## ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "САНКТ-ПЕТЕРБУРГСКИЙ

## ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М. А. БОНЧ-БРУЕВИЧА"

## Факультет инфокоммуникационных сетей и систем Кафедра программной инженерии и вычислительной техники

**КУРСОВАЯ РАБОТА**

**по дисциплине «Объектно-ориентированное программирование» ВАРИАНТ 5**

## Выполнил: студент 2 курса группы ИКПИ-14 Дунаев В.Е.

## Санкт-Петербург 2022

**Оглавление**

1. [Постановка задачи 3](#_bookmark0)
2. [Выбор и обоснование типа разрабатываемого контейнера 4](#_bookmark1)
3. [Разработка структуры программы 5](#_bookmark2)
4. [Разработка собственных классов 6](#_bookmark3)
5. [Разработка интерфейса 9](#_bookmark4)
6. [Инструкция пользователю 11](#_bookmark5)
7. [Код программы 19](#_bookmark6)
8. [Заключение 44](#_bookmark7)
9. [Список литературы 44](#_bookmark8)

# Постановка задачи

Создать базу данных, работающую на основе контейнера, который следует разработать в процессе выполнения курсовой работы. В качестве прототипа контейнера можно использовать контейнеры vector или list из библиотеки STL. Для работы с контейнером следует предусмотреть итератор. Остальные требования к контейнеру определяются студентом самостоятельно.

Разрабатываемая база данных должна быть создана c использованием C++ и библиотеки Qt4. Среда программировании Qt Creater.

**В качестве элементов базы данных по варианту 5 должны использоваться характеристики радиодеталей(тип, номинал и т. д.).**

**База данных должна обеспечивать выполнение следующих операций:**

## Создание базы данных,

## Объединение баз данных,

## Добавления и удаления записей,

## Редактирования записей,

## Просмотра базы данных,

## Поиск данных,

## Запись на диск,

## Чтение с диска.

**Интерфейсная часть программы должна содержать следующие компоненты:**

## Окно «О программе»,

## Главное меню,

## Всплывающее меню,

## Строку состояния,

## Панель инструментов,

## Подсказки по командам меню и панели инструментов.

# Выбор и обоснование типа разрабатываемого контейнера

**В курсовой работе необходимо использовать контейнер для хранения объектов.**

**В качестве прототипа разрабатываемого контейнера выбран vector** (динамический массив), так как он наиболее легкий в реализации, в понимании, а также его можно быстро написать.

**Разрабатываемый контейнер должен обладать следующими функциями**:

## Создание пустого контейнера;

## Создание контейнера размера n;

## Конструкторы и операторы копирования и перемещения;

## Получение элемента по индексу;

## Вставка элемента в произвольную позицию;

## Очистка контейнера;

## Удаление элемента по индексу;

## Наличие класса итератора и итераторов begin и end.

**Также в классе, характеризующем радиодетали,** все поля сделаны публичными для упрощения и сокращения времени разработки(все писалось за одну ночь...)

1. **Разработка структуры программы Программа состоит из интерфейса и кода.**

**Интерфейс:**

* **Основное окно программы**.

## Главное меню MenuBar.

## Панель инструментов ToolBar.

## Таблица TableWidget.

## Строка поиска по таблице LineEdit.

## Статусная строка StatusBar.

* **Диалог добавления / редактирования элемента**.

## 3 метки Label: тип, название, аргументы и комментарий к функции.

## 3 текстовые поля LineEdit.

**Код:**

## Основной класс формы QT — окно главной программы.

## Класс формы QT — диалог добавления / редактирования записи.

## Шаблонный класс контейнера типа vector.

## Класс функции языка программирования.

# Разработка собственных классов

## **Radio** — класс характеризующий радиодеталь.

* **Поля класса**:

## Имя: QString type.

## Характеристики: QString specifications.

## Цена: int cost.

* **Методы класса**:

## Оператор присваивания

## (В данном классе все поля являются публичными, поэтому для них не требуются геттеры и сеттеры и большего и не надо)

## **Vector** — класс контейнера выбранного типа.

* **Поля класса**:

## Текущий размер массива: int sz.

## Резервируемый размер массива: int cap.

## Указатель на массив с элементами: T \* array.

* **Методы класса**:

## Оператор получения элемента по индексу.

## Получение размера массива: size.

## Вставка элемента в произвольную позицию: insert.

## Добавление элемента в начало: push\_front.

## Добавление элемента в конец: push\_back.

## Очистка массива: clear.

## Получение булевого значения "пустой ли массив".

## Удаление элемента произвольного индекса: erase.

## Удаление первого элемента: pop\_front.

## Удаление последнего элемента: pop\_back.

**MainWindow** — класс главного окна.

* **Поля класса**:

## Строка, содержащая текущее имя файла: QString current\_file.

## Контейнер с элементами: Vector<Radio> array.

* **Методы класса**:

## Выбор действия "Новый файл": on\_new\_bd\_triggered.

## Выбор действия "Открыть": on\_open\_bd\_triggered.

## Выбор действия "Сохранить": on\_save\_bd\_triggered.

## Выбор действия "Сохранить как": on\_save\_as\_bd\_triggered.

## Выбор действия "Объединить": on\_merge\_bd\_triggered.

## Выбор действия "Добавить запись": on\_add\_sigle\_to\_bd\_triggered.

## Выбор действия "Редактировать запись": on\_edit\_single\_in\_bd\_triggered.

## Выбор действия "Удалить запись": on\_delete\_some\_triggered.

## Выбор действия "Очистить": on\_close\_bd\_triggered.

## Выбор действия "О программе": on\_author\_triggered.

## Выбор действия "Выйти": on\_exit\_\_triggered.

## Выбор действия "Помощь" : on\_help\_and\_tips\_triggered.

## Поиск : on\_search\_in\_bd\_triggered и on\_pushButton\_clicked.

## **Pop\_up** — класс диалогового окна для добавления / редактирования записи.

* **Методы класса**:

## Установка типа радиодетали: type.

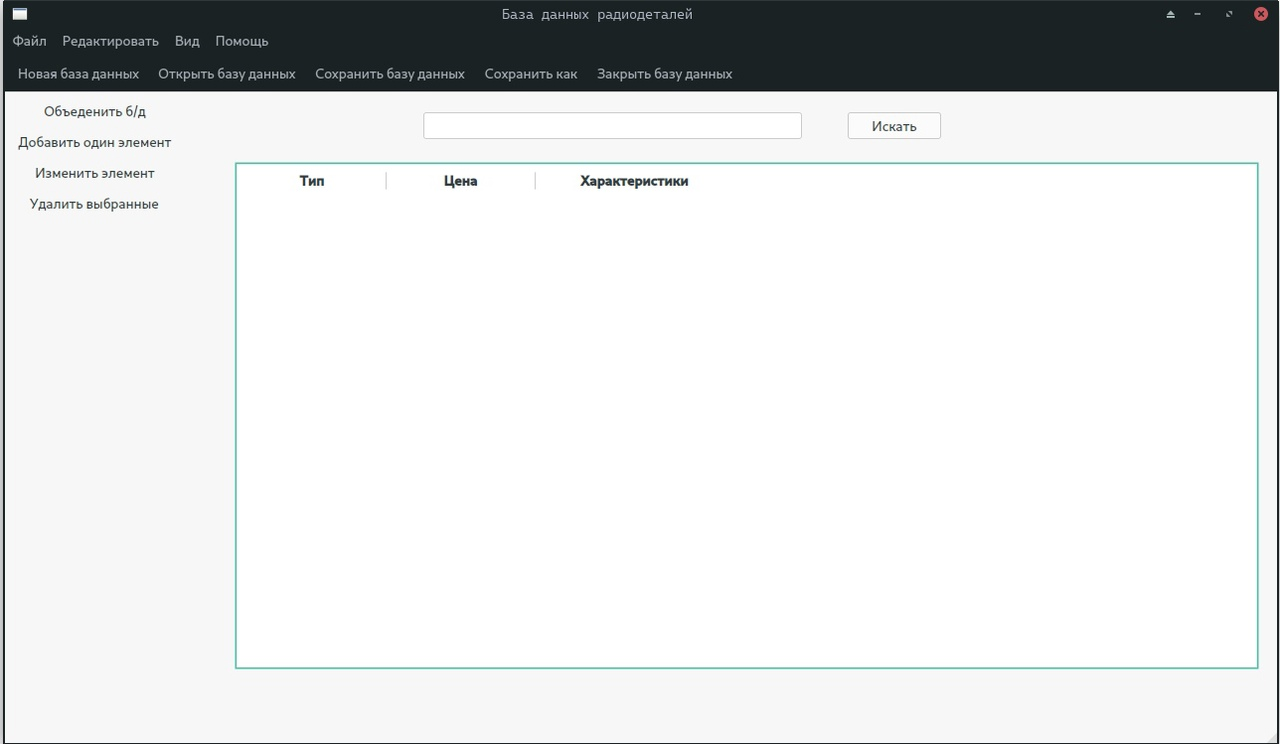
## Установка цены радиодетали: cost.

## Установка характеристик радиодетали: specs.

## (Получение данных в результате диалога происходит благодаря передачи в него ссылки на Radio).

# Разработка интерфейса

1. **Основное окно программы**



## 

## 

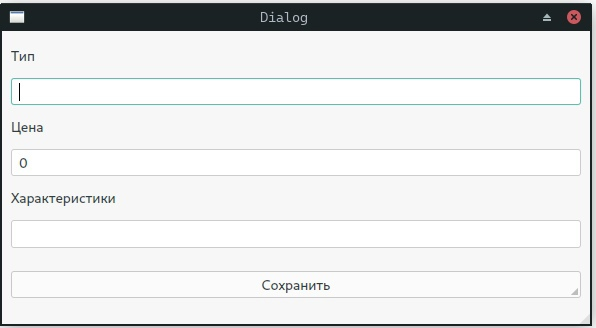
## 

## 

## Главное меню, панель инструментов, таблица, строка поиска и статусная

## строка (пустая)

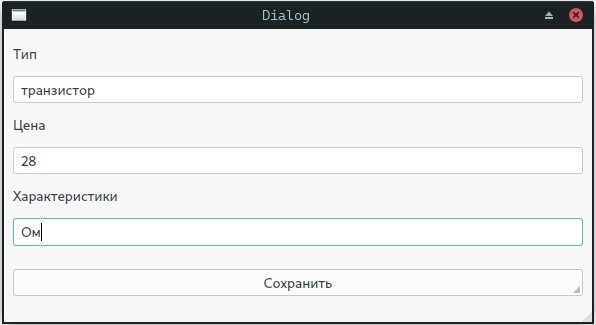
1. **Диалог добавления / редактирования записи** (размер окна не меняется).



