет программного обеспечения, поэтому в ней не будет персональных данных. Но для системы должна быть обеспечена безопасность.

Кроме этого должны быть обеспечены базовые требования системы: надежность и целостность.

1. **ТЕХНИЧЕСКИЕ СРЕДСТВА РАЗРАБОТКИ**

Любая информационная система обычно состоит из двух компонентов: база данных и программное обеспечение для взаимодействия пользователей с базой данных.

## **База данных**

База данных – представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны при помощи ЭВМ. Большинство баз данных создаются, обеспечиваются и обрабатываются при помощи средств СУБД – систем управления базами данных.

СУБД – совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных; комплекс программ, позволяющих создавать базу данных и манипулировать данными. Система обеспечивает безопасность, надежность хранения и целостность данных, а также предоставляет средства администрирования БД.

Примерами таких систем являются:

1. Microsoft Access
2. Microsoft SQL Server
3. Paradox
4. FoxPro
5. Oracle Database
6. Firebird

Microsoft Access – файл-серверная СУБД для работы с локальной базой данных, размещенной на локальном диске.

Преимущества:

1. Проста в использовании
2. Можно создать БД без знания языка запросов.
3. Низкая нагрузка на процессор.

Недостатки

1. Для создания базы данных требуется приобретение лицензии, что нежелательно для снижения затрат;
2. Синхронизация путем файловых блокировок, снижающая производительность системы
3. Потенциально высокая загрузка локальной сети, также снижает производительность
4. Невозможность централизованного управления, а значит, что безопасность данных практически отсутствует из-за множества копий.
5. Ограничение обеспечения надежности, доступности и безопасности

Microsoft SQL Server – клиент-серверная СУБД для работы с базой данных, расположенной на сервере SQL Server

Преимущества:

1. Централизованное управление, повышающее безопасность системы
2. Более низкая нагрузка локальной сети со значительным повышением производительности
3. Высокая надежность, доступность, безопасность
4. Использует распространенный язык запросов Transact SQL

Недостатки:

1. Повышенные требования к серверу, что приведет к дополнительным затратам на оборудование.

## **Программное обеспечение**

Для доступа к информационной системе, не прибегая к непосредственному доступу к базе данных, нужно разработать программную часть информационной системы, дающую доступ к просмотру уже сформированных готовых данных и изменению только необходимых данных.

Разработка такого ПО выполняется при помощи специальных средств разработки с использованием определенных языков программирования.

Одна из наиболее распространенных систем разработки программного обеспечения – Microsoft Visual Studio – включает интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данная программа позволяет создавать приложения с поддержкой технологии Windows Forms – оптимальной технологии для разработки пользовательского интерфейса для информационных систем.

Microsoft Visual Studio поддерживает различные языки программирования, такие как C, C++, C#, Python, Java.

Одним из важных компонентов, обеспечивающих связь и работу программы с базой данных, является модель доступа к данным ADO.NET – модель доступа приложений NET к данным, а также набор библиотек, предназначенных для взаимодействия с различными хранилищами данных из приложения.

Преимущество - возможность работы с отсоединенными источниками данных, представляющих собой структуру организации данных в памяти ОЗУ.

* 1. **Сетевая топология**

Есть разные виды сетевых топологий: звезда, шина, кольцо и ячеистая.

Была выбрана топология звезда, так как снижает вероятность сбоя сети, подключая все узлы к центральному узлу. Это повысит безопасность и надежность системы.

* 1. **Операционные системы**

Операционная система – комплекс программ, обеспечивающий управление аппаратными средствами компьютера, организующий работу с файлами и выполнение прикладных программ, осуществляющий ввод и вывод данных.

Одна из самых распространенных операционных систем – ОС семейства Microsoft Windows:

* Microsoft Windows XP, поддержка прекращена 8 апреля 2014 года
* Microsoft Windows Vista, поддержка прекращена 11 апреля 2017 года
* Microsoft Windows 7, поддержка прекращена 14 января 2020 года
* Microsoft Windows 8, поддержка прекращена 12 января 2016 года
* Microsoft Windows 10, поддерживаемая на данный момент версия операционной системы.

При прекращении дальнейшей поддержки операционных систем значительно повышается риск угроз безопасности, поэтому следует выбирать только поддерживаемые и обновляемые ОС.

Таким образом выбираем Microsoft Windows 10.

1. **ФУНКЦИОНАЛЬНАЯ И ОРГАНИЗАЦИОННАЯ СТРУКТУРА**

Участниками информационной системы учета программного обеспечения будут преподаватели и сотрудники с правами администратора.

Войдя в систему, они должны пройти регистрацию.

Преподаватели должны иметь возможность искать информацию о преподавателях по номеру кабинета, фамилии преподавателя и по преподаваемым предметам. Также могут искать информацию о программном обеспечении, где оно установлено, тип и срок лицензии.

Администраторы имеют все возможности преподавателей, а также имеют возможность добавлять и изменять преподавателей, преподаваемые ими предметы, обновлять программное обеспечение, изменять и добавлять его, а также удалять, если не будет продления лицензии.

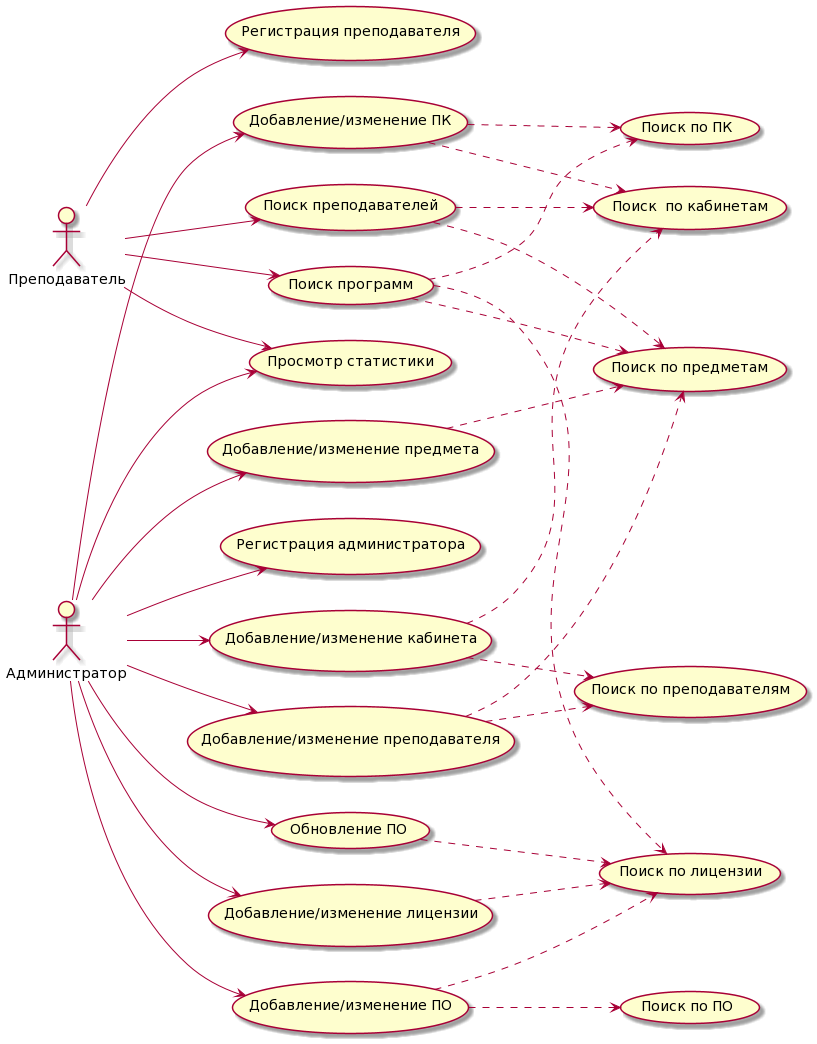


Рисунок 3.1 – Диаграмма прецедентов использования

1. **ОРГАНИЗАЦИЯ ИНФОРМАЦИОННОЙ БАЗЫ**



Рисунок 4.1 – Схема базы данных

Таблица Teachers содержит информацию о преподавателях. Структура представлена на таблице 1.

Таблица 1 – Teachers

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Tnum | Int | Not null | Первичный |  |  | 1 |
| Tfam | Nvarchar | Not null |  | 50 |  | Сандова |
| Tnam | Nvarchar | Not null |  | 50 |  | Эльвира |
| Totch | Nvarchar | Not null |  | 50 |  | Геннадьевна |
| Dolg | Nvarchar | Not null |  | 50 |  | Преподаватель |

Таблица Predmet содержит информацию о предметах. Структура представлена на таблице 2.

Таблица 2 – Predmet

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Pnum | Int | Not null | Первичный |  |  | 1 |
| pnam | Nvarchar | Not null |  | 100 |  | БД |
| Lab\_hours | int | Not null |  |  |  | 85 |
| Spec | Nvarchar | Not null |  | 40 |  | 09.02.05 |

Таблица Prepodav является ассоциативной и содержит информацию о взаимосвязи преподавателей и предметов. Структура представлена в таблице 3.

Таблица 3 – Ассоциация: Teachers – Predmet

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Tnum | Int | Not null | Внешний |  | Teachers.tnum | 1 |
| Pnum | Int | Not null | Внешний |  | Predmet.pnum | 1 |

Таблица Kabs содержит информацию о кабинетах и ответственных за них преподавателях. Структура представлена в таблице 4.

Таблица 4 – Kabs

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Knum | Int | Not null | Первичный |  |  | 1 |
| Knam | Nvarchar | Not null |  | 25 |  | 1/412 |
| Teach | Int | Null | Внешний |  | Teachers.tnum | 1 |

Таблица License содержит информацию о лицензиях. Структура представлена в таблице 5.

Таблица 5 – License

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Lnum | Int | Not null | Первичный |  |  | 1 |
| Lnam | Nvarchar | Not null |  | 100 |  | GNU v2 |
| Dlit | Int | Not null |  |  |  | 3 |

Таблица Programs содержит информацию о программном обеспечении. Структура представлена в таблице 6.

Таблица 6 – Programs

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Prnum | Int | Not null | Первичный |  |  | 1 |
| Prnam | Nvarchar | Not null |  | 100 |  | Visual studio |
| Version | Nvarchar | Not null |  | 50 |  | 2019 |
| License | Int | Not null | Внешний |  | License.lnum | 1 |
| Inst\_dat | Datetime | Not null |  |  |  | 27.01.2021 |
| Expr\_dat | Datetime | Not null |  |  |  | 27.01.2023 |
| Value | Int | Not null |  |  |  | 35 |

Таблица PC содержит информацию о рабочих станциях и их расположении. Структура представлена в таблице 7.

Таблица 7 – PC

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Pcnum | Int | Not null | Первичный |  |  | 1 |
| Pcnet | Nvarchar | Not null |  | 50 |  | 1/412 -1 |
| Kab | Int | Not null | Внешний |  | Kabs.knum | 1 |

Таблица Install является ассоциативной и содержит информацию о взаимосвязи таблиц Programs и PC. Структура приведена в таблице 8.

Таблица 8 – Ассоциация: Programs-PC

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Prog | Int | Not null | Внешний |  | Programs.prnum | 1 |
| Pc | Int | Not null | Внешний |  | PC.pcnum | 1 |

Таблица ProgPred является ассоциативной и содержит информацию о взаимосвязи таблиц Programs и Predmet. Структура приведена в таблице 9.

Таблица 9 – Ассоциация: Programs-Predmet

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Prog | Int | Not null | Внешний |  | Programs.prnum | 1 |
| Predm | Int | Not null | Внешний |  | Predmet.pnum | 1 |

Таблица ProgUpd содержит информации о количестве требуемых обновлений за каждый год. Структура представлена в таблице 10.

Таблица 10 – ProgUpd

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Years | Int | Not null |  |  |  | 2021 |
| All\_Upd | Int | Not null |  |  |  | 15 |
| Upd | Int | Not null |  |  |  | 14 |

Таблица Sotr содержит информацию о пользователях информационной системы. Структура приведена в таблице 11.

Таблица 11 – Sotr

| Наименование полей | Тип данных | Нулевой статус | Ключ | Ограничение | Ссылки | Пример |
| --- | --- | --- | --- | --- | --- | --- |
| Snum | Int | Not null | Первичный |  |  | 1 |
| Snam | Nvarchar | Not null |  | 50 |  | Admin |
| Pass | Nvarchar | Not null |  | 50 |  | Dj2D7KBtdVkl63f3 |
| Role | Nvarchar | Not null |  | 50 |  | Администратор |

Структура связей всех приведенных таблиц показана на рисунке 4.1.

Для большинства таблиц были созданы триггеры на добавление и изменение.

Их код расположен в приложении Б.

1. **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАЗРАБОТКИ**

Поиск при нажатии на ячейку:

//Фильтр ассоциативных таблиц

pr = 0;

if ((sender as DataGridView).SelectedCells.Count > 0)

{

licenseBindingSource.Filter = "lnum = " + Convert.ToInt32((sender as DataGridView).Rows[(sender as DataGridView).SelectedCells[0].RowIndex].Cells[3].Value);

if (licenseBindingSource.Count == 0)

{

licenseBindingSource.Filter = "lnum = 0";

MessageBox.Show("У данной программы отсутствует лицензия");

}

int prog = Convert.ToInt32((sender as DataGridView).Rows[(sender as DataGridView).SelectedCells[0].RowIndex].Cells[0].Value);

installBindingSource.Filter = "prog = " + prog;

//Создание строки фильтра для таблицы ПК и фильтрация

int count = installBindingSource.Count;

string filt;

if (count != 0)//Если программа установлена на ПК

{

filt = "(";

for (int i = 0; i < dataGridView7.Rows.Count; i++)

{

filt += Convert.ToInt32(dataGridView7[1, i].Value) + ", ";

}

if (installBindingSource.Count == 0)

filt = "(0)";

else

filt = filt.Remove(filt.Length - 1 - 1);

pC1BindingSource.Filter = "PCnum in " + filt + ")";

}

else

{

pC1BindingSource.Filter = "PCnum = 0";

MessageBox.Show("Данная программа нигде не установлена");

}

//Создание строки фильтра для таблицы предметов и фильтрация

progPredBindingSource.Filter = "prog = " + prog;

count = progPredBindingSource.Count;

if (count != 0)//Если программа используется в предмете

{

filt = "(";

for (int i = 0; i < dataGridView8.Rows.Count; i++)

{

filt += dataGridView8[1, i].FormattedValue + ", ";

}

filt = filt.Remove(filt.Length - 1 - 1);

predmetBindingSource.Filter = "pnum in " + filt + ")";

}

else

{

predmetBindingSource.Filter = "pnum = 0";

MessageBox.Show("Данная программа нигде не используется");

}

}

else

{

MessageBox.Show("Программа не выбрана");

}

Поиск при вводе текста:

pr = 0;

//Очистка фильтра

programsBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

//Создание фильтра с кодами найденных кабинетов

string filt = "(";

for (int i = 0; i < dataGridView5.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView5.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView5.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены преподаватели

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода кабинета, которого точно не будет