## После нажатия OK все пробелы в текстовых строках удаляются слева и справа и нормализуются в середине до одного, а точки с запятой преобразуются в запятые. Таким образом, все элементы базы данных приводятся к одному виду.

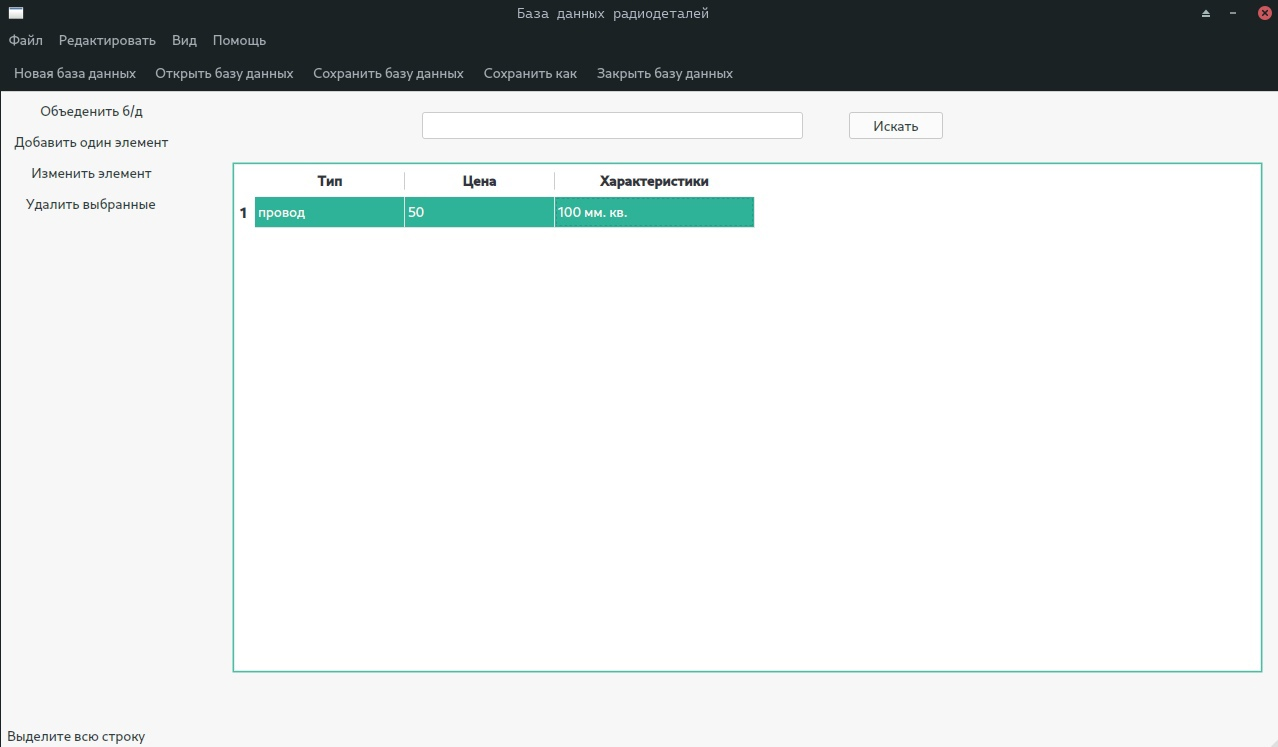
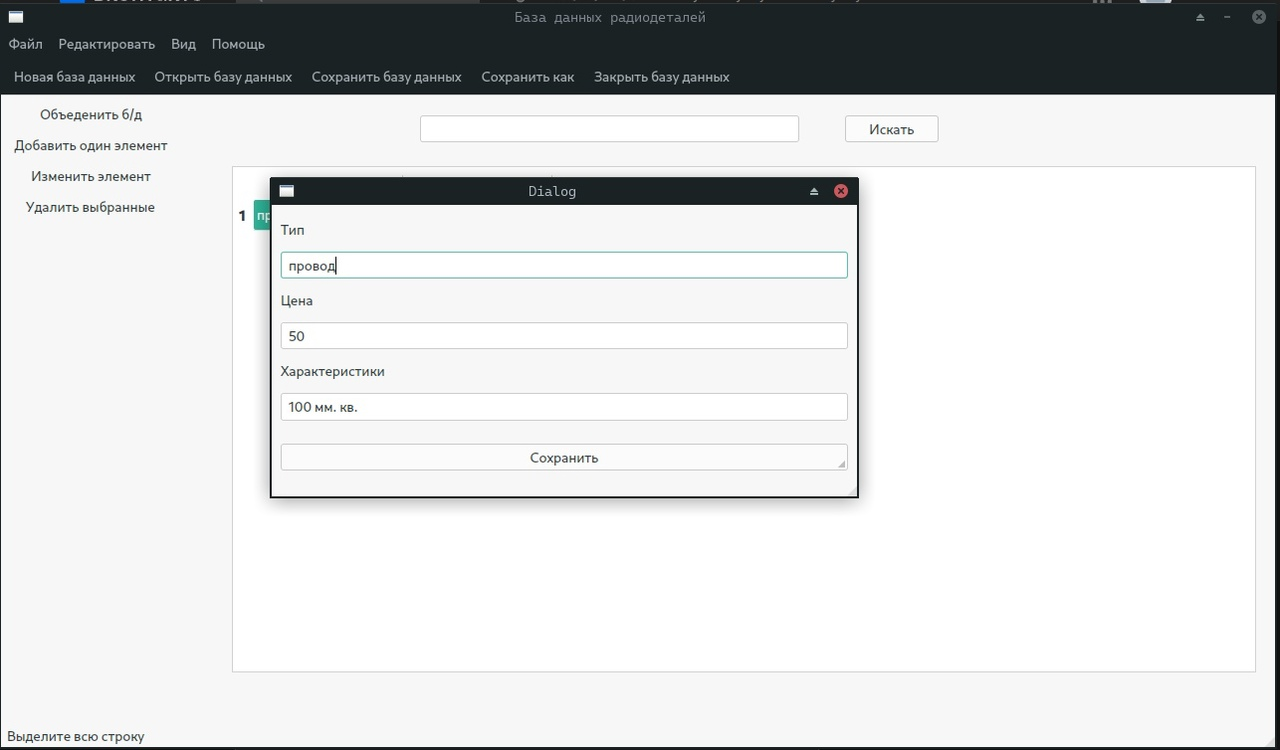
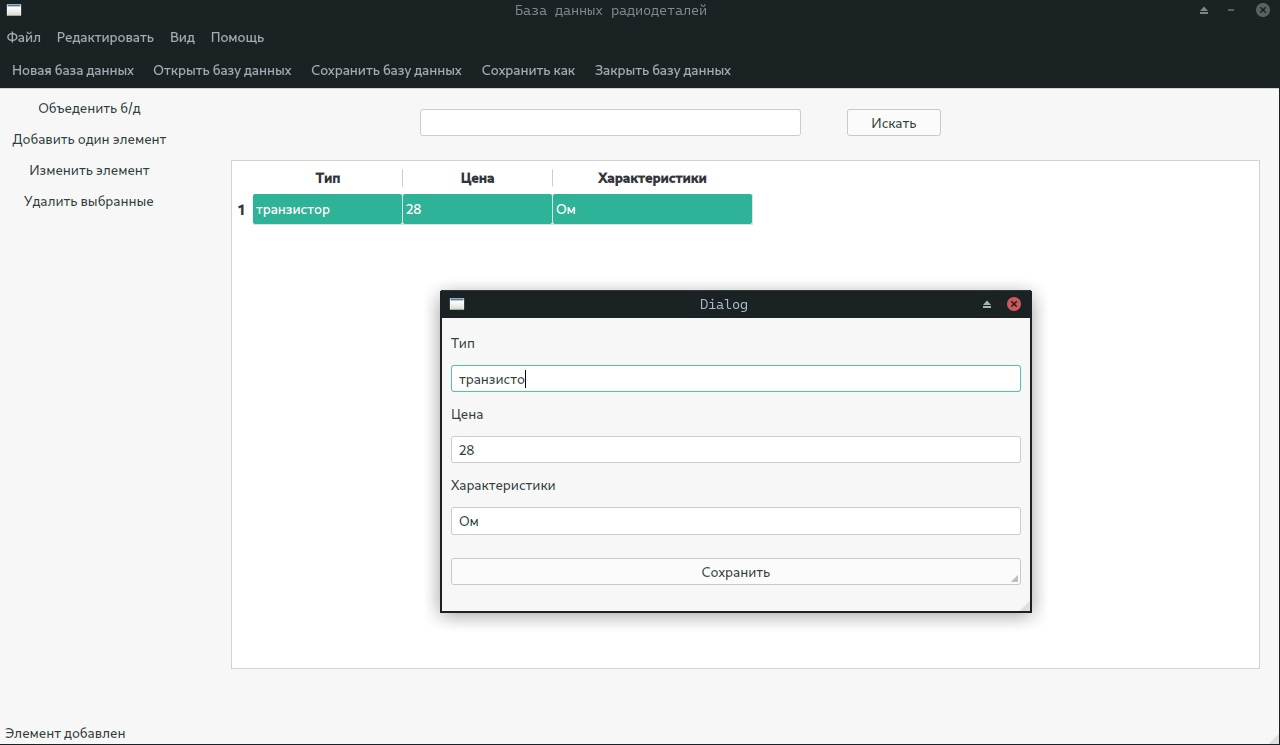
# Инструкция пользователю

1. **Добавление записи .**

****

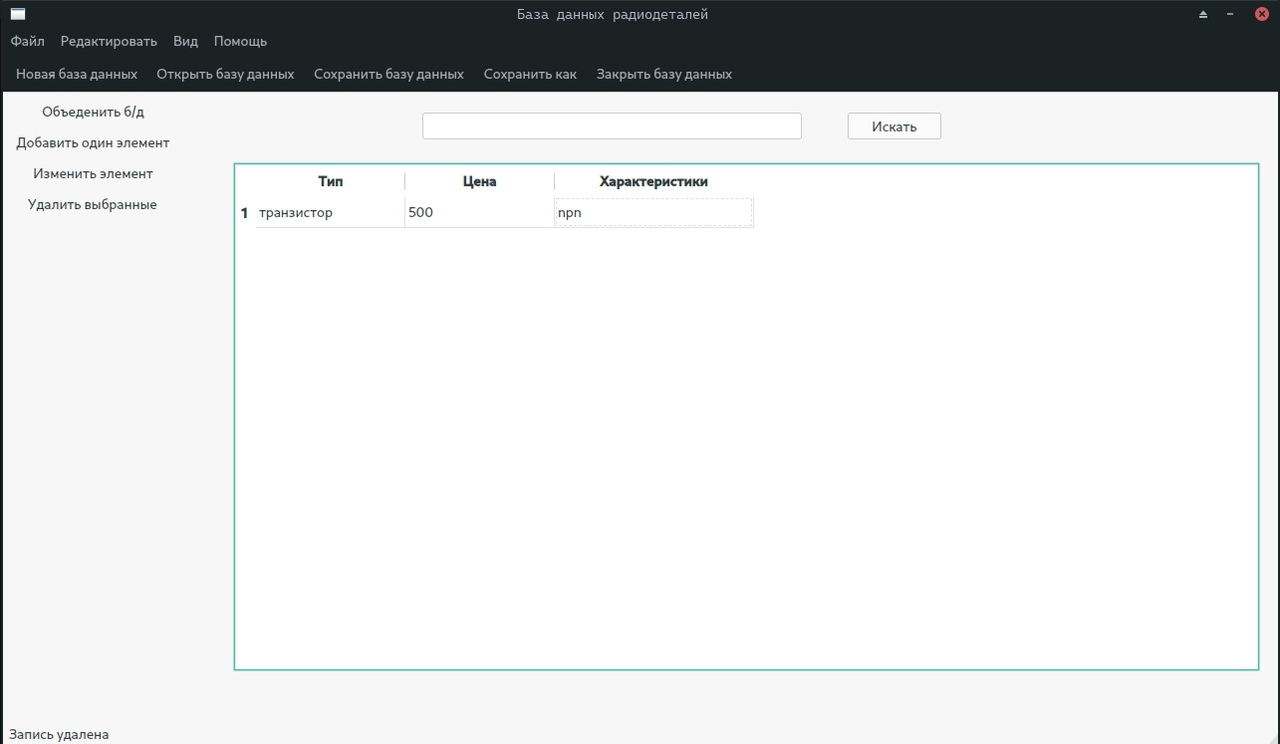
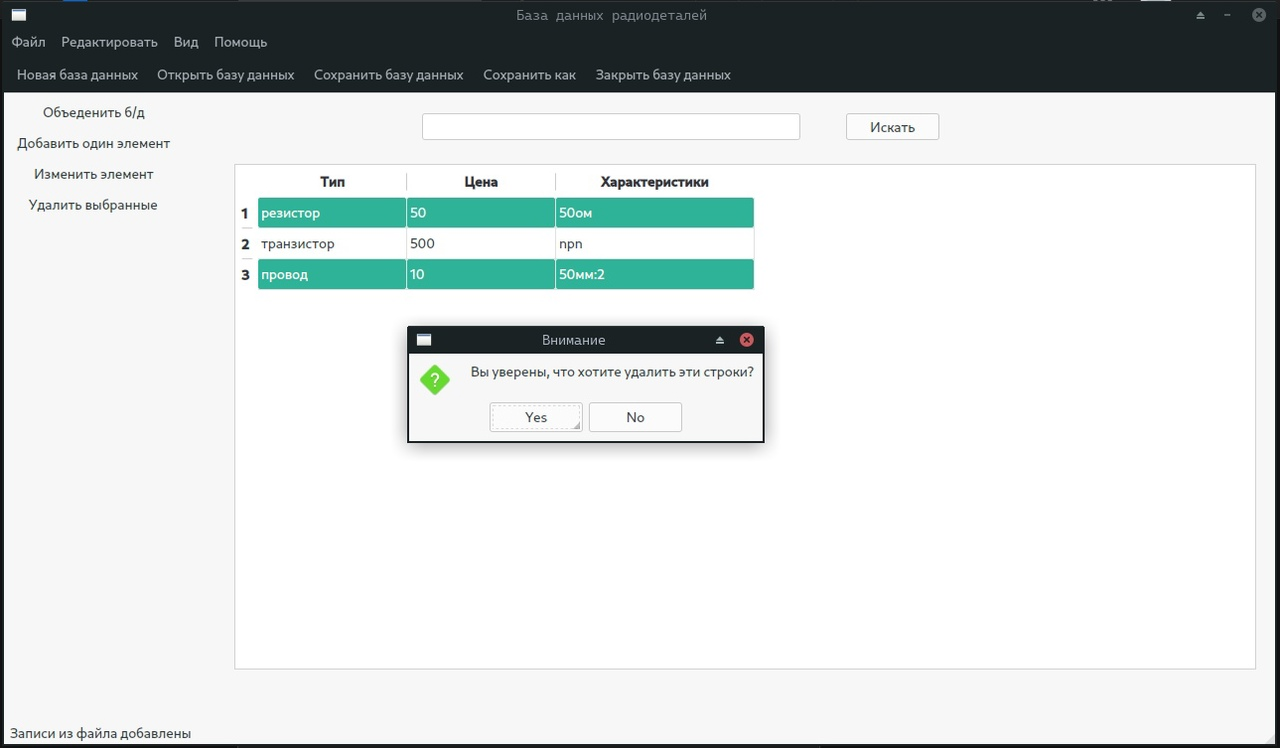
## 

## Диалог добавления записи и отображение результата в главном окне в таблице и в статусной строке

1. **Редактирование записи .**

## Диалог редактирования записи и отображение результата в главном окне в таблице и в статусной строке. Также можно дважды кликнуть по любой строке, чтобы её отредактировать.

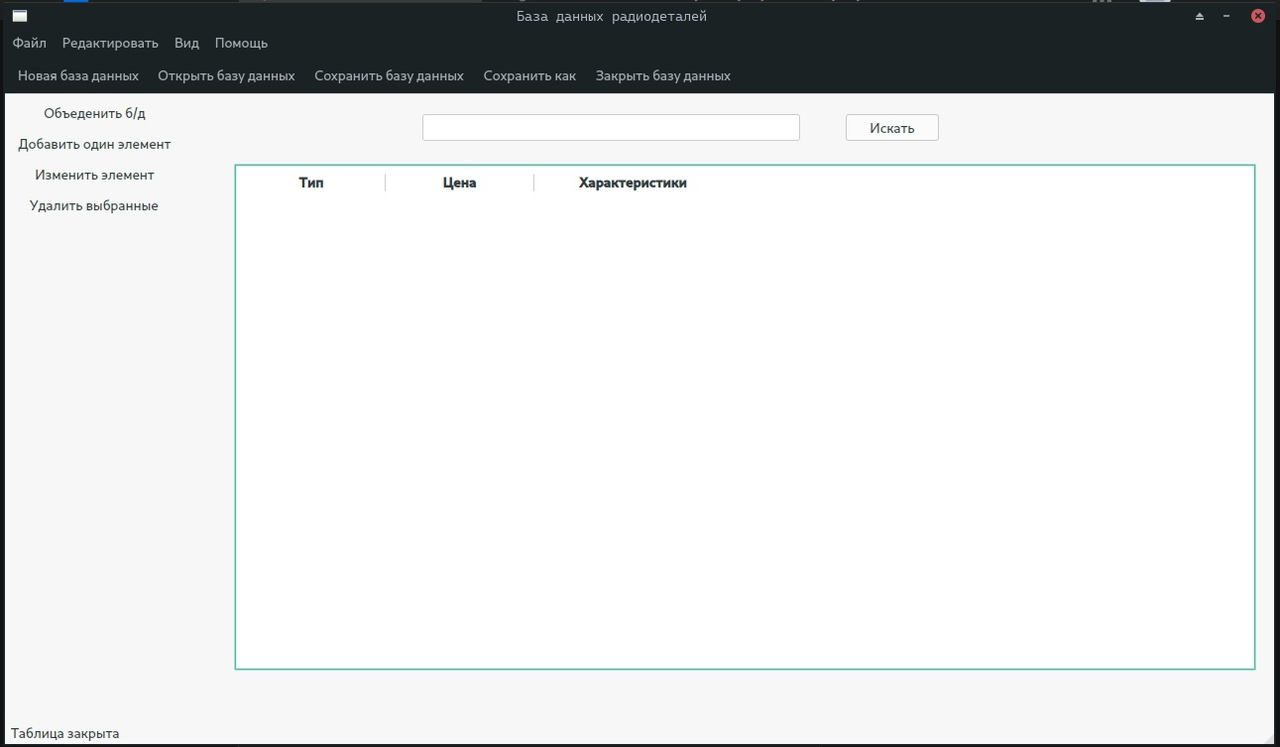
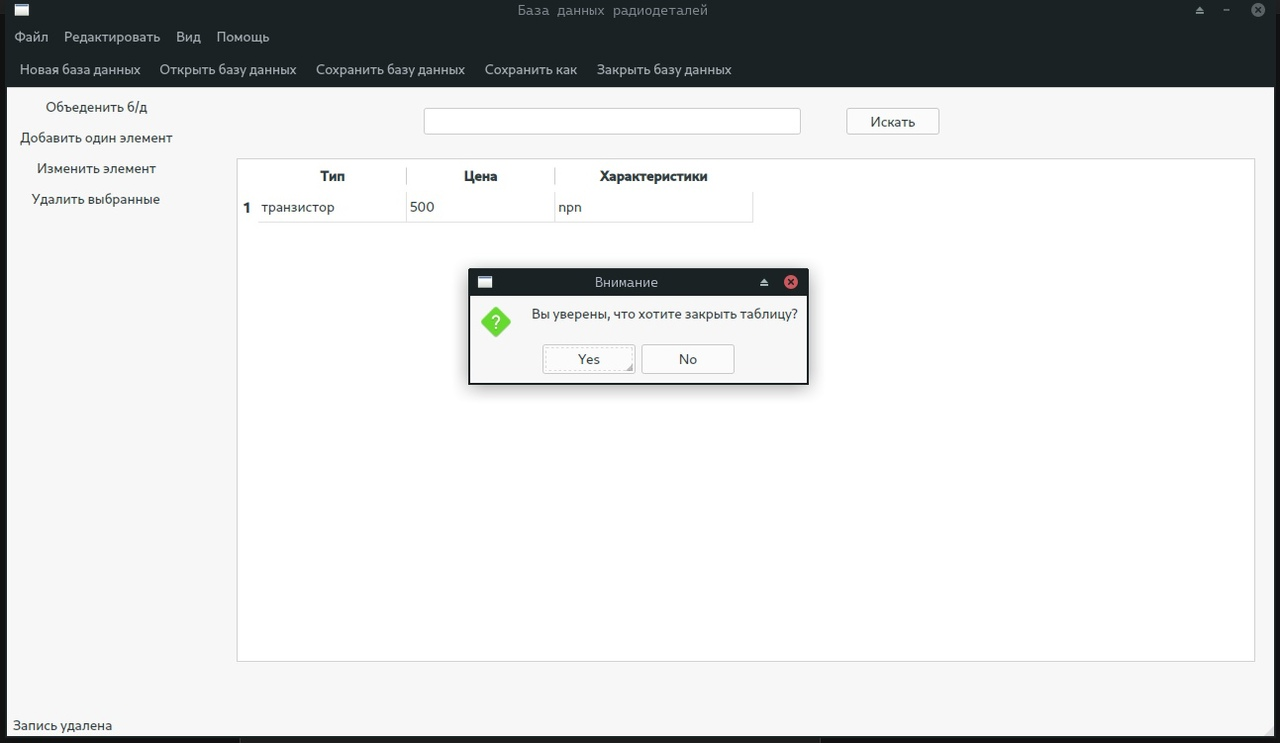
1. **Удаление записи .**