}

//Создание фильтра

programsBindingSource.Filter = "prnum in " + filt + ")";

if (programsBindingSource.Count == 0)

{

MessageBox.Show("Программы, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

pr = 0;

//Очистка фильтра

predmetBindingSource.RemoveFilter();

if ((sender as TextBox).Text != "")

{

if (textBox4.Text == "")

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView2.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()))

{

filt += dataGridView2.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "', не найдены");

}

}

else

{

//Создание фильтра с кодами найденных предметов

string filt = "(";

for (int i = 0; i < dataGridView2.Rows.Count; i++)

{

//Если ячейка содержит введенный текст

if (dataGridView2.Rows[i].Cells[1].FormattedValue.ToString().ToLower().StartsWith((sender as TextBox).Text.ToLower()) && dataGridView2.Rows[i].Cells[3].FormattedValue.ToString().ToLower().StartsWith(textBox4.Text.ToLower()))

{

filt += dataGridView2.Rows[i].Cells[0].FormattedValue + ", ";

}

}

//Если найдены предметы

if (filt.Length > 1)

{

filt = filt.Remove(filt.Length - 1 - 1);//Удаляет последнюю запятую и пробел

}

else

{

filt = "(0";//вводим значение кода предмета, которого точно не будет

}

//Создание фильтра

predmetBindingSource.Filter = "pnum in " + filt + ")";

if (predmetBindingSource.Count == 0)

{

MessageBox.Show("Предметы, содержащие '" + (sender as TextBox).Text + "' и в специальности, содержащей '" + textBox4.Text + "', не найдены");

}

}

}

Обновление ПО:

if (comboBox1.SelectedIndex < 0)

{

MessageBox.Show("Выберите лицензию");

}

else

{

SqlCommand com = new SqlCommand("Update Programs set license = " + Convert.ToInt32(comboBox1.SelectedValue) + ", expr\_dat = '" + dateTimePicker3.Value + "' where prnum = " + Convert.ToInt32(textBox4.Text) + "", con);

com.Connection.Open();

com.ExecuteNonQuery();

com = new SqlCommand("Update ProgUpd set upd = upd + 1 where Years = " + dateTimePicker2.Value.Year, con);

com.ExecuteNonQuery();

com = new SqlCommand("if not exists(select \* from progupd where years = " + dateTimePicker3.Value.Year + ") insert into progupd values (" + dateTimePicker3.Value.Year + ",1,0) else update progupd set all\_upd = all\_upd+1 where years = " + dateTimePicker3.Value.Year, con);

com.ExecuteNonQuery();

com.Connection.Close();

button1.Enabled = false;

MessageBox.Show("Программа обновлена");

}

Руководство оператора находится в приложении А.

1. **ОБЕСПЕЧЕНИЕ БЕЗОПАСНОСТИ ИНФОРМАЦИОННОЙ СИСТЕМЫ**

Любая информационная система, независимо от того, для чего она предназначена, должна обеспечивать качественную безопасность данных. Для защиты информационных систем и информации, хранящейся в них, существует 4 уровня защиты информации:

* Организационный – регламентация средств применения средств автоматизации, систему контроля за деятельностью персонала, мероприятия по защите носителей, внесение изменений в ПО, недопущение несанкционированного копирования данных, борьбу с распространением вирусов в программных средах.
* Технический - реализуется специальными аппаратными средствами для предотвращения повреждения, несанкционированного копирования и использования программных средств.
* Программный – обеспечивается специальными программами в составе ОС и СУБД.
* Криптографический – используется для закрытия информации при ее передаче по каналам связи и хранения в памяти компьютера, а также аутентификации источника информации.

Доступ к информационной системе ограничивается требованием аутентификация пользователя в системе.

Имеется два уровня доступа: администратор и преподаватель.

Пользователь с уровнем доступа администратора имеет полный доступ к системе: добавление и изменение данных, обновление программного обеспечения, имеет прямой доступ к базе данных.

**ЗАКЛЮЧЕНИЕ**

Мной была разработана информационная система учета программного обеспечения цикловой комиссии.

При ее разработке были использованы современные средства разработки приложений на платформе Windows Forms: Visual Studio.

Для раборы с базой данных, ее проектирования, управления и обработки была использована СУБД SQL Server Express.

Был реализован все требуемые функциональные возможности, обеспечена безопасность, надежность и целостность.

ПРИЛОЖЕНИЕ А

Руководство оператора

КУРСОВОЙ ПРОЕКТ

ПМ.04 Обеспечение проектной деятельности

ИФОРМАЦИОННАЯ СИСТЕМА УЧЕТА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ЦИКЛОВОЙ КОМИССИИ

Руководство оператора

КП.09.02.05.ИП41.05.01 Д1

**Аннотация**

Информационная система учета программного обеспечения предназначена поиска и контроля программного обеспечения, лицензий, преподавателей, преподаваемых предметов, кабинетов и рабочих станций.

**ОСНОВНАЯ ЧАСТЬ**

При запуске программы открывается форма входа в систему, где нужно выбрать имя пользователя из списка и ввести пароль.

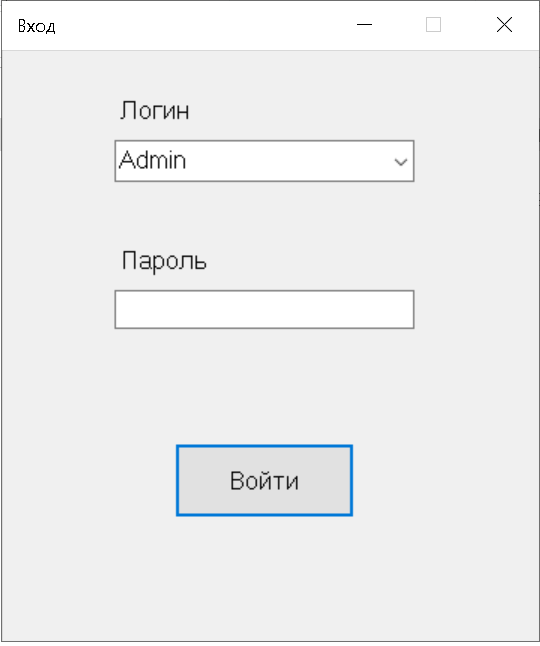


Рисунок А.1

После ввода правильного пароля откроется новое окно, на котором располагаются все основные элементы. Окно откроется в подсистеме «Преподаватели».

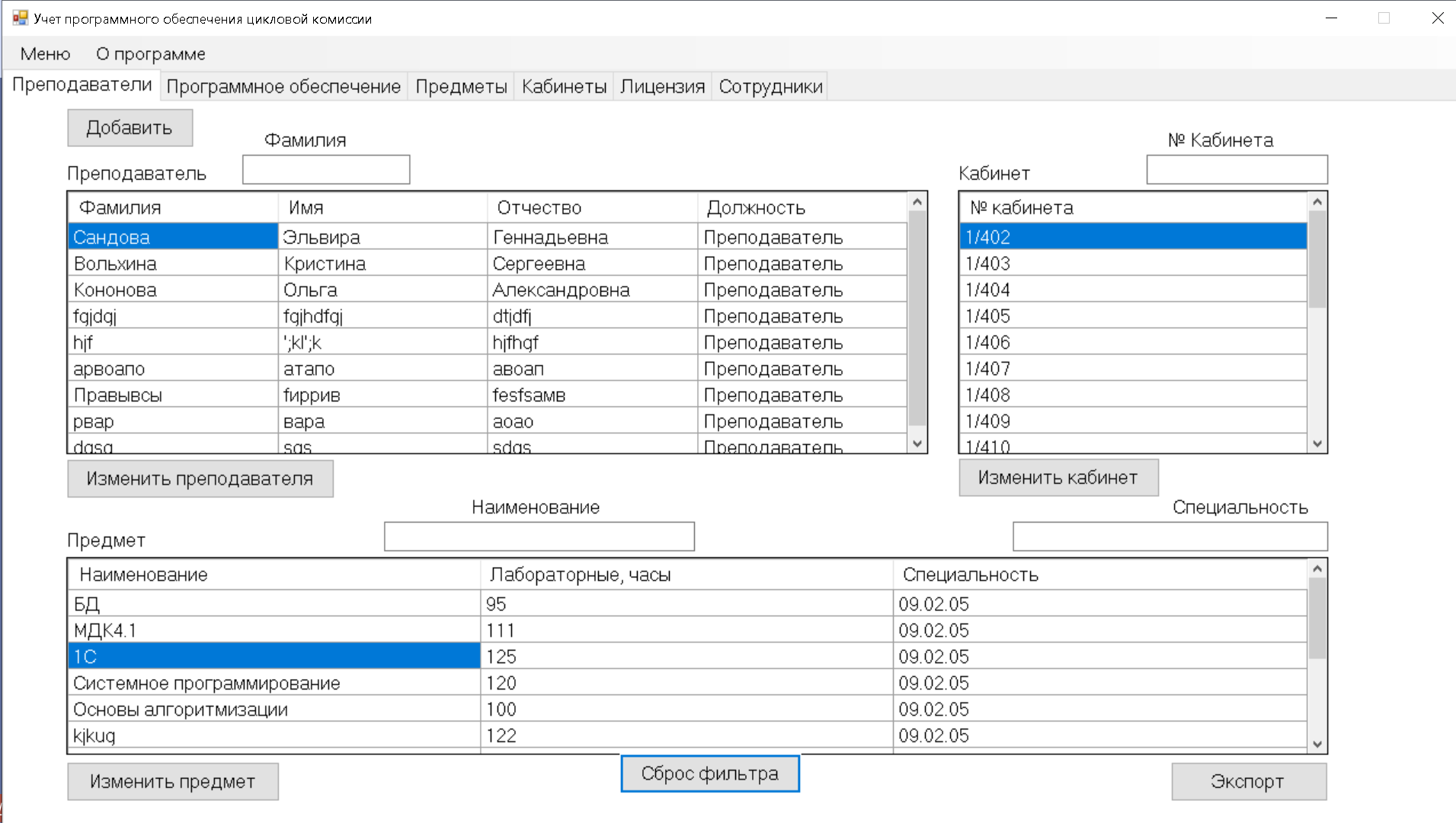


Рисунок А.2

При нажатии на ячейку таблицы «Преподаватели» производится поиск и фильтрация данных, относящихся к выбранному преподавателю, в таблицах «Кабинеты» и «Предметы».

Поле ввода «фамилия» производит поиск по фамилии преподавателей.

Поле ввода «№ кабинета» производит поиск по № кабинетов.

Поля ввода «Наименование» и «Специальность» производят поиск по наименованию предмета и специальности.

Кнопка «Сброс фильтра» сбрасывает фильтры таблиц.

Кнопка «Добавить» открывает новую форму, в которой можно добавить предмет, преподавателя и кабинет.

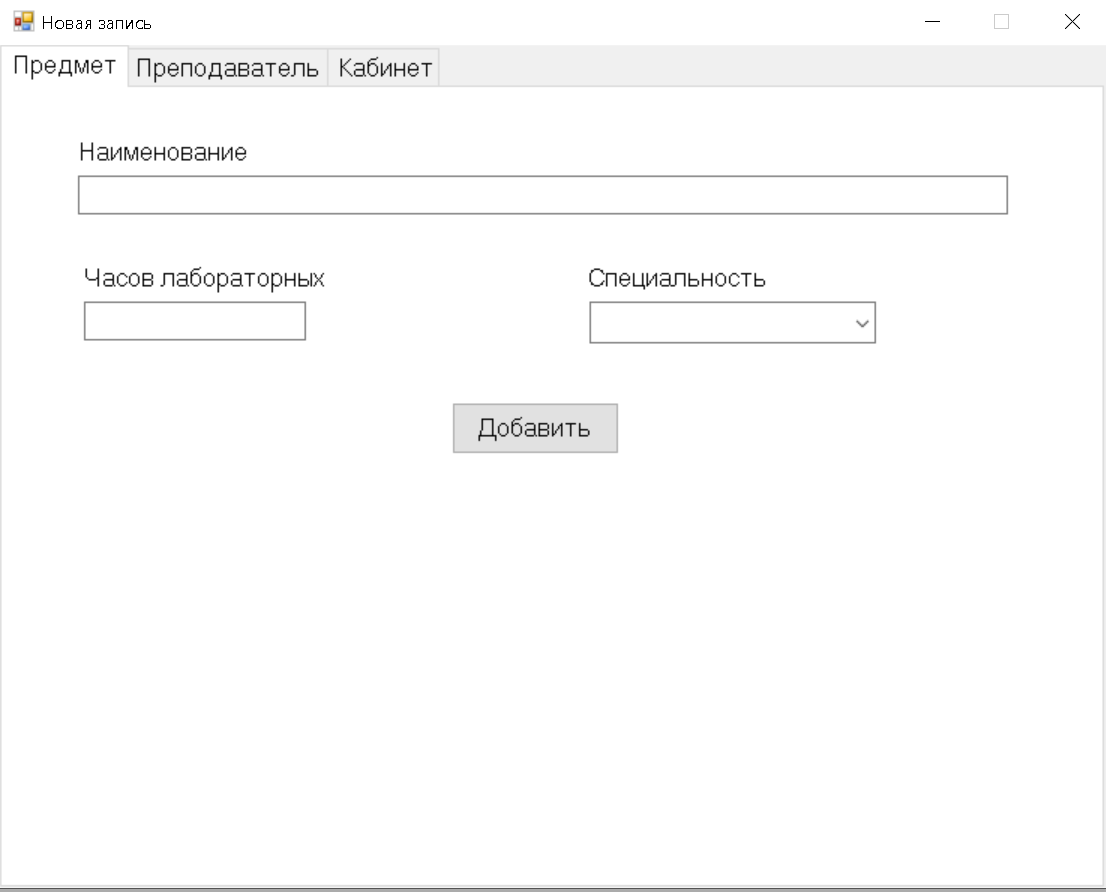


Рисунок А.3

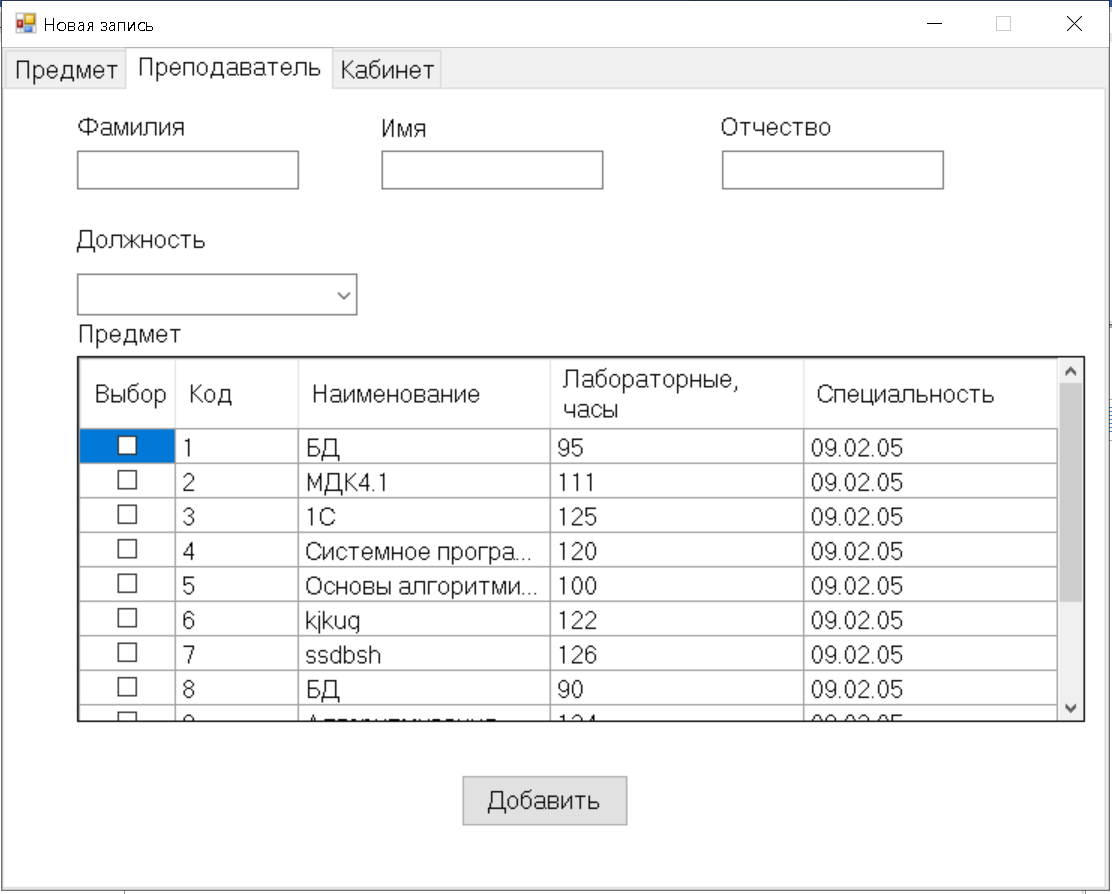


Рисунок А.4

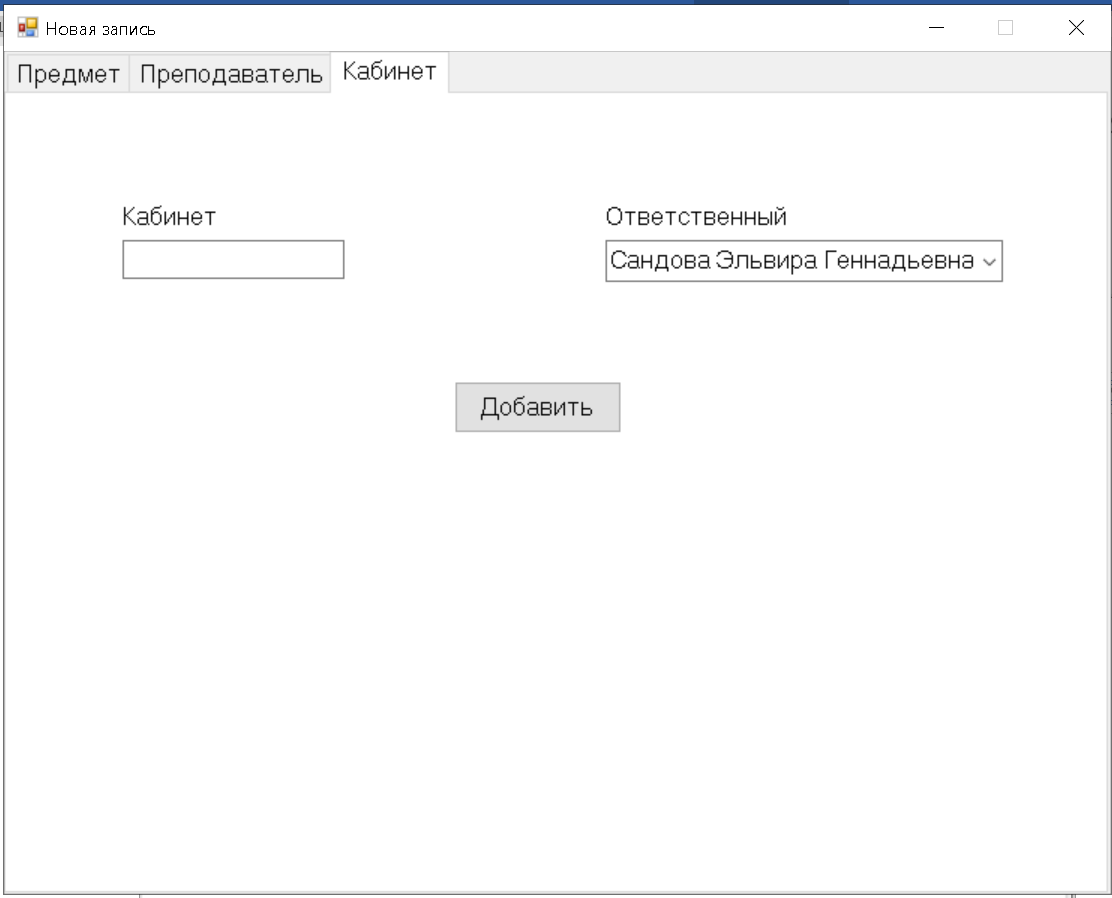


Рисунок А.5

Кнопка «Изменить преподавателя» открывает новое окно, где можно изменить информацию о выбранном преподавателе.

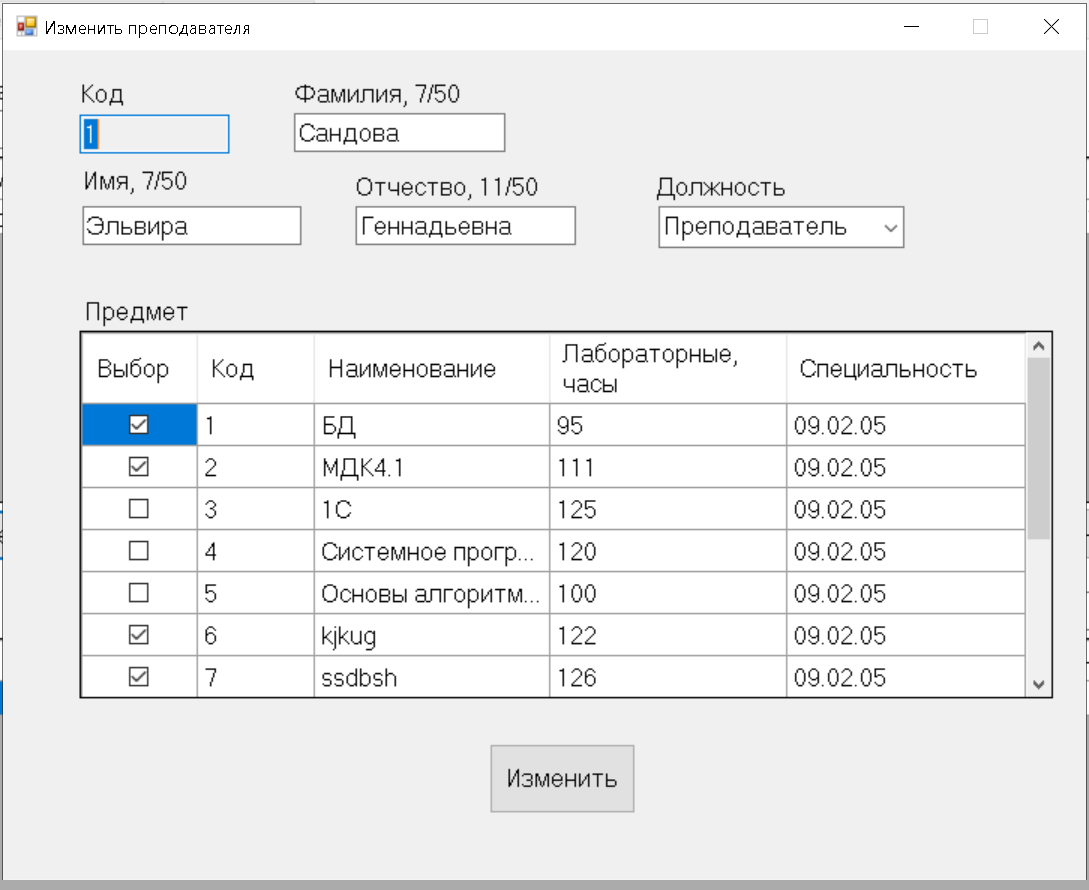


Рисунок А.6

Кнопка «Изменить кабинет» открывает новое окно, где можно изменить информацию о выбранном кабинете.

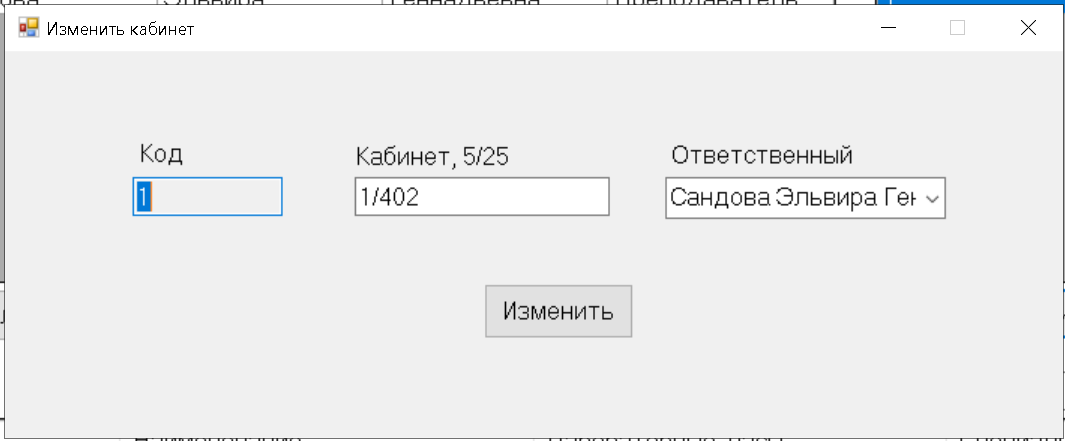


Рисунок А.7

Кнопка «Изменить предмет» открывает новое окно, где можно изменить информацию о выбранном предмете.

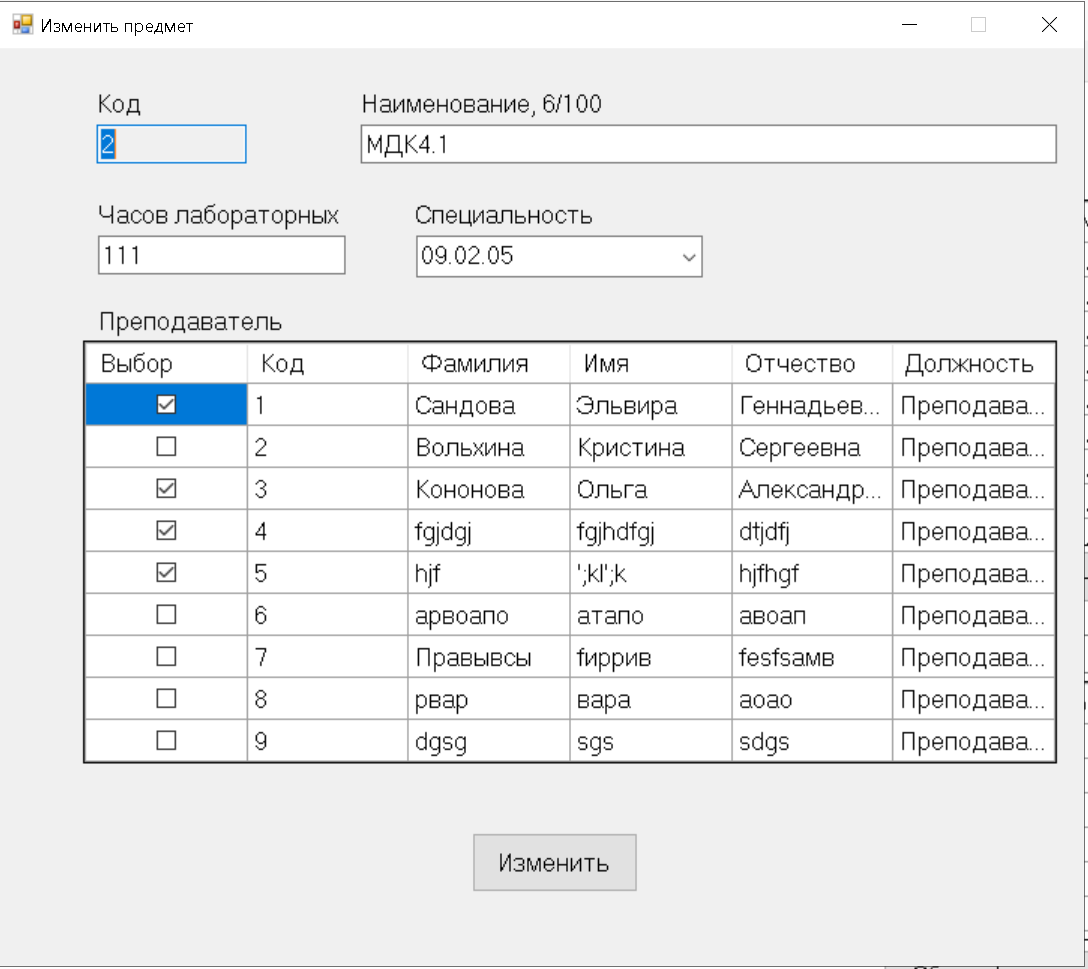


Рисунок А.8

Кнопка «Экспорт» выводит в документ Microsoft WORD информацию о выбранном преподавателе: фамилию, имя, отчество преподавателя, его кабинеты и предметы.