****

## Удаление записи (записей) с подтверждением.

## Для того, чтобы выделить несколько, необходимо зажать Ctrl при выборе.

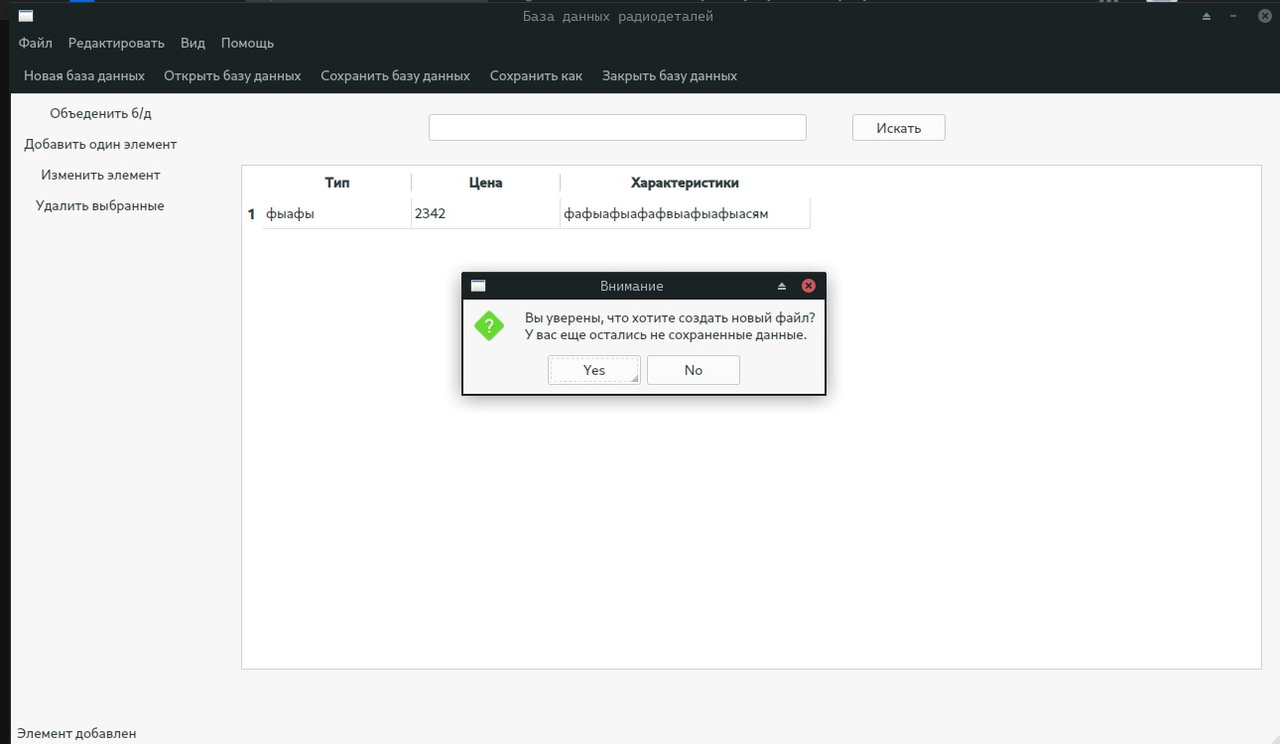
1. **Очистить записи .**

****

## Очистка записей с подтверждением.

## Подтверждение не отображается, если таблица пуста.

1. **Создание новой базы данных .**

****

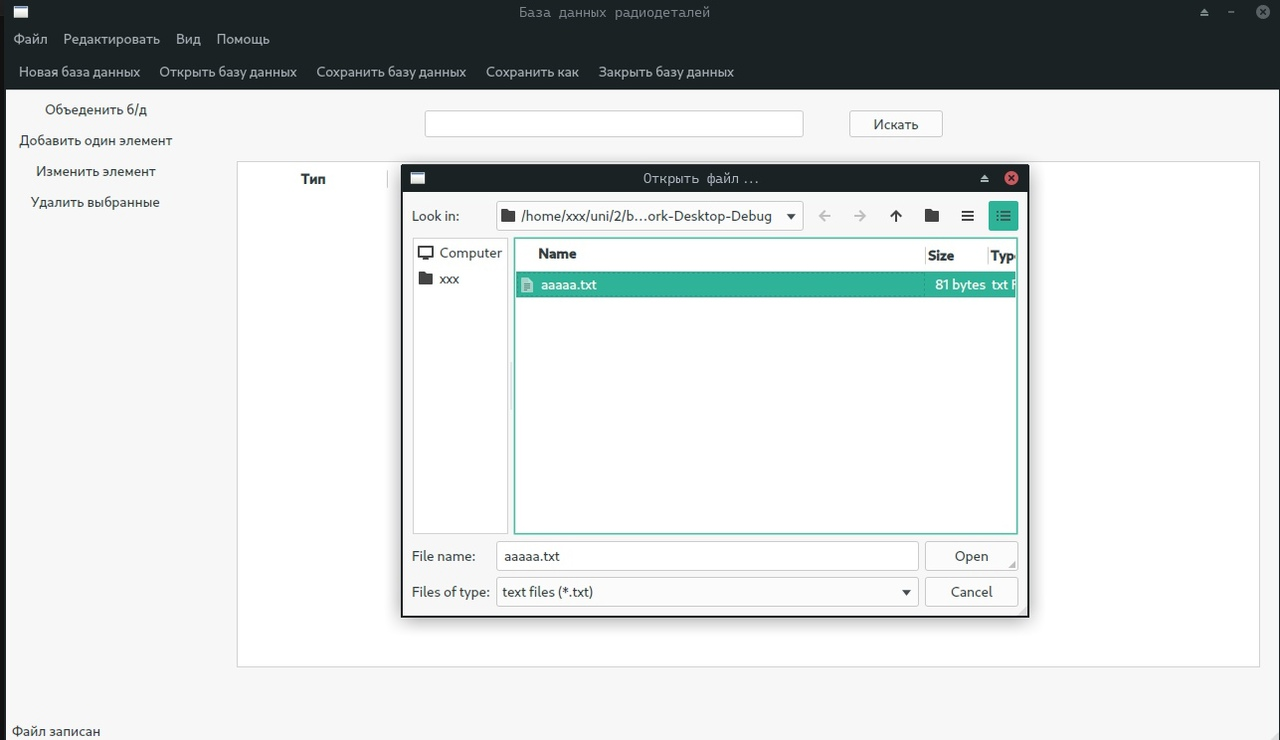
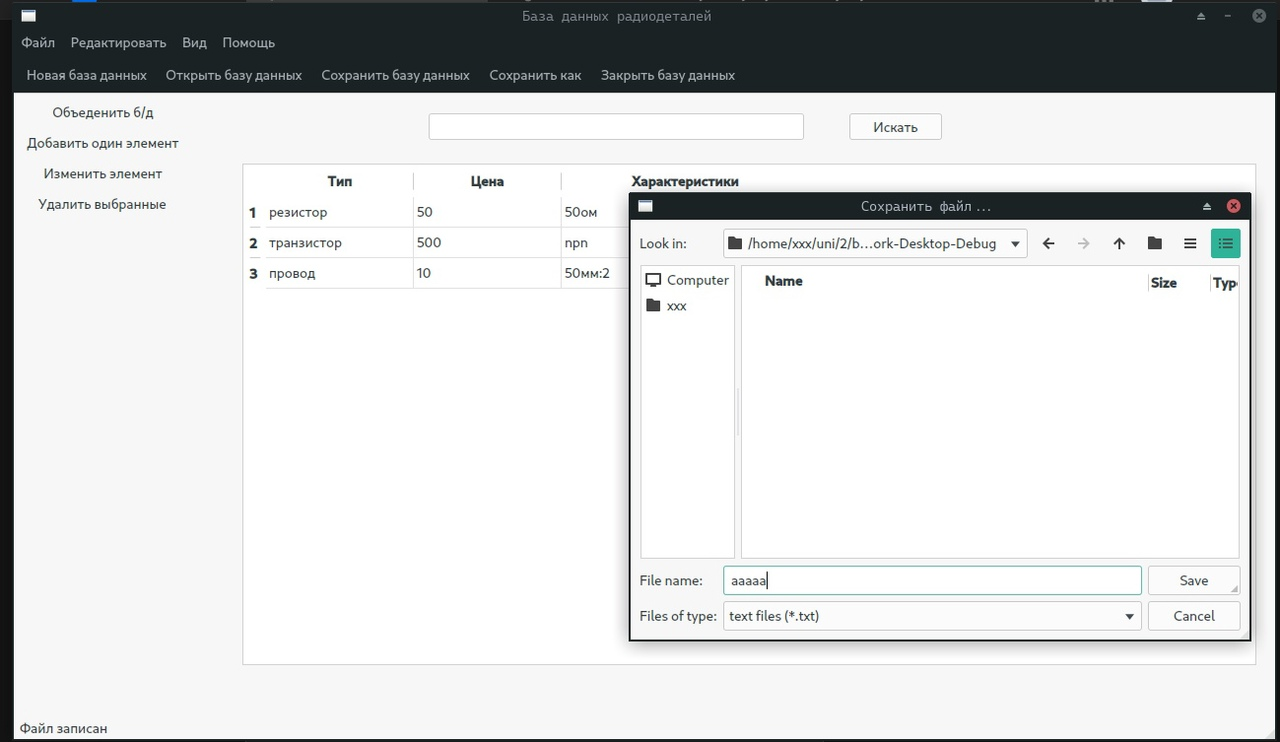
## 