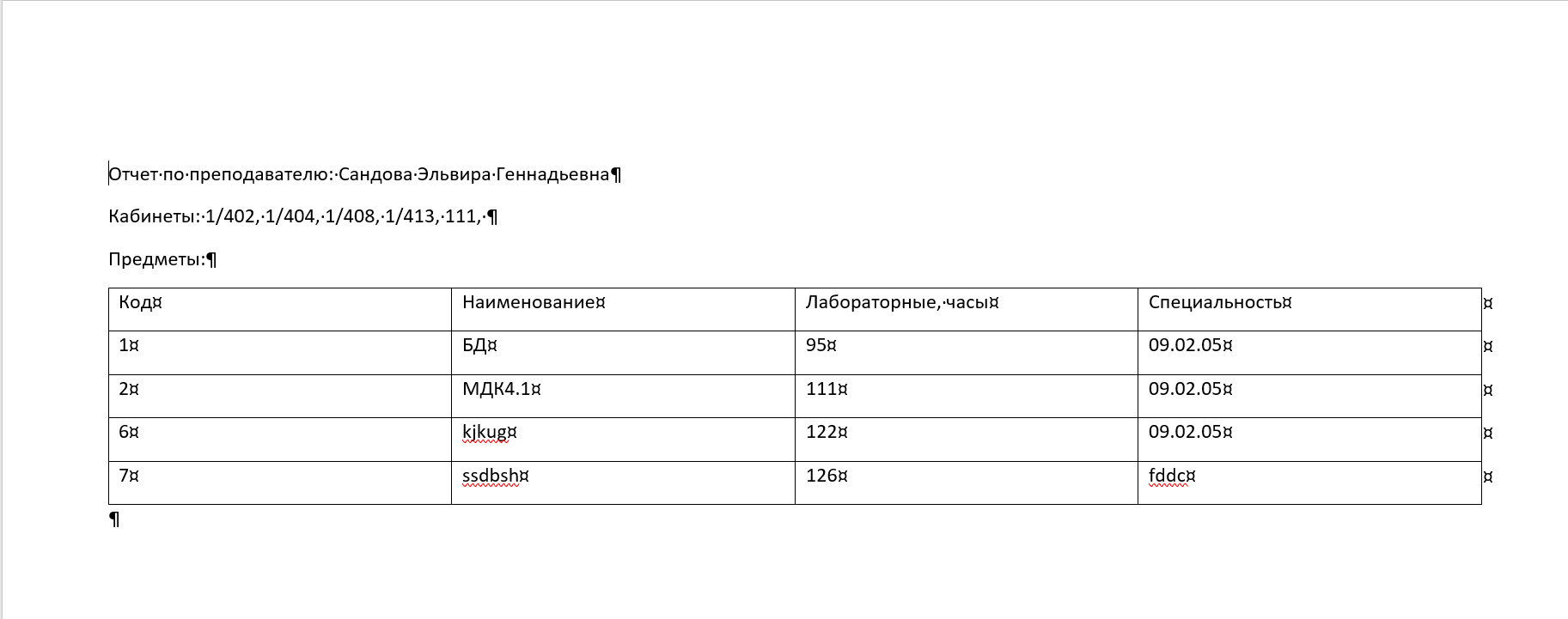


Рисунок А.9

При переходе в подсистему «Программное обеспечение» видим перед собой окно.

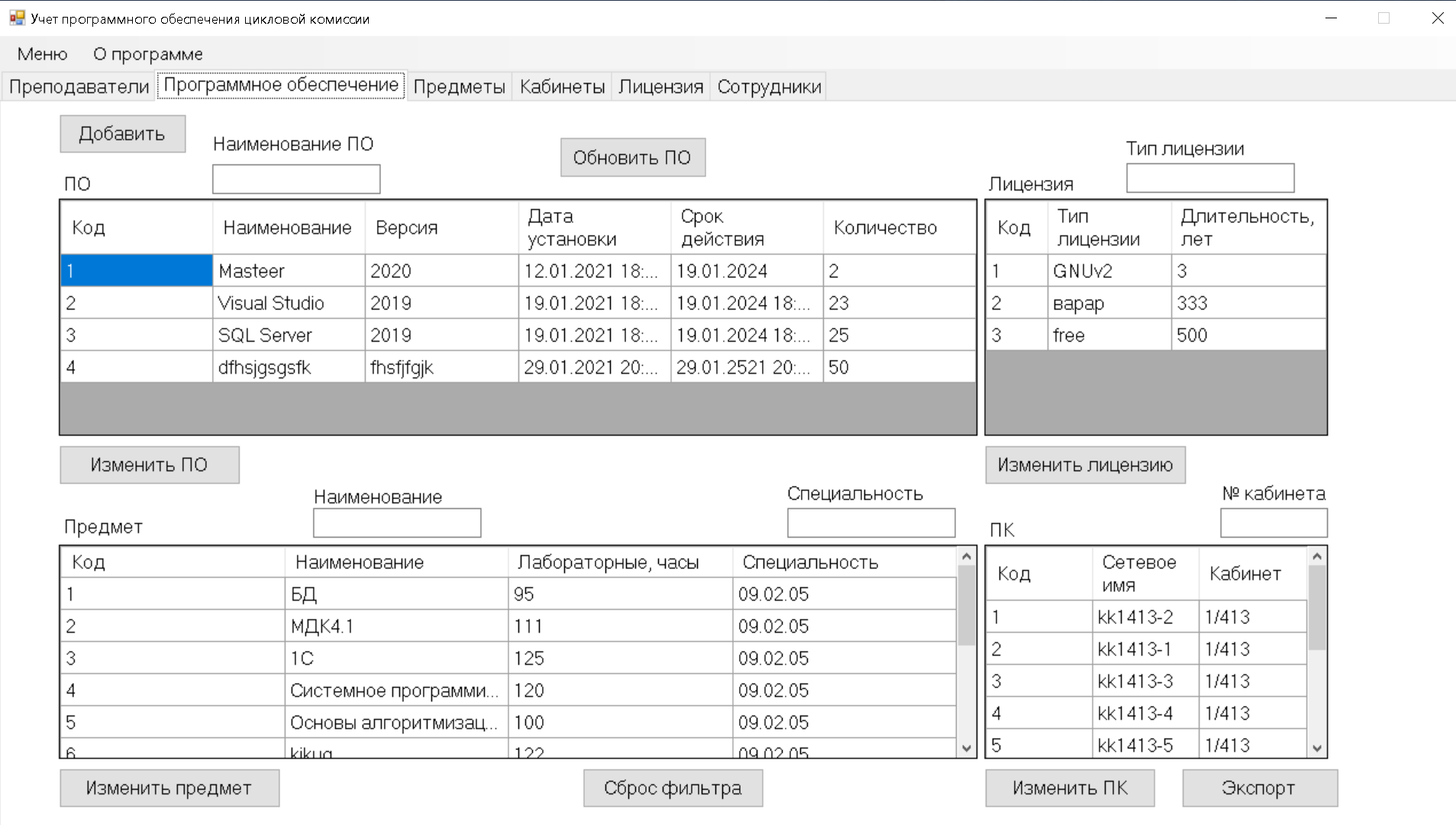


Рисунок А.10

При нажатии на ячейку таблицы «ПО» производится поиск и фильтрация данных, относящихся к выбранному ПО, в таблицах «Предмет», «ПК» и «Лицензия».

Поле ввода «наименование ПО» производит поиск по наименованию программного обеспечения.

Поле ввода «Тип лицензии» производит поиск по лицензии.

Поля ввода «Наименование» и «Специальность» производят поиск по наименованию предмета и специальности.

Поле ввода «№ кабинета» производит поиск ПК по кабинетам.

Кнопка «Добавить» открывает новую форму, в которой можно добавить лицензию, ПК и ПО.