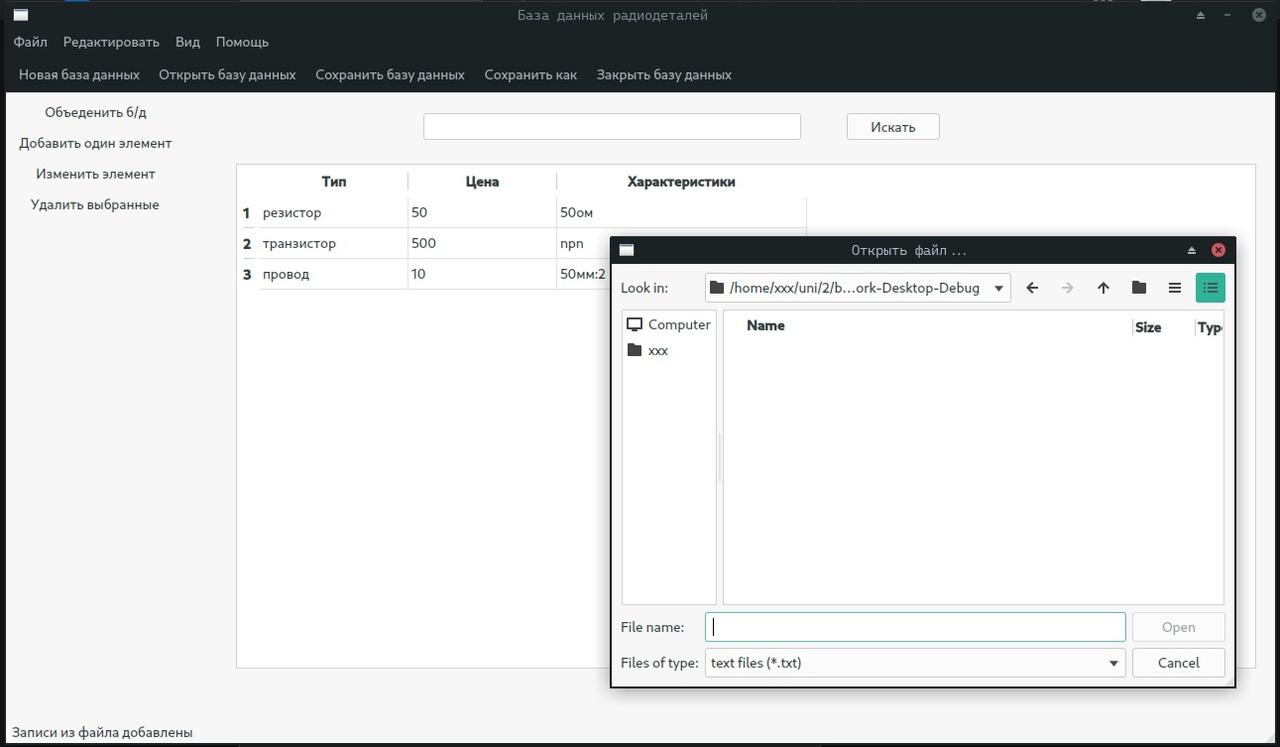
## Создание нового файла с подтверждением.

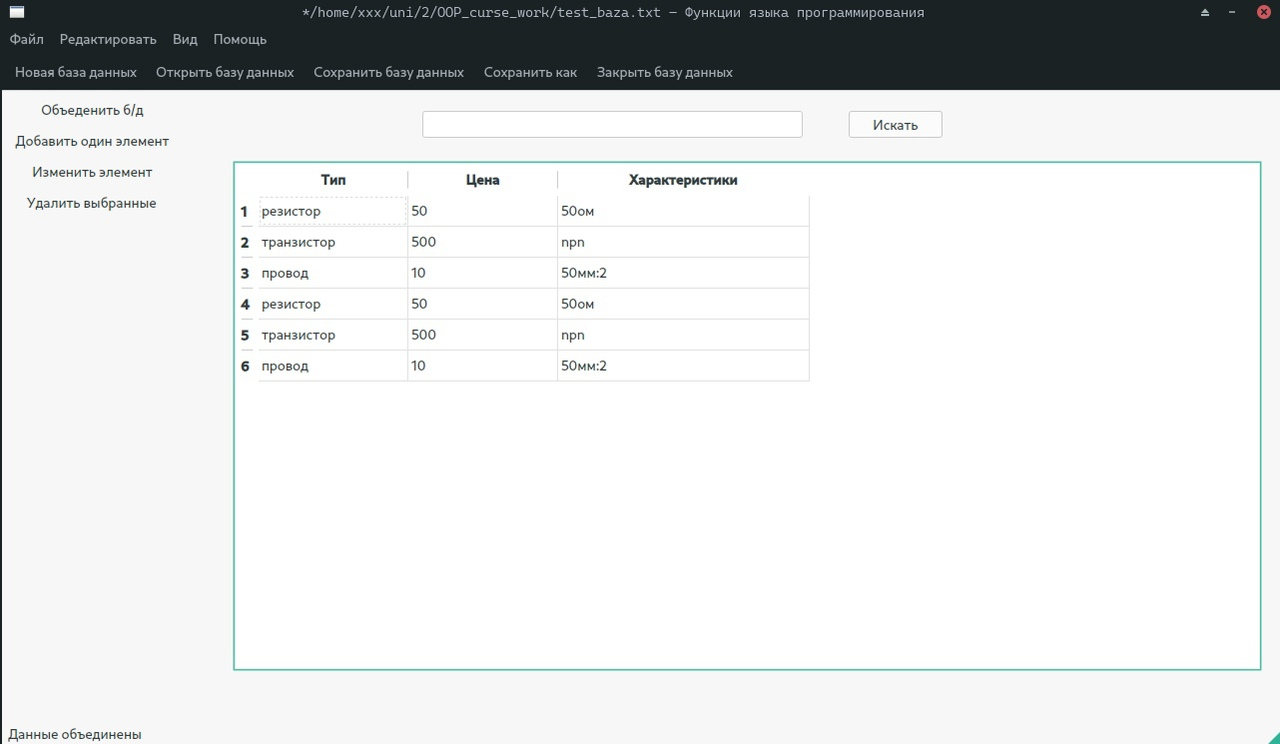
## Подтверждение не отображается, если база данных пуста и никакой файл не

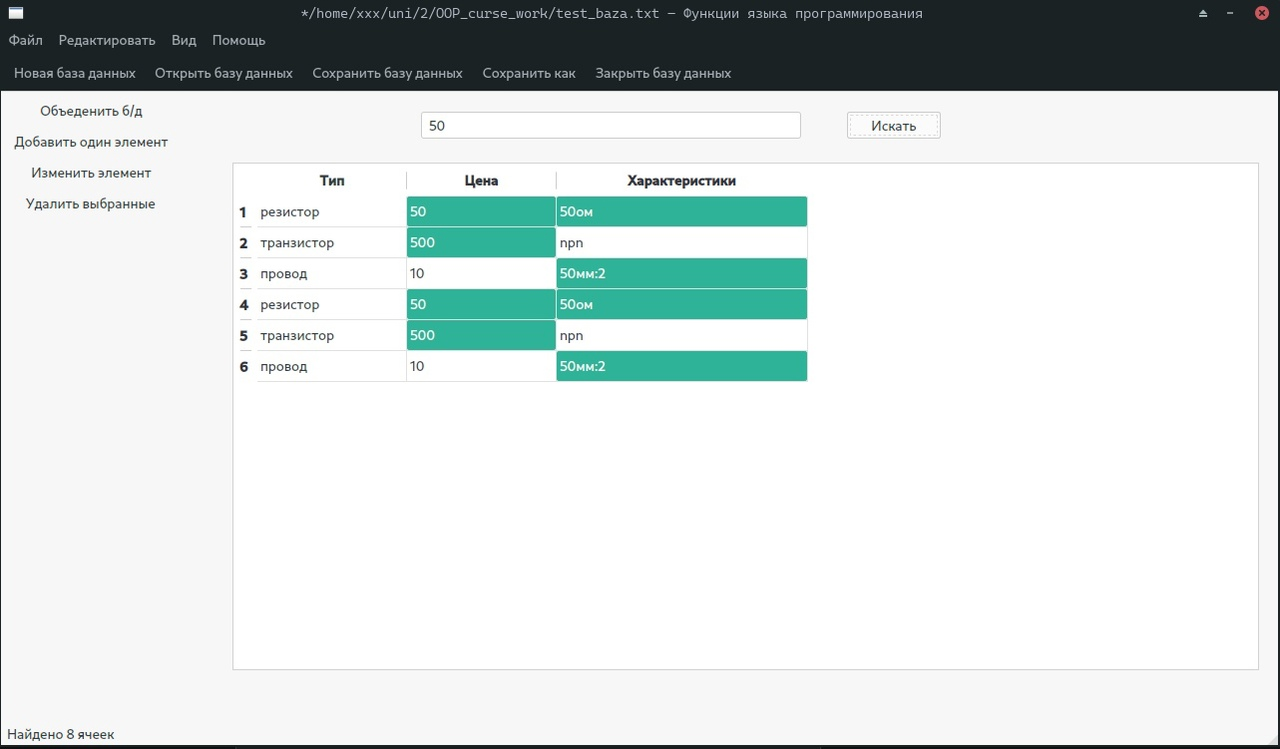
## открыт.

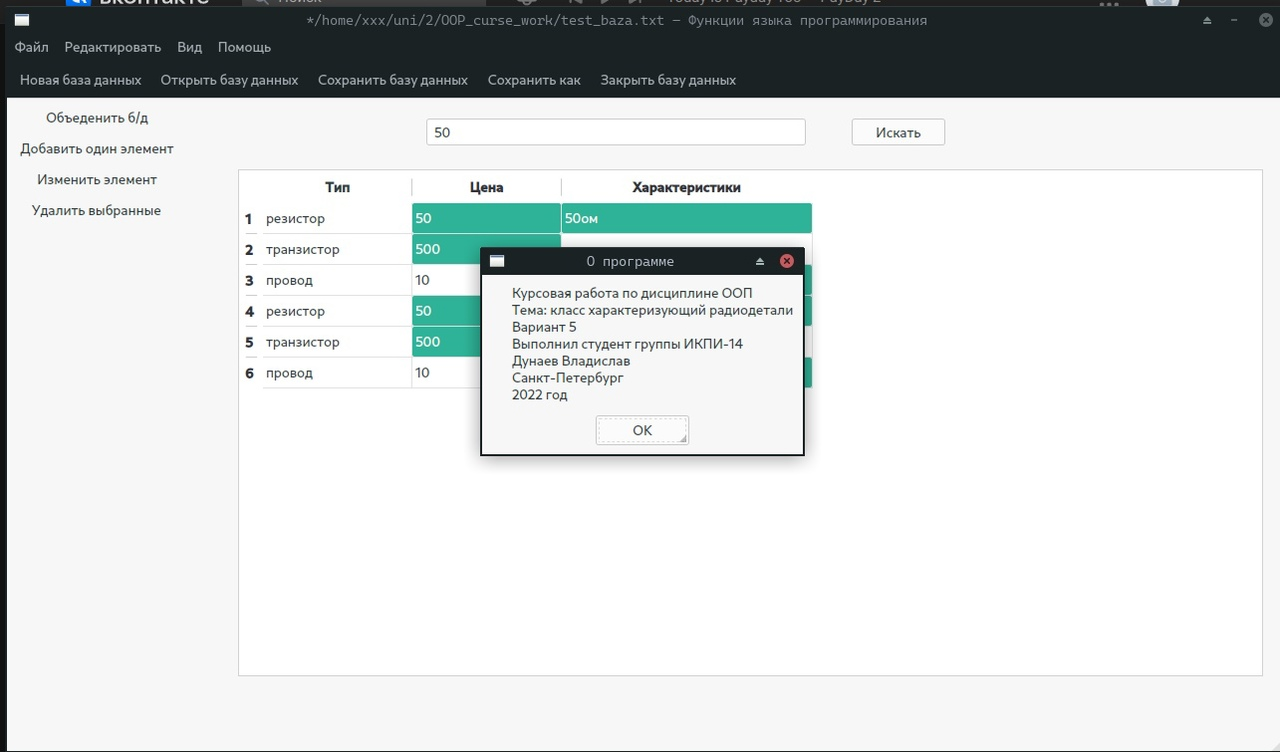
1. **Сохранение и открытие базы данных.**