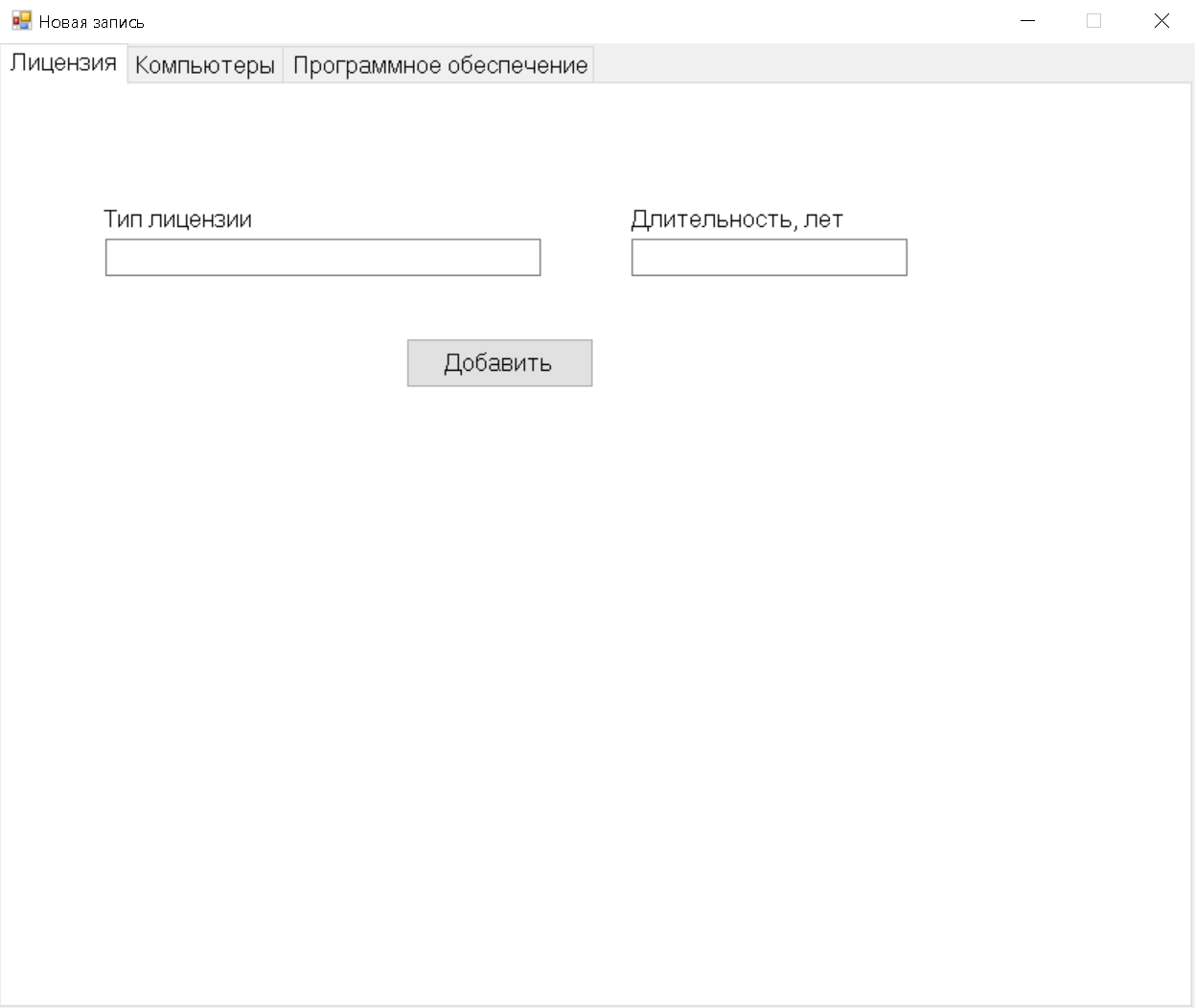


Рисунок А.11

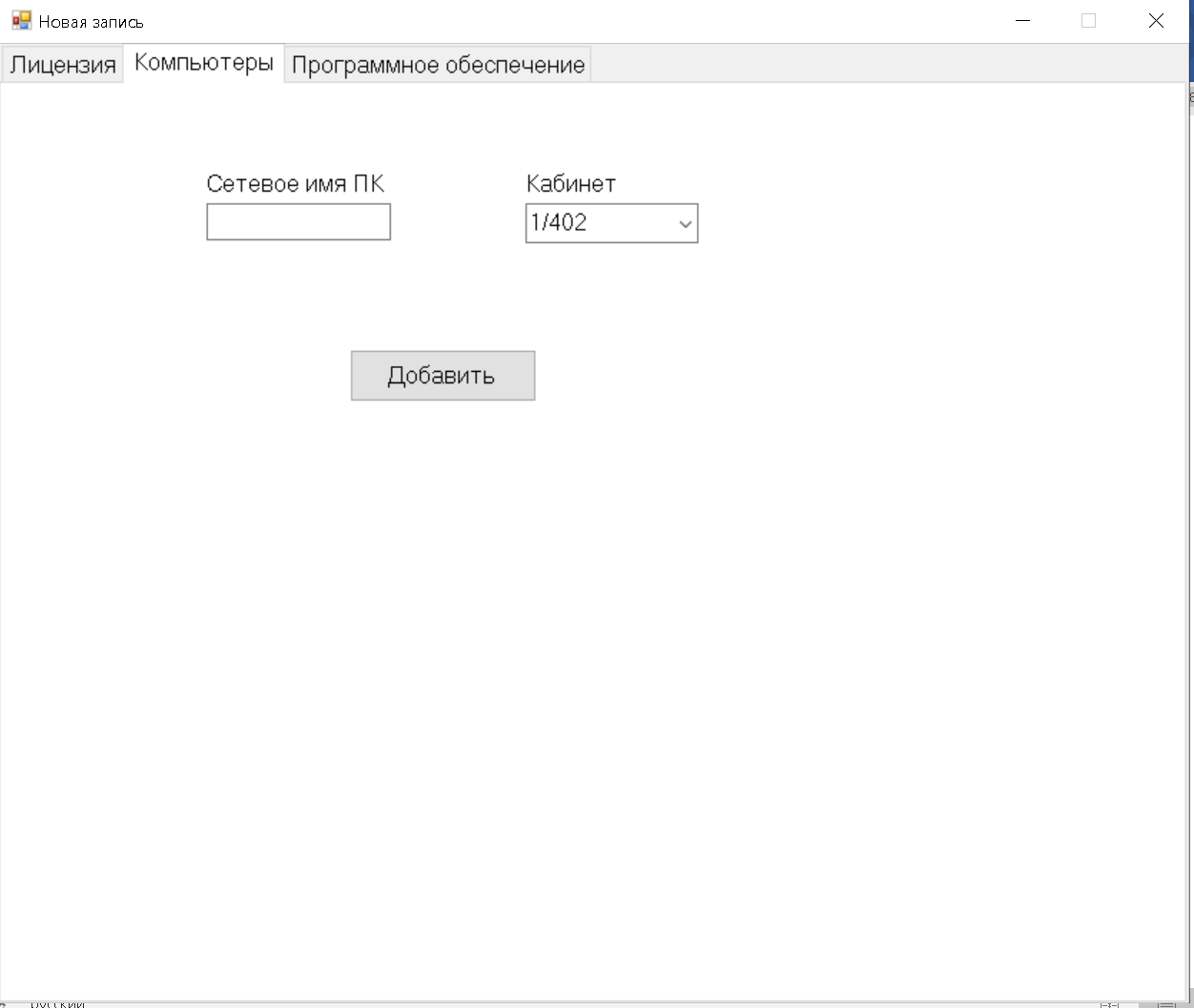


Рисунок А.12

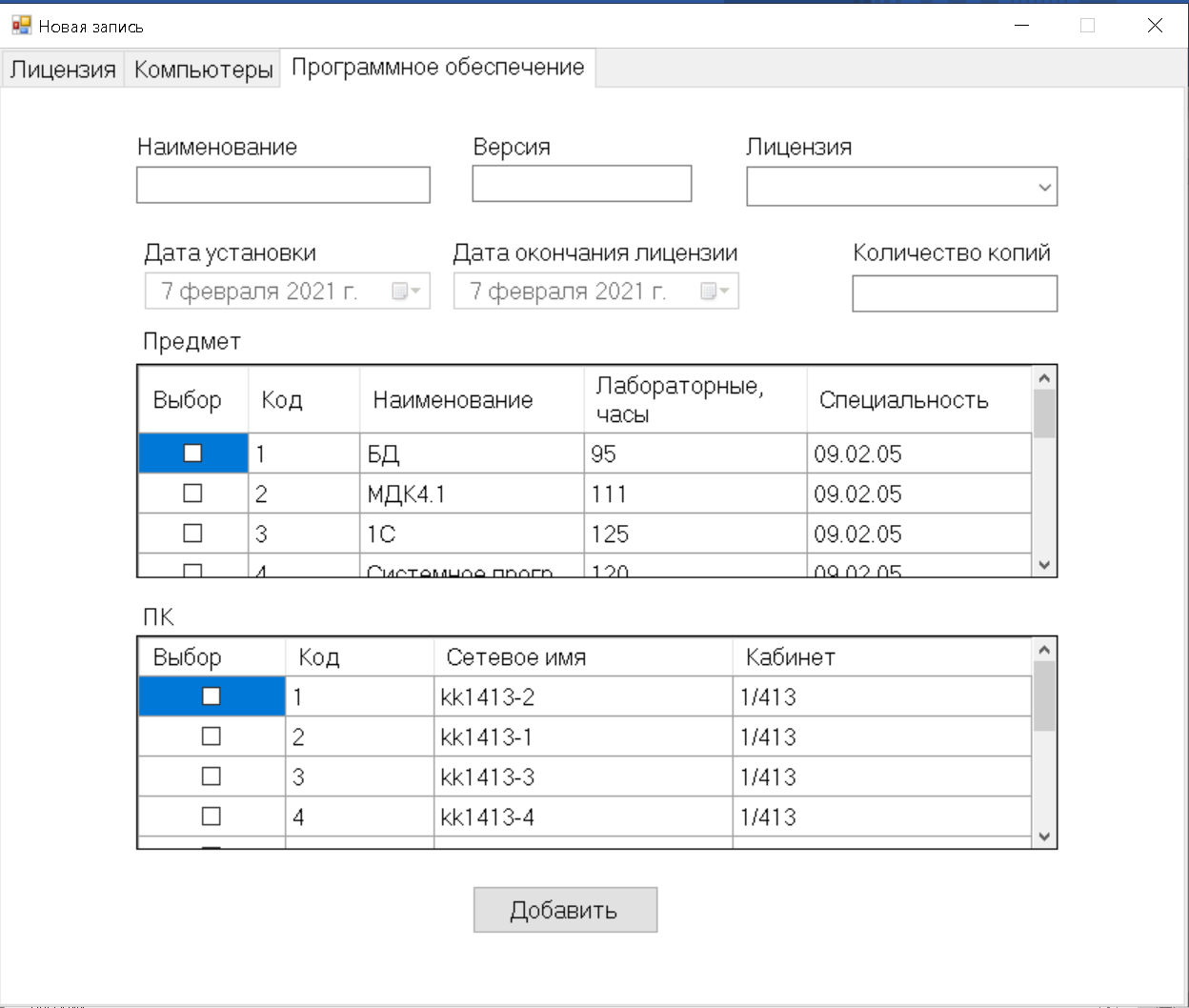


Рисунок А.13

Кнопка «Изменить ПО» открывает новое окно, где можно изменить информацию о выбранном ПО.

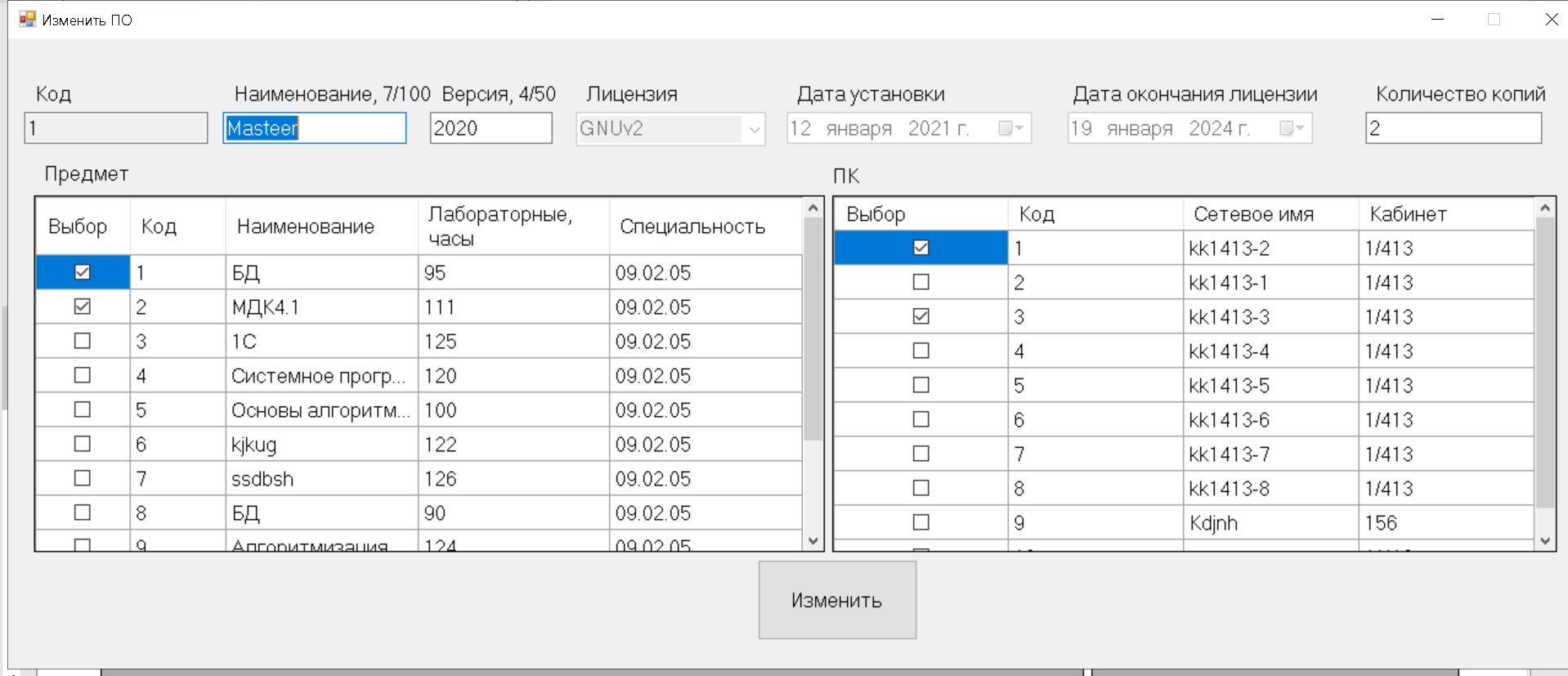


Рисунок А.14

Кнопка «Изменить предмет» открывает новое окно, где можно изменить информацию о выбранном предмете.

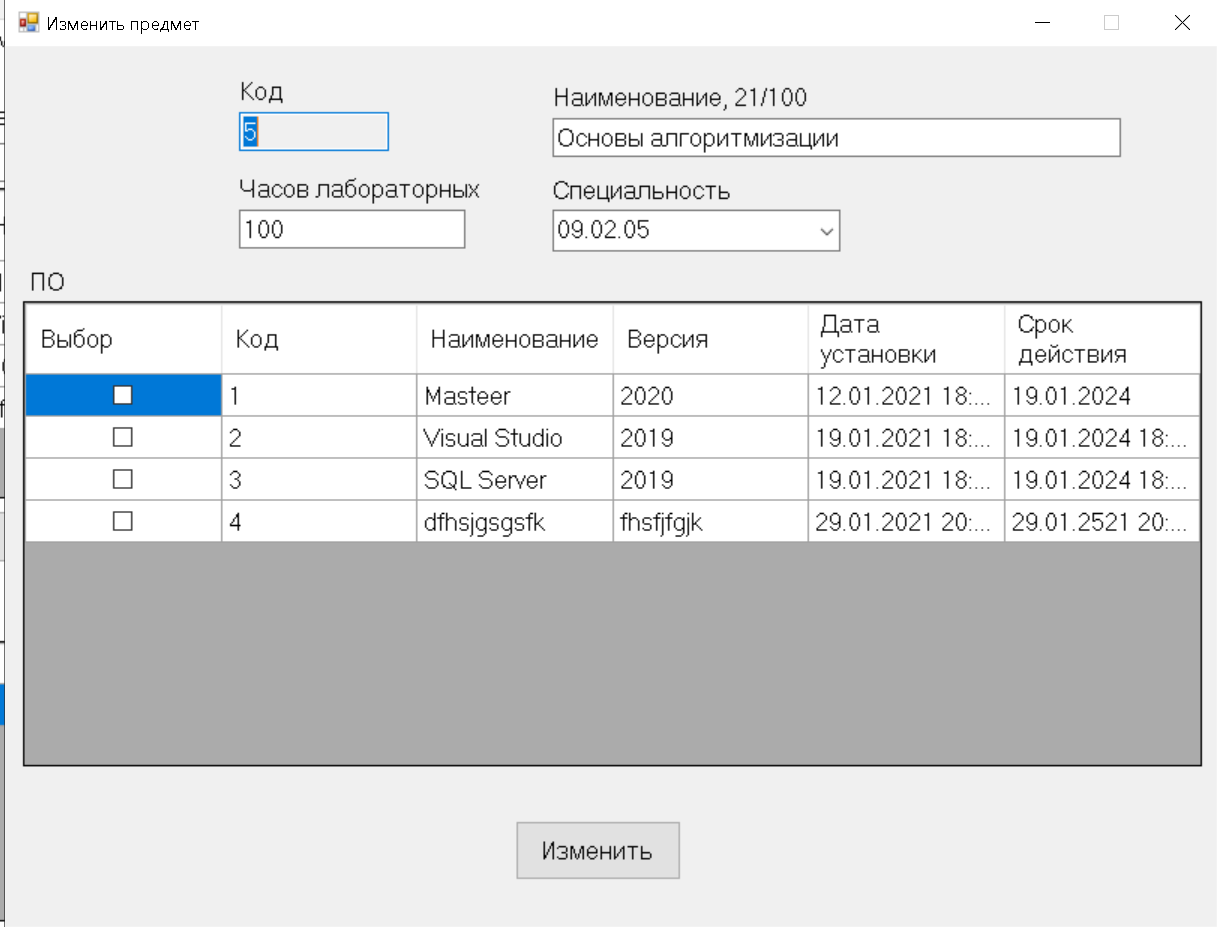


Рисунок А.15

Кнопка «Изменить лицензию» открывает новое окно, где можно изменить информацию о выбранной лицензии.

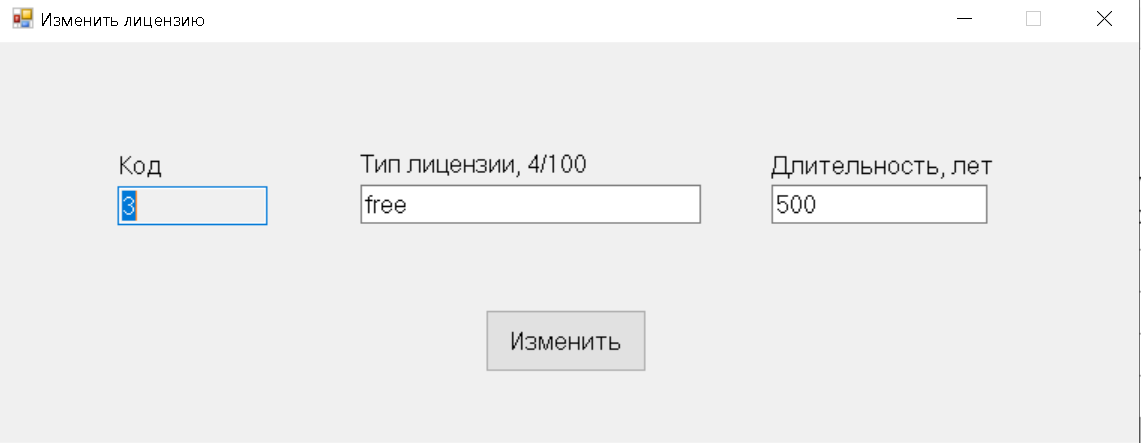


Рисунок А.16

Кнопка «Изменить ПК» открывает новое окно, где можно изменить информацию о выбранном ПК.

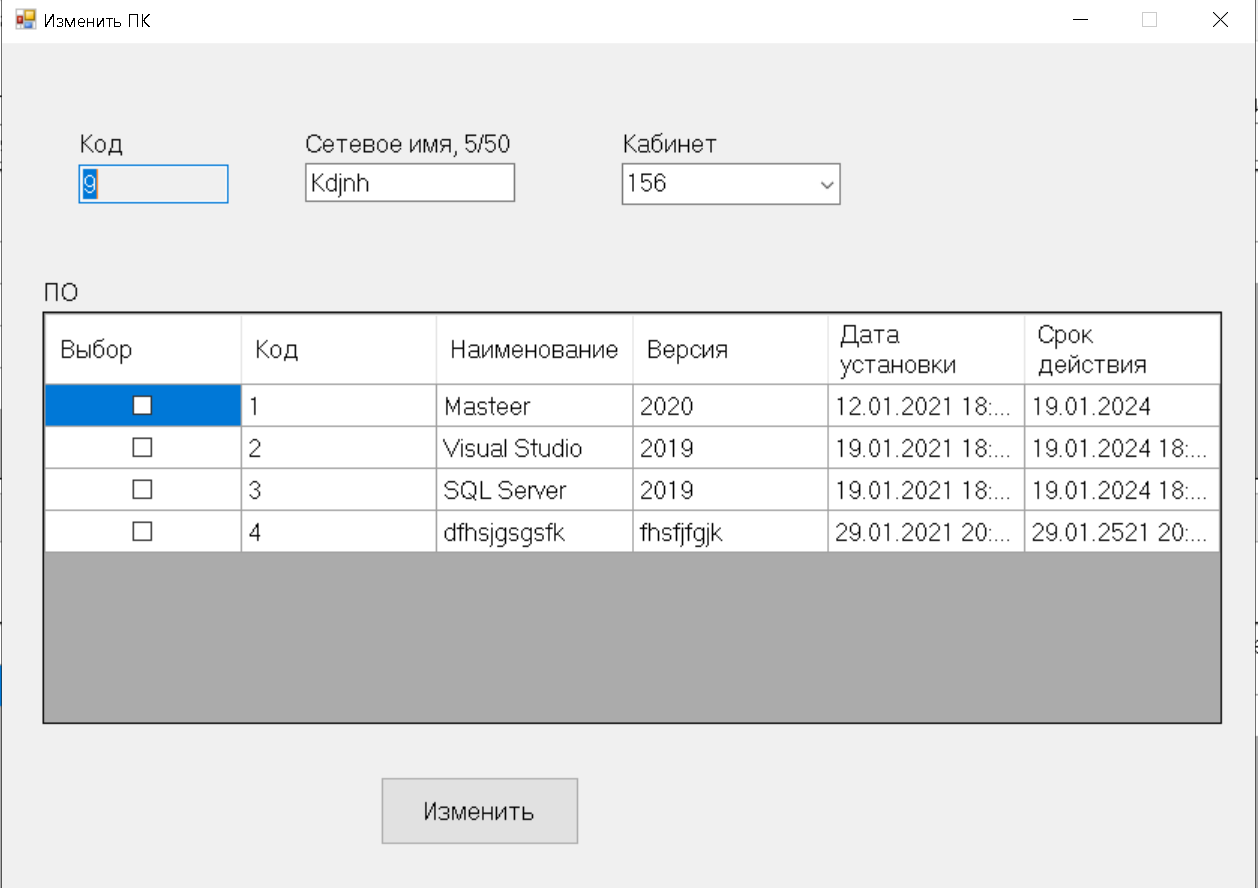


Рисунок А.17

Кнопка «Обновить ПО» открывает новое окно, где можно обновить ПО.