****

1. **Объединение баз данных .**

****

1. **Поиск.**
2. **О программе .**

****

# Код программы

# OOP\_curse\_work.pro

# QT += core gui

# greaterThan(QT\_MAJOR\_VERSION, 4): QT += widgets

# CONFIG += c++17

# # You can make your code fail to compile if it uses deprecated APIs.

# # In order to do so, uncomment the following line.

# #DEFINES += QT\_DISABLE\_DEPRECATED\_BEFORE=0x060000 # disables all the APIs deprecated before Qt 6.0.0

# SOURCES += \

# main.cpp \

# mainwindow.cpp \

# pop\_up.cpp

# HEADERS += \

# Vector.hpp \

# mainwindow.h \

# pop\_up.h \

# radio\_components.hpp

# FORMS += \

# mainwindow.ui \

# pop\_up.ui

# # Default rules for deployment.

# qnx: target.path = /tmp/$${TARGET}/bin

# else: unix:!android: target.path = /opt/$${TARGET}/bin

# !isEmpty(target.path): INSTALLS += target

# OOP\_curse\_work.pro.user

# <?xml version="1.0" encoding="UTF-8"?>

# <!DOCTYPE QtCreatorProject>

# <!-- Written by QtCreator 9.0.1, 2022-12-29T00:30:58. -->

# <qtcreator>

# <data>

# <variable>EnvironmentId</variable>

# <value type="QByteArray">{193def09-a86b-4b95-b269-dece51ed3bd0}</value>

# </data>

# <data>

# <variable>ProjectExplorer.Project.ActiveTarget</variable>

# <value type="qlonglong">0</value>

# </data>

# <data>

# <variable>ProjectExplorer.Project.EditorSettings</variable>

# <valuemap type="QVariantMap">

# <value type="bool" key="EditorConfiguration.AutoIndent">true</value>

# <value type="bool" key="EditorConfiguration.AutoSpacesForTabs">false</value>

# <value type="bool" key="EditorConfiguration.CamelCaseNavigation">true</value>

# <valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.0">

# <value type="QString" key="language">Cpp</value>

# <valuemap type="QVariantMap" key="value">

# <value type="QByteArray" key="CurrentPreferences">CppGlobal</value>

# </valuemap>

# </valuemap>

# <valuemap type="QVariantMap" key="EditorConfiguration.CodeStyle.1">

# <value type="QString" key="language">QmlJS</value>

# <valuemap type="QVariantMap" key="value">

# <value type="QByteArray" key="CurrentPreferences">QmlJSGlobal</value>

# </valuemap>

# </valuemap>

# <value type="qlonglong" key="EditorConfiguration.CodeStyle.Count">2</value>

# <value type="QByteArray" key="EditorConfiguration.Codec">UTF-8</value>

# <value type="bool" key="EditorConfiguration.ConstrainTooltips">false</value>

# <value type="int" key="EditorConfiguration.IndentSize">4</value>

# <value type="bool" key="EditorConfiguration.KeyboardTooltips">false</value>

# <value type="int" key="EditorConfiguration.MarginColumn">80</value>

# <value type="bool" key="EditorConfiguration.MouseHiding">true</value>

# <value type="bool" key="EditorConfiguration.MouseNavigation">true</value>

# <value type="int" key="EditorConfiguration.PaddingMode">1</value>

# <value type="bool" key="EditorConfiguration.PreferSingleLineComments">false</value>

# <value type="bool" key="EditorConfiguration.ScrollWheelZooming">true</value>

# <value type="bool" key="EditorConfiguration.ShowMargin">false</value>

# <value type="int" key="EditorConfiguration.SmartBackspaceBehavior">0</value>

# <value type="bool" key="EditorConfiguration.SmartSelectionChanging">true</value>

# <value type="bool" key="EditorConfiguration.SpacesForTabs">true</value>

# <value type="int" key="EditorConfiguration.TabKeyBehavior">0</value>

# <value type="int" key="EditorConfiguration.TabSize">8</value>

# <value type="bool" key="EditorConfiguration.UseGlobal">true</value>

# <value type="bool" key="EditorConfiguration.UseIndenter">false</value>

# <value type="int" key="EditorConfiguration.Utf8BomBehavior">1</value>

# <value type="bool" key="EditorConfiguration.addFinalNewLine">true</value>

# <value type="bool" key="EditorConfiguration.cleanIndentation">true</value>

# <value type="bool" key="EditorConfiguration.cleanWhitespace">true</value>

# <value type="QString" key="EditorConfiguration.ignoreFileTypes">\*.md, \*.MD, Makefile</value>

# <value type="bool" key="EditorConfiguration.inEntireDocument">false</value>

# <value type="bool" key="EditorConfiguration.skipTrailingWhitespace">true</value>

# <value type="bool" key="EditorConfiguration.tintMarginArea">true</value>

# </valuemap>

# </data>

# <data>

# <variable>ProjectExplorer.Project.PluginSettings</variable>

# <valuemap type="QVariantMap">

# <valuemap type="QVariantMap" key="AutoTest.ActiveFrameworks">

# <value type="bool" key="AutoTest.Framework.Boost">true</value>

# <value type="bool" key="AutoTest.Framework.CTest">false</value>

# <value type="bool" key="AutoTest.Framework.Catch">true</value>

# <value type="bool" key="AutoTest.Framework.GTest">true</value>

# <value type="bool" key="AutoTest.Framework.QtQuickTest">true</value>

# <value type="bool" key="AutoTest.Framework.QtTest">true</value>

# </valuemap>

# <valuemap type="QVariantMap" key="AutoTest.CheckStates"/>

# <value type="int" key="AutoTest.RunAfterBuild">0</value>

# <value type="bool" key="AutoTest.UseGlobal">true</value>

# <valuemap type="QVariantMap" key="ClangTools">

# <value type="bool" key="ClangTools.AnalyzeOpenFiles">true</value>

# <value type="bool" key="ClangTools.BuildBeforeAnalysis">true</value>

# <value type="QString" key="ClangTools.DiagnosticConfig">Builtin.DefaultTidyAndClazy</value>

# <value type="int" key="ClangTools.ParallelJobs">6</value>

# <valuelist type="QVariantList" key="ClangTools.SelectedDirs"/>

# <valuelist type="QVariantList" key="ClangTools.SelectedFiles"/>

# <valuelist type="QVariantList" key="ClangTools.SuppressedDiagnostics"/>

# <value type="bool" key="ClangTools.UseGlobalSettings">true</value>

# </valuemap>

# </valuemap>

# </data>

# <data>

# <variable>ProjectExplorer.Project.Target.0</variable>

# <valuemap type="QVariantMap">

# <value type="QString" key="DeviceType">Desktop</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Desktop</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Desktop</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">{bdfdc62c-351c-4abd-b4d0-702830a7676e}</value>

# <value type="qlonglong" key="ProjectExplorer.Target.ActiveBuildConfiguration">0</value>

# <value type="qlonglong" key="ProjectExplorer.Target.ActiveDeployConfiguration">0</value>

# <value type="qlonglong" key="ProjectExplorer.Target.ActiveRunConfiguration">0</value>

# <valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.0">

# <value type="int" key="EnableQmlDebugging">0</value>

# <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/xxx/uni/2/OOP\_curse\_work/../build-OOP\_curse\_work-Desktop-Debug</value>

# <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/xxx/uni/2/build-OOP\_curse\_work-Desktop-Debug</value>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>

# <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>

# <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.BuildStepList.StepsCount">2</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Build</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Build</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>

# <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.BuildStepList.StepsCount">1</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Clean</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Clean</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>

# </valuemap>

# <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>

# <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>

# <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>

# <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">false</value>

# <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Debug</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>

# <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">2</value>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.1">

# <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/xxx/uni/2/OOP\_curse\_work/../build-OOP\_curse\_work-Desktop-Release</value>

# <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/xxx/uni/2/build-OOP\_curse\_work-Desktop-Release</value>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>

# <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>

# <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.BuildStepList.StepsCount">2</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Build</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Build</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>

# <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.BuildStepList.StepsCount">1</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Clean</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Clean</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>

# </valuemap>

# <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>

# <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>

# <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>

# <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">false</value>

# <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Release</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>

# <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>

# <value type="int" key="QtQuickCompiler">0</value>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.Target.BuildConfiguration.2">

# <value type="int" key="EnableQmlDebugging">0</value>

# <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory">/home/xxx/uni/2/OOP\_curse\_work/../build-OOP\_curse\_work-Desktop-Profile</value>

# <value type="QString" key="ProjectExplorer.BuildConfiguration.BuildDirectory.shadowDir">/home/xxx/uni/2/build-OOP\_curse\_work-Desktop-Profile</value>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">QtProjectManager.QMakeBuildStep</value>

# <value type="bool" key="QtProjectManager.QMakeBuildStep.QMakeForced">false</value>

# <valuelist type="QVariantList" key="QtProjectManager.QMakeBuildStep.SelectedAbis"/>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.1">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.BuildStepList.StepsCount">2</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Build</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Build</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Build</value>

# </valuemap>

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.1">

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildStepList.Step.0">

# <value type="bool" key="ProjectExplorer.BuildStep.Enabled">true</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.MakeStep</value>

# <value type="QString" key="Qt4ProjectManager.MakeStep.MakeArguments">clean</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.BuildStepList.StepsCount">1</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Clean</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Clean</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Clean</value>

# </valuemap>

# <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">2</value>

# <value type="bool" key="ProjectExplorer.BuildConfiguration.ClearSystemEnvironment">false</value>

# <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.CustomParsers"/>

# <value type="bool" key="ProjectExplorer.BuildConfiguration.ParseStandardOutput">false</value>

# <valuelist type="QVariantList" key="ProjectExplorer.BuildConfiguration.UserEnvironmentChanges"/>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Profile</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4BuildConfiguration</value>

# <value type="int" key="Qt4ProjectManager.Qt4BuildConfiguration.BuildConfiguration">0</value>

# <value type="int" key="QtQuickCompiler">0</value>

# <value type="int" key="SeparateDebugInfo">0</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.Target.BuildConfigurationCount">3</value>

# <valuemap type="QVariantMap" key="ProjectExplorer.Target.DeployConfiguration.0">

# <valuemap type="QVariantMap" key="ProjectExplorer.BuildConfiguration.BuildStepList.0">

# <value type="qlonglong" key="ProjectExplorer.BuildStepList.StepsCount">0</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DefaultDisplayName">Deploy</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.DisplayName">Deploy</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.BuildSteps.Deploy</value>

# </valuemap>

# <value type="int" key="ProjectExplorer.BuildConfiguration.BuildStepListCount">1</value>

# <valuemap type="QVariantMap" key="ProjectExplorer.DeployConfiguration.CustomData"/>

# <value type="bool" key="ProjectExplorer.DeployConfiguration.CustomDataEnabled">false</value>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">ProjectExplorer.DefaultDeployConfiguration</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.Target.DeployConfigurationCount">1</value>

# <valuemap type="QVariantMap" key="ProjectExplorer.Target.RunConfiguration.0">

# <value type="bool" key="Analyzer.Perf.Settings.UseGlobalSettings">true</value>

# <value type="bool" key="Analyzer.QmlProfiler.Settings.UseGlobalSettings">true</value>

# <value type="bool" key="Analyzer.Valgrind.Settings.UseGlobalSettings">true</value>

# <valuelist type="QVariantList" key="CustomOutputParsers"/>

# <value type="int" key="PE.EnvironmentAspect.Base">2</value>

# <valuelist type="QVariantList" key="PE.EnvironmentAspect.Changes"/>

# <value type="QString" key="ProjectExplorer.ProjectConfiguration.Id">Qt4ProjectManager.Qt4RunConfiguration:/home/xxx/uni/2/OOP\_curse\_work/OOP\_curse\_work.pro</value>

# <value type="QString" key="ProjectExplorer.RunConfiguration.BuildKey">/home/xxx/uni/2/OOP\_curse\_work/OOP\_curse\_work.pro</value>

# <value type="bool" key="RunConfiguration.UseCppDebugger">false</value>

# <value type="bool" key="RunConfiguration.UseCppDebuggerAuto">true</value>

# <value type="bool" key="RunConfiguration.UseLibrarySearchPath">true</value>

# <value type="bool" key="RunConfiguration.UseQmlDebugger">false</value>

# <value type="bool" key="RunConfiguration.UseQmlDebuggerAuto">true</value>

# <value type="QString" key="RunConfiguration.WorkingDirectory.default">/home/xxx/uni/2/build-OOP\_curse\_work-Desktop-Debug</value>