Открывается только если до конца действия лицензии меньше месяца.

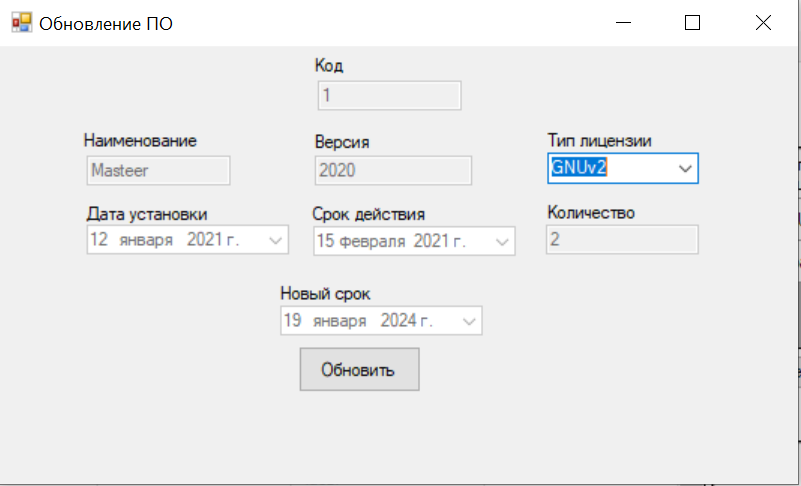


Рисунок А.18

Кнопка «Экспорт» выводит в документ Microsoft WORD информацию о выбранном ПО: наименование, версию, лицензию, предметы, ПК.

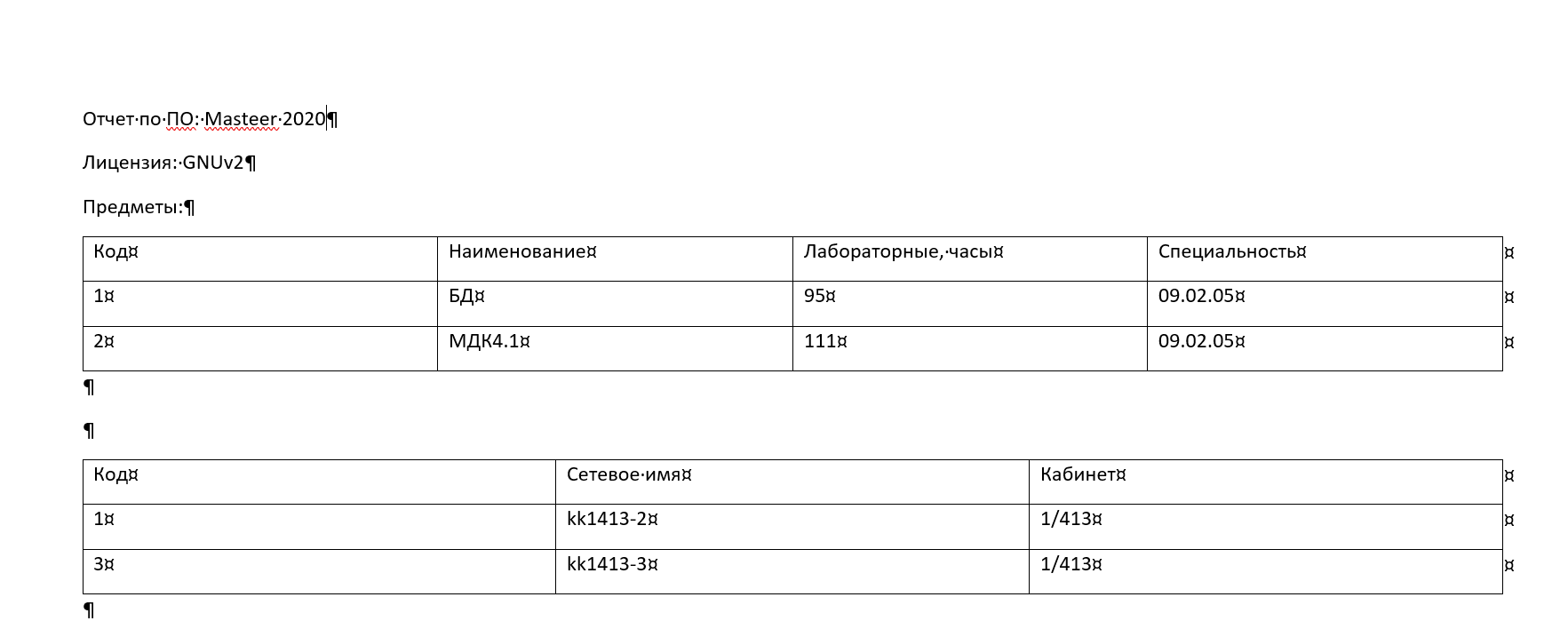


Рисунок А.19

При переходе в подсистему «Сотрудники» на экран выводится следующее.

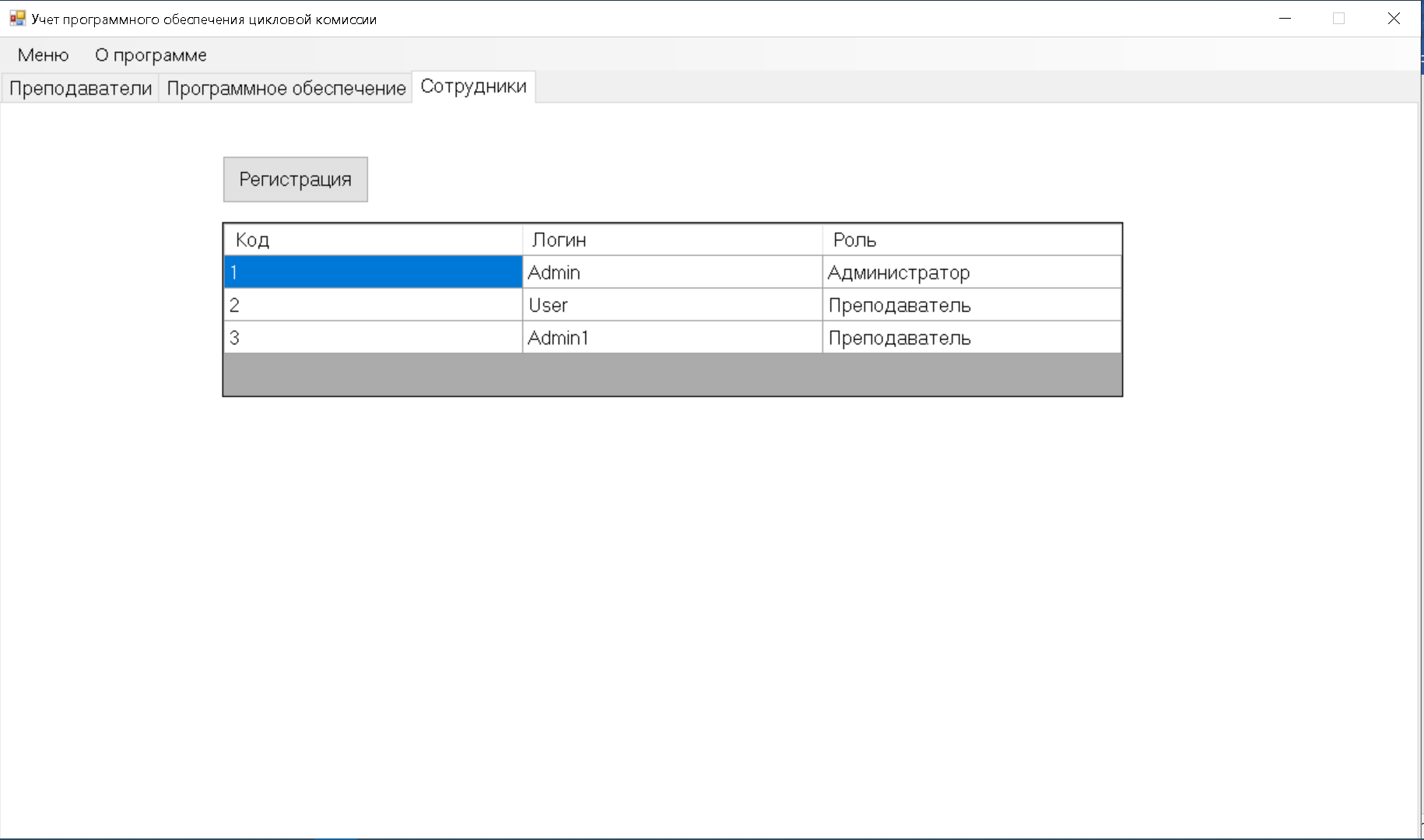


Рисунок А.20

Кнопка «Регистрация» открывает окно, где можно зарегистрировать сотрудника и назначить ему роль: администратор или преподаватель.

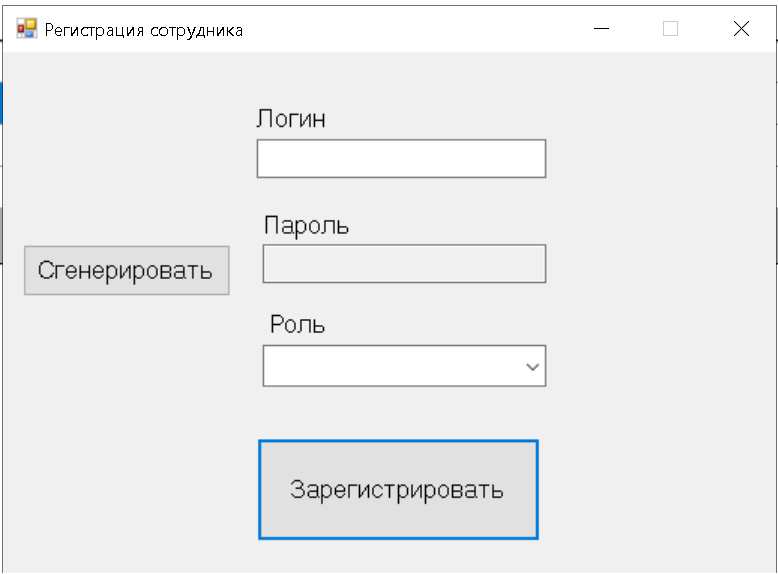


Рисунок А.21

Пункт «Предметы».

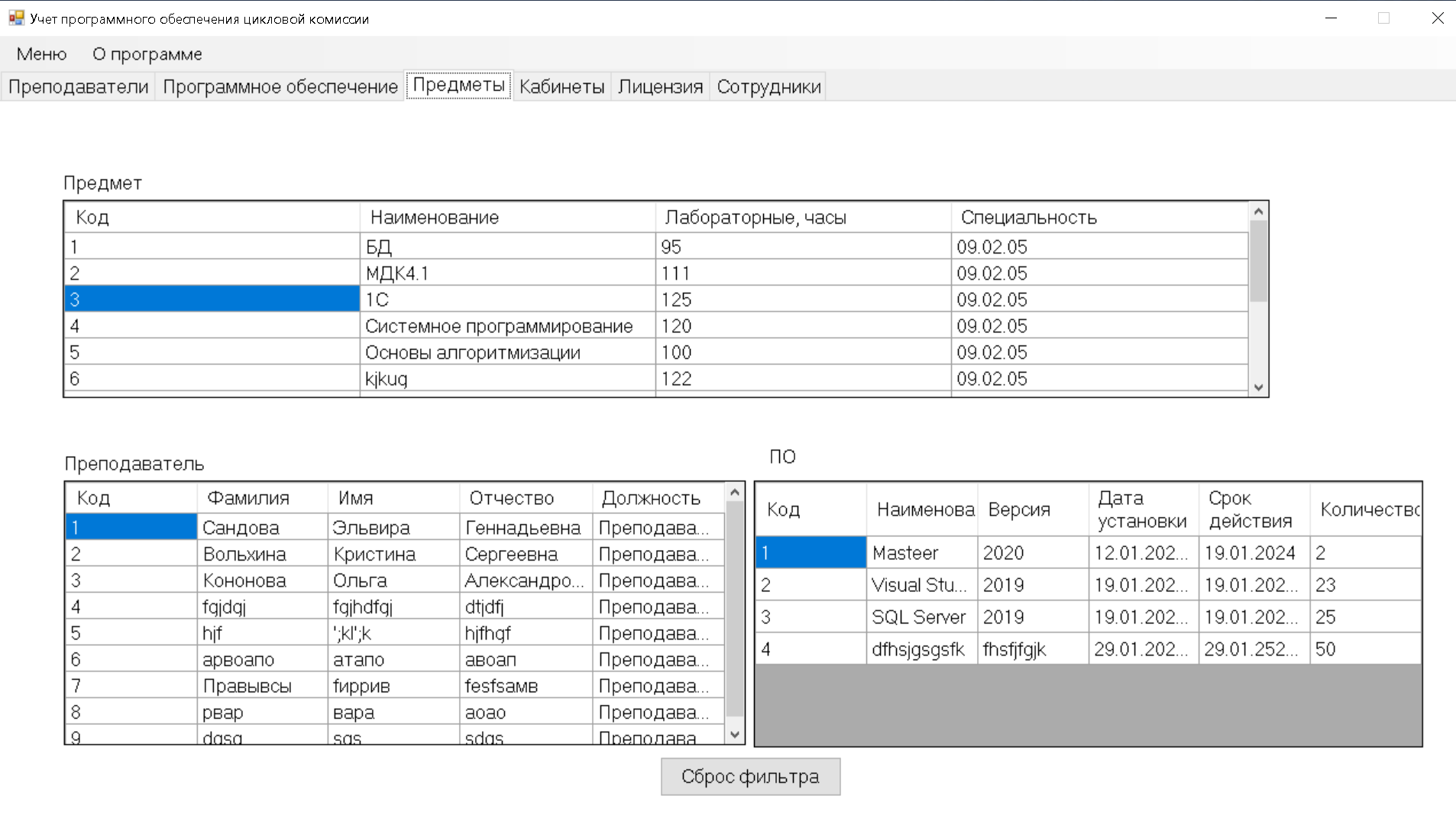


Рисунок А.22

При нажатии на ячейку таблицы «Предметы» производится поиск и фильтрация таблицы «Преподаватели» и «ПО».

Пункт «Кабинеты».

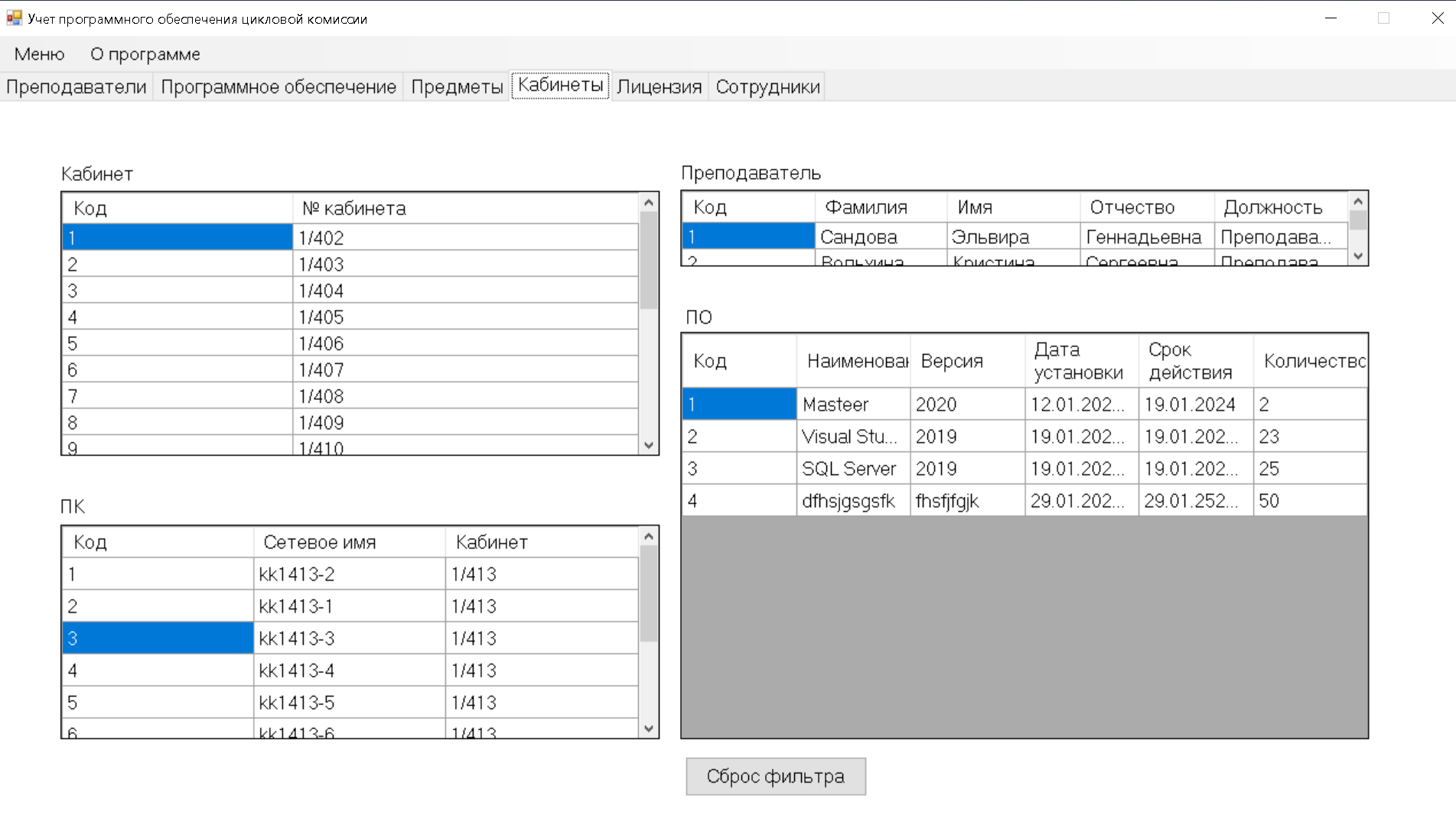


Рисунок А.23

При нажатии на ячейку таблицы «Кабинеты» производится поиск и фильтрация таблицы «Преподаватели» и «ПК».

При нажатии на ячейку таблицы «ПК» производится поиск и фильтрация данных, относящихся к выбранному ПК, в таблице «ПО».

Пункт «Лицензия».

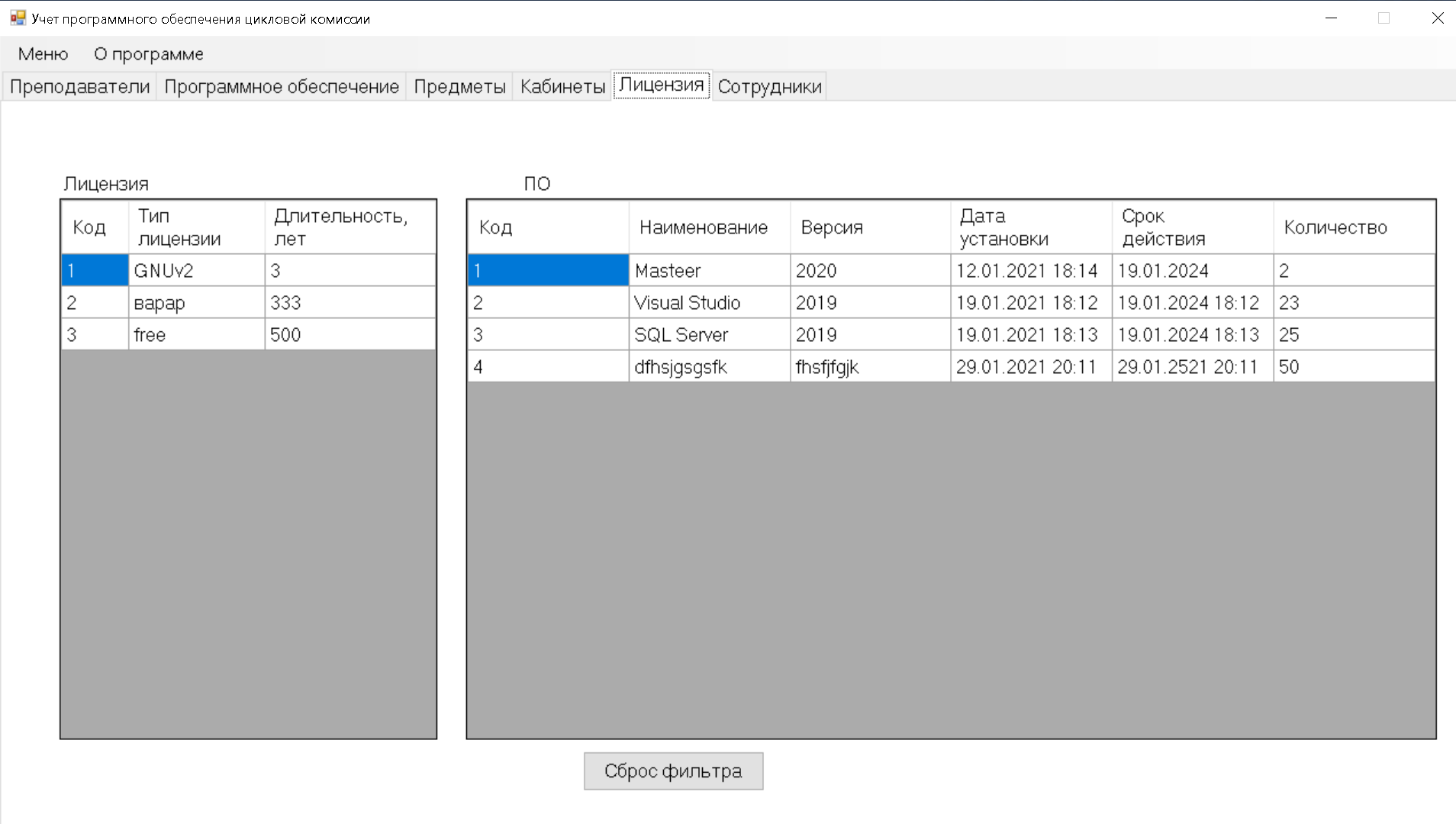


Рисунок А.24

При нажатии на ячейку таблицы «Лицензия» производится поиск и фильтрация данных, относящихся к выбранной лицензии, в таблице «ПО».

Меню состоит из следующих пунктов:

Меню

Обновить таблицы

Статистика обновления

Сменить пользователя

Выход

О программе

О программе

«Обновить таблицы» производит обновление всех таблиц на случай, если отображаются все существующие данные из базы данных.

«Сменить пользователя» производит выход из формы и переход на экран входа. Необходимо подтвердить выход.

«Выход» закрывает приложение полностью.

«О программе» выводит сообщение с информацией о программе и авторе.

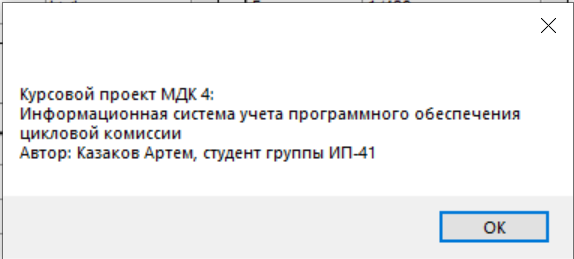


Рисунок А.25

«Статистика обновления» открывает новую форму со столбчатой диаграммой, отражающей состояние обновления программного обеспечения по годам.

Синим отображается общее число программ, которое необходимо обновить в указанном году.

Оранжевым отображается число обновленных программ за указанный год.

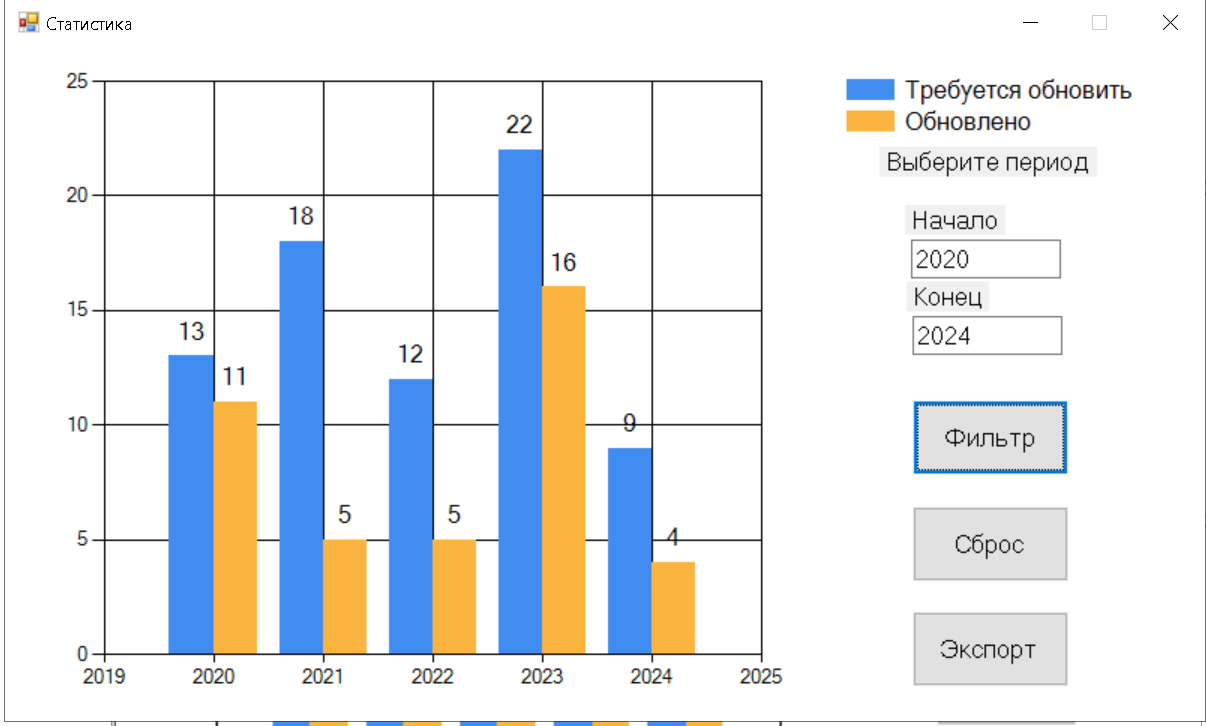


Рисунок А.26

Поля «Начало» и «Конец» - период дат в годах, в данном промежутке диаграмма фильтруется при нажатии кнопки «Фильтр».

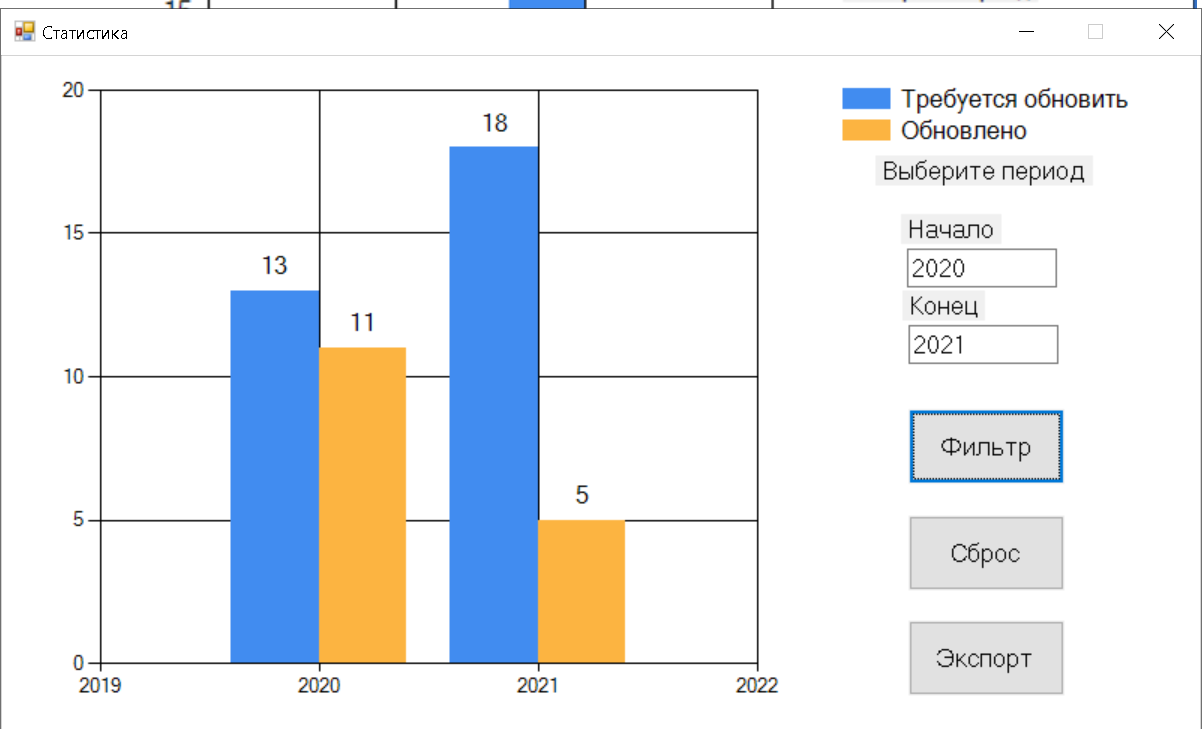


Рисунок А.27

Кнопка «Сброс» очищает фильтры и возвращает всю информацию об обновлениях.

Кнопка «Экспорт выводит отображаемую диаграмму в документ Excel.

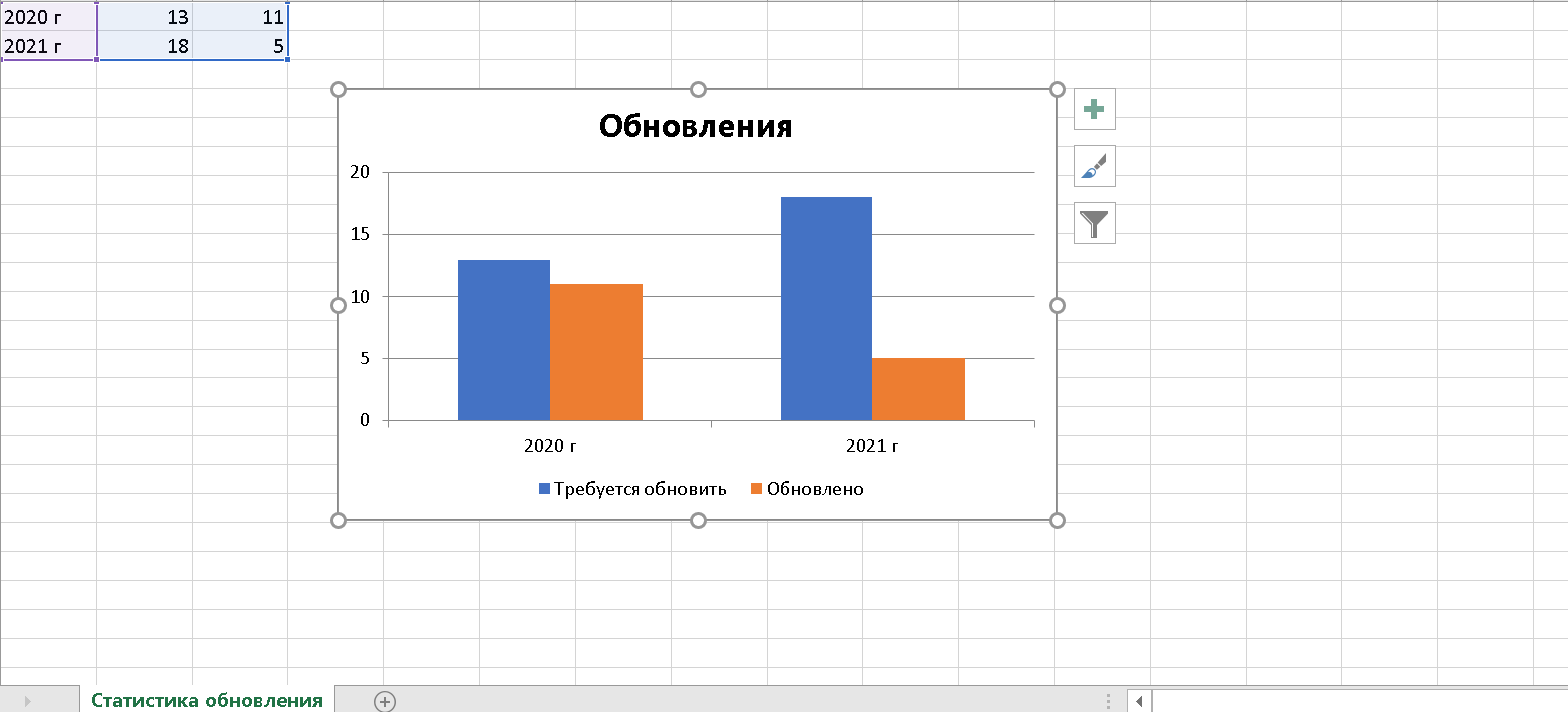


Рисунок А.28

**СООБЩЕНИЯ ОПЕРАТОРУ**

При присутствии пустых полей выводится сообщение: «Введите данные для ввода», «Введите обязательные поля», «Заполните поля»

При неправильном вводе пароля выводится сообщение: «Пароль не верен».

Если таблица «Предметы» станет пустой, то выводится сообщение: «Данный преподаватель не ведет ни один предмет», «Данная программа нигде не используется».

Если таблица «Кабинеты» станет пустой, то выводится сообщение: «У выбранного преподавателя нет кабинетов, за которые он несет ответственность».

Если таблица «Преподаватели» станет пустой, то выводится сообщение: «Данный предмет никто не преподает», «За выбранный кабинет никто не отвечает».

При вводе отсутствующей в таблице фамилии выводится сообщение: «Преподаватели, содержащие в фамилии …, не найдены».

При вводе отсутствующего в таблице кабинета выводится сообщение: «Кабинеты, содержащие …, не найдены», «ПК, содержащие …, не найдены».

При вводе отсутствующих в таблице предметов выводятся соответствующие сообщения: «Предметы, содержащие …, не найдены», «Предметы, содержащие …, и в специальности, содержащей …, не найдены».

При вводе отсутствующего в таблице ПО выводится сообщение: «Программы, содержащие …, не найдены».

При вводе отсутствующей в таблице лицензии выводится сообщение: «Лицензии, содержащие …, не найдены».

При вводе в поля неверных данных, таких как вместо цифр буквы, выводится сообщение: «Введены неверные данные».

Если введенные данные совпадают с уже существующей в базе, то выводится сообщение: «Запись уже существует».

При успешном занесении данных выводится сообщение: «Запись добавлена».

Если при внесении преподавателя не выбран предмет, то выводится сообщение: «Введите предметы, которые ведет преподаватель».

При успешном изменении записи выводится сообщение: «Запись изменена».

Если таблица «ПК» станет пустой, то выводится сообщение: «Данная программа нигде не установлена».

Если таблица «Лицензия» станет пустой, то выводится сообщение: «У данной программы отсутствует лицензия».

Если таблица «ПО» станет пустой, то выводится сообщение: «У данного предмета нет установленных программ», «На данном ПК нет установленных программ», «Нет программ с данной лицензией».

При успешном обновлении выводится сообщение: «ПО обновлено».

Если при внесении ПО не выбран предмет и ПК, то выводится сообщение: «Введите предмет и ПК».

При наличии пустых полей выводится сообщение: «Выберите лицензию».

Для создания пароля необходимо нажать кнопку «Сгенерировать», после этого сгенерируется новый шестнадцатизначный пароль, состоящий из строчных и прописных латинских букв и цифр.

При вводе уже существующего пользователя выводится сообщение: «Пользователь уже зарегистрирован».

При успешной регистрации выведется сообщение: «Пользователь зарегистрирован».

Если граничные даты не введены, то выводится сообщение: «Введите границы поиска».

Если введены неверно, то выводится сообщение «Введены неверные года».

Если Диаграмма становится пустой, то выводится сообщение: «В этих границах информация отсутствует».

ПРИЛОЖЕНИЕ Б

Код триггеров таблиц базы данных

CREATE trigger [dbo].[ins\_ins] on [dbo].[Install] instead of insert

as

declare @prog int, @pc int

select @prog = prog, @pc = pc from inserted ins

if not exists(select \* from Install where prog = @prog and pc = @pc)

insert into Install values (@prog,@pc)

CREATE trigger [dbo].[ins\_kab] on [dbo].[Kabs] instead of insert

as

declare @knum int, @knam nvarchar(25), @teach int

if exists(select \* from Kabs where knum > 0)

select @knum = Max(knum) from Kabs

else

select @knum = 0

select @knam = knam, @teach = teach from inserted ins

if not exists(select \* from Kabs where knam = @knam and teach = @teach)

insert into Kabs values (@knum + 1,@knam,@teach)

CREATE trigger [dbo].[upd\_kab] on [dbo].[Kabs] instead of update

as

declare @knum int, @knam nvarchar(25), @teach int

select @knum = knum, @knam = knam, @teach = teach from inserted

if not exists(select \* from Kabs where knam = @knam and teach = @teach and not knum = @knum)

Update Kabs set knam = @knam, teach = @teach where knum = @knum

CREATE trigger [dbo].[ins\_lic] on [dbo].[License] instead of insert

as

declare @lnum int, @lnam nvarchar(100), @dlit int

if exists(select \* from License where lnum > 0)

select @lnum = Max(lnum) from License

else

select @lnum = 0

select @lnam = lnam, @dlit = dlit from inserted ins

if not exists(select \* from License where lnam = @lnam and dlit = @dlit)

insert into License values (@lnum + 1,@lnam,@dlit)

CREATE trigger [dbo].[upd\_lic] on [dbo].[License] instead of update

as

declare @lnum int, @lnam nvarchar(100), @dlit int

select @lnum = lnum, @lnam = lnam, @dlit = dlit from inserted ins

if not exists(select \* from License where lnam = @lnam and dlit = @dlit and not lnum = @lnum)

Update License set lnam = @lnam, dlit = @dlit where lnum = @lnum

CREATE trigger [dbo].[ins\_pc] on [dbo].[PC] instead of insert

as

declare @pcnum int, @pcnet nvarchar(50), @kab int

if exists(select \* from PC where pcnum > 0)

select @pcnum = Max(pcnum) from PC

else

select @pcnum = 0

select @pcnet = pcnet, @kab = kab from inserted ins

if not exists(select \* from pc where pcnet = @pcnet and kab = @kab)

insert into PC values (@pcnum + 1,@pcnet,@kab)

CREATE trigger [dbo].[upd\_pc] on [dbo].[PC] instead of Update

as

declare @pcnum int, @pcnet nvarchar(50), @kab int

select @pcnum = pcnum, @pcnet = pcnet, @kab = kab from inserted ins

if not exists(select \* from pc where pcnet = @pcnet and kab = @kab and not PCnum = @pcnum)

Update PC set pcnet = @pcnet, kab = @kab where pcnum = @pcnum

CREATE trigger [dbo].[ins\_pred] on [dbo].[Predmet] instead of insert

as

declare @pnum int, @pnam nvarchar(100), @hours int, @spec nvarchar(40)

if exists(select \* from Predmet where pnum > 0)

select @pnum = Max(pnum) from Predmet

else

select @pnum = 0

select @pnam = pnam, @hours = lab\_hours, @spec = spec from inserted ins

if not exists(select \* from Predmet where pnam = @pnam and lab\_hours = @hours and spec = @spec)

insert into Predmet values (@pnum + 1,@pnam,@hours,@spec)

CREATE trigger [dbo].[upd\_pred] on [dbo].[Predmet] instead of update

as

declare @pnum int, @pnam nvarchar(100), @hours int, @spec nvarchar(40)

select @pnum = pnum, @pnam = pnam, @hours = lab\_hours, @spec = spec from inserted ins

if not exists(select \* from Predmet where pnam = @pnam and lab\_hours = @hours and spec = @spec and not pnum = @pnum)

Update Predmet set pnam = @pnam,lab\_hours = @hours,spec = @spec where pnum = @pnum

CREATE trigger [dbo].[ins\_prepod] on [dbo].[Prepodav] instead of insert

as

declare @tnum int, @pnum int

select @tnum = tnum, @pnum = pnum from inserted ins

if not exists(select \* from Prepodav where tnum = @tnum and pnum = @pnum)

insert into prepodav values (@tnum,@pnum)

CREATE trigger [dbo].[ins\_progpred] on [dbo].[ProgPred] instead of insert

as

declare @prog int, @predm int

select @prog = prog, @predm = predm from inserted ins

if not exists(select \* from ProgPred where prog = @prog and predm = @predm)

insert into ProgPred values (@prog,@predm)

CREATE trigger [dbo].[ins\_prog] on [dbo].[Programs] instead of insert

as

declare @prnum int, @prnam nvarchar(100), @vers nvarchar(50), @lice int, @inst datetime, @expr datetime, @value int

if exists(select \* from Programs where prnum > 0)

select @prnum = Max(prnum) from Programs

else

select @prnum = 0

select @prnam = prnam, @vers = version, @lice = license, @inst = inst\_dat, @expr = expr\_dat, @value = value from inserted ins

if not exists(select \* from programs where prnam = @prnam and version = @vers and license = @lice and inst\_dat = @inst and expr\_dat = @expr and value = @value)

Begin

insert into Programs values (@prnum + 1,@prnam,@vers,@lice,@inst,@expr,@value)

if exists (select \* from ProgUpd where years = datepart(yy,@expr))

Update ProgUpd set All\_upd= All\_Upd+1

else

Begin

insert into ProgUpd (Years,All\_Upd,Upd) values (DATEPART(yy,@expr),1,0)

end

end

CREATE trigger [dbo].[upd\_prog] on [dbo].[Programs] instead of update

as

declare @prnum int, @prnam nvarchar(100), @vers nvarchar(50), @lice int, @inst datetime, @expr datetime, @value int

select @prnum = prnum, @prnam = prnam, @vers = version, @lice = license, @inst = inst\_dat, @expr = expr\_dat, @value = value from inserted ins

if not exists(select \* from programs where prnam = @prnam and version = @vers and license = @lice and inst\_dat = @inst and expr\_dat = @expr and value = @value and not prnum = @prnum)

else

update Programs set prnam = @prnam,version = @vers,license = @lice,inst\_dat = @inst,expr\_dat = @expr,value = @value where prnum = @prnum

CREATE trigger [dbo].[ins\_sotr] on [dbo].[Sotr] instead of insert

as

declare @snum int, @snam nvarchar(50), @pass nvarchar(50), @role nvarchar(50)

if exists(select \* from Sotr where snum > 0)

select @snum = Max(snum) from Sotr

else

select @snum = 0

select @snam = snam, @pass = pass, @role = role from inserted ins

if not exists(select \* from sotr where snam = @snam)

insert into Sotr values (@snum + 1,@snam,@pass, @role)

CREATE trigger [dbo].[upd\_sotr] on [dbo].[Sotr] instead of update

as

declare @snum int, @snam nvarchar(50), @pass nvarchar(50), @role nvarchar(50)

select @snum = snum, @snam = snam, @pass = pass, @role = role from inserted ins

if not exists(select \* from sotr where pass = @pass)

update Sotr set snam = @snam, pass = @pass, role = @role where snum = @snum

CREATE trigger [dbo].[ins\_teach] on [dbo].[Teachers] instead of insert

as

declare @tnum int, @tfam nvarchar(50), @tnam nvarchar(50), @totch nvarchar(50), @dolg nvarchar(50)

if exists(select \* from Teachers where tnum > 0)

select @tnum = Max(tnum) from Teachers

else

select @tnum = 0

select @tfam = tfam, @tnam = tnam, @totch = totch, @dolg = dolg from inserted ins

if not exists(select \* from teachers where tfam = @tfam and tnam = @tnam and totch = @totch and dolg = @dolg)

insert into Teachers values (@tnum + 1,@tfam,@tnam,@totch,@dolg)

CREATE trigger [dbo].[upd\_teach] on [dbo].[Teachers] instead of update

as

declare @tnum int, @tfam nvarchar(50), @tnam nvarchar(50), @totch nvarchar(50), @dolg nvarchar(50)

select @tnum = tnum, @tfam = tfam, @tnam = tnam, @totch = totch, @dolg = dolg from inserted ins

if not exists(select \* from teachers where tfam = @tfam and tnam = @tnam and totch = @totch and dolg = @dolg and not tnum = @tnum)

update Teachers set tfam = @tfam,tnam = @tnam,totch = @totch,dolg = @dolg where tnum = @tnum