# </valuemap>

# <value type="qlonglong" key="ProjectExplorer.Target.RunConfigurationCount">1</value>

# </valuemap>

# </data>

# <data>

# <variable>ProjectExplorer.Project.TargetCount</variable>

# <value type="qlonglong">1</value>

# </data>

# <data>

# <variable>ProjectExplorer.Project.Updater.FileVersion</variable>

# <value type="int">22</value>

# </data>

# <data>

# <variable>Version</variable>

# <value type="int">22</value>

# </data>

# </qtcreator>

# Vector.hpp

# #ifndef VECTOR\_HPP

# #define VECTOR\_HPP

# #include <algorithm>

# #include <cstddef>

# #include <stdexcept>

# template <class T>

# class Vector {

# public:

# explicit Vector(int n = 0) : sz(n) {

# if (n >= cap) {

# cap = n \* 2;

# }

# array = new T[cap]{};

# }

# Vector(const Vector &other) : Vector(other.sz) {

# sz = other.sz;

# std::copy(other.array, other.array + sz, array);

# }

# Vector(Vector &&other) noexcept {

# sz = other.sz, cap = other.cap, array = other.array;

# other.sz = 0, other.cap = 0, other.array = nullptr;

# }

# Vector &operator=(const Vector &other) {

# Vector temp(other);

# swap(\*this, temp);

# return \*this;

# }

# Vector &operator=(Vector &&other) noexcept {

# if (this != &other) {

# swap(\*this, other);

# delete[] other.array, other.array = nullptr;

# other.sz = 0, other.cap = 0;

# }

# return \*this;

# }

# T &operator[](int i) { return array[i]; }

# ~Vector() { delete[] array; }

# static int size(const Vector &other) { return other.sz; }

# void add(const Vector &other, const int &k) {

# int mn = std::min(Vector::size(other), k);

# for (int i = 0; i < mn; ++i) {

# push\_back(other.array[i]);

# }

# }

# void insert(const T &d, const int &k) noexcept(false) {

# if (k == 0) {

# push\_front(d);

# return;

# } else if (k == sz) {

# push\_back(d);

# return;

# } else if (k > 0 && k < sz) {

# push\_back(d);

# T t = array[k], p;

# for (int i = k; i < sz; ++i) {

# p = array[i], array[i] = t, t = p;

# }

# array[k] = d;

# return;

# }

# throw std::out\_of\_range("Vector. Method insert. Out of range");

# }

# void push\_back(const T &d) {

# if (sz < cap) {

# array[sz] = d, ++sz;

# return;

# }

# T \*b = array;

# array = new T[cap \* 2]{};

# std::copy(b, b + sz, array);

# delete[] b;

# cap \*= 2;

# }

# void push\_front(const T &d) {

# push\_back(d);

# T t = array[0], p;

# for (int i = 1; i < sz; ++i) {

# p = array[i], array[i] = t, t = p;

# }

# array[0] = d;

# }

# void clear() {

# delete[] array;

# array = new T[10]{};

# sz = 0, cap = 10;

# }

# int size() const noexcept { return sz; }

# bool empty() const noexcept { return sz == 0; }

# T erase(int i) noexcept(false) {

# if (sz > 0 && i < sz) {

# T d = array[i];

# std::copy(array + i + 1, array + sz, array + i);

# --sz;

# return d;

# }

# throw std::out\_of\_range("Vector. Method erase. Out of range");

# }

# T pop\_front() { return erase(0); }

# T pop\_back() {

# T d = T();

# if (sz > 0) {

# d = array[sz - 1], --sz;

# }

# return d;

# }

# friend void swap(Vector &first, Vector &second) noexcept {

# std::swap(first.standart, second.standart);

# std::swap(first.array, second.array);

# std::swap(first.cap, second.cap);

# std::swap(first.sz, second.sz);

# }

# class iterator {

# public:

# T \*i;

# iterator() : i(0) {}

# explicit iterator(T \*n) : i(n) {}

# iterator(const iterator &o) : i(o.i) {}

# T &operator\*() const { return \*i; }

# T \*operator->() const { return i; }

# T &operator[](int j) const { return i[j]; }

# bool operator==(const iterator &o) const { return i == o.i; }

# bool operator!=(const iterator &o) const { return i != o.i; }

# bool operator<(const iterator &other) const { return i < other.i; }

# bool operator<=(const iterator &other) const { return i <= other.i; }

# bool operator>(const iterator &other) const { return i > other.i; }

# bool operator>=(const iterator &other) const { return i >= other.i; }

# iterator &operator++() {

# ++i;

# return \*this;

# }

# const iterator operator++(int) {

# T \*n = i;

# ++i;

# return iterator(n);

# }

# iterator &operator--() {

# i--;

# return \*this;

# }

# const iterator operator--(int) {

# T \*n = i;

# i--;

# return iterator(n);

# }

# iterator &operator+=(int j) {

# i += j;

# return \*this;

# }

# iterator &operator-=(int j) {

# i -= j;

# return \*this;

# }

# iterator operator+(int j) const { return iterator(i + j); }

# iterator operator-(int j) const { return iterator(i - j); }

# int operator-(iterator j) const { return int(i - j.i); }

# };

# Vector<T>::iterator begin() { return iterator(&array[0]); }

# Vector<T>::iterator end() { return iterator(&array[sz]); }

# private:

# int sz = 0;

# int cap = 10;

# T \*array = nullptr;

# };

# #endif

# Main.cpp

# #include "mainwindow.h"

# #include <QApplication>

# #include "Vector.hpp"

# #include "radio\_components.hpp"

# int main(int argc, char \*argv[])

# {

# QApplication a(argc, argv);

# MainWindow w;

# w.show();

# return a.exec();

# }

# Mainwindow.cpp

# #include "mainwindow.h"

# #include "pop\_up.h"

# #include "ui\_mainwindow.h"

# #include <QMessageBox>

# #include <QDebug>

# #include <QFileInfo>

# #include <QFileDialog>

# MainWindow::MainWindow(QWidget \*parent)

# : QMainWindow(parent)

# , ui(new Ui::MainWindow)

# {

# ui->setupUi(this);

# ui->tableWidget->setColumnCount(3);

# ui->tableWidget->setColumnWidth(0, 150);

# ui->tableWidget->setColumnWidth(1, 150);

# ui->tableWidget->setColumnWidth(2, 200);

# setWindowTitle("База данных радиодеталей");

# }

# MainWindow::~MainWindow()

# {

# delete ui;

# }

# void MainWindow::on\_exit\_\_triggered()

# {

# QMessageBox::StandardButton reply = QMessageBox::question(this, "Выход", "Вы действительно хотите выйти?",

# QMessageBox::Yes | QMessageBox::No);

# if (reply == QMessageBox::Yes) {

# QApplication::quit();

# } else {

# ui->statusbar->showMessage("отказался выходить");

# }

# }

# void MainWindow::on\_add\_single\_to\_bd\_triggered()

# {

# Radio single{};

# pop\_up window(nullptr, &single);

# window.setModal(true);

# window.exec();

# if (single.specifications != "" && single.cost != 0 && single.type != "") {

# ui->tableWidget->setRowCount(ui->tableWidget->rowCount() + 1);

# ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 0, new QTableWidgetItem(single.type));

# ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 1, new QTableWidgetItem(QString::number(single.cost)));

# ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 2, new QTableWidgetItem(single.specifications));

# array.push\_back(single);

# ui->statusbar->showMessage("Элемент добавлен");

# } else {

# ui->statusbar->showMessage("Элемент не добавлен");

# }

# }

# void MainWindow::on\_search\_in\_bd\_triggered()

# {

# MainWindow::on\_pushButton\_clicked();

# }

# void MainWindow::on\_pushButton\_clicked()

# {

# QString arg1 = ui->lineEdit->text();

# ui->tableWidget->setCurrentCell(-1, -1);

# if (arg1 == "") {

# return;

# }

# ui->tableWidget->setSelectionMode(QAbstractItemView::SelectionMode::MultiSelection);

# auto find\_items = ui->tableWidget->findItems(arg1, Qt::MatchContains);

# int len = find\_items.size();

# for (int i = 0; i < len; ++i) {

# auto item = find\_items.at(i);

# ui->tableWidget->setItemSelected(item, true);

# }

# ui->tableWidget->setSelectionMode(QAbstractItemView::SelectionMode::SingleSelection);

# ui->statusbar->showMessage("Найдено " + QString::number(len) + " ячеек");

# }

# int MainWindow::on\_new\_bd\_triggered()

# {

# if (ui->tableWidget->rowCount() > 0) {

# QMessageBox msgBox;

# msgBox.setWindowTitle("Внимание");

# msgBox.setText("Вы уверены, что хотите создать новый файл?\nУ вас еще остались не сохраненные данные.");

# msgBox.setIcon(QMessageBox::Question);

# msgBox.setStandardButtons(QMessageBox::Yes | QMessageBox::No);

# msgBox.setDefaultButton(QMessageBox::Yes);

# if (msgBox.exec() == QMessageBox::No) {

# return 0;

# }

# }

# ui->tableWidget->setRowCount(0);

# array.clear();

# current\_file = "";

# return 1;

# }

# void MainWindow::on\_author\_triggered()

# {

# QMessageBox::about(this,

# "О программе",

# "Курсовая работа по дисциплине ООП\n"

# "Тема: класс характеризующий радиодетали \n"

# "Вариант 5\n"

# "Выполнил студент группы ИКПИ-14\n"

# "Дунаев Владислав\n"

# "Санкт-Петербург\n"

# "2022 год");

# }

# void MainWindow::on\_delete\_some\_triggered()

# {

# auto list = ui->tableWidget->selectionModel()->selectedRows();

# if (list.size() > 0) {

# QMessageBox msgBox;

# msgBox.setWindowTitle("Внимание");

# msgBox.setText("Вы уверены, что хотите удалить эти строки?");

# msgBox.setIcon(QMessageBox::Question);

# msgBox.setStandardButtons(QMessageBox::Yes | QMessageBox::No);

# msgBox.setDefaultButton(QMessageBox::Yes);

# if (msgBox.exec() == QMessageBox::Yes) {

# for (auto k = list.rbegin(); k != list.rend(); ++k) {

# auto idx = (\*k).row();

# array.erase(idx);

# ui->tableWidget->removeRow(idx);

# }

# ui->statusbar->showMessage("Запись удалена");

# }

# } else ui->statusbar->showMessage("Выделите всю строку");

# }

# void MainWindow::on\_close\_bd\_triggered()

# {

# if (ui->tableWidget->rowCount() > 0) {

# QMessageBox msgBox;

# msgBox.setWindowTitle("Внимание");

# msgBox.setText("Вы уверены, что хотите закрыть таблицу?");

# msgBox.setIcon(QMessageBox::Question);

# msgBox.setStandardButtons(QMessageBox::Yes | QMessageBox::No);

# msgBox.setDefaultButton(QMessageBox::Yes);

# if (msgBox.exec() == QMessageBox::Yes) {

# ui->tableWidget->setRowCount(0);

# array.clear();

# ui->statusbar->showMessage("Таблица закрыта");

# current\_file = "";

# }

# } else ui->statusbar->showMessage("Таблица и так пустая");

# }

# void MainWindow::on\_open\_bd\_triggered(bool isflag = false)

# {

# QString file\_name = QFileDialog::getOpenFileName(this, "Открыть файл...", ".", "text files (\*.txt);;All files (\*.\*);;");

# QFileInfo check\_file(file\_name);

# if (check\_file.exists() && check\_file.isFile()) {

# if (!isflag) {

# if(!MainWindow::on\_new\_bd\_triggered()) {

# return;

# }

# }

# QFile file(file\_name);

# if (file.open(QIODevice::ReadOnly)) {

# QTextStream in(&file);

# while (!in.atEnd()) {

# Radio t;

# QString line = in.readLine();

# QStringList fields = line.split(";");

# if (fields.size() == 3) {

# t.type = fields[0].simplified();

# t.cost = fields[1].simplified().toInt();

# t.specifications = fields[2].simplified();

# ui->tableWidget->setRowCount(ui->tableWidget->rowCount() + 1);

# ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 0, new QTableWidgetItem(t.type));

# ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 1, new QTableWidgetItem(QString::number(t.cost)));

# ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 2, new QTableWidgetItem(t.specifications));

# array.push\_back(t);

# }

# }

# file.close();

# ui->statusbar->showMessage("Записи из файла добавлены");

# current\_file = file\_name;

# } else ui->statusbar->showMessage("Файл '" + file\_name + "' не удалось открыть на чтение (" + file.errorString() + ")");

# } else ui->statusbar->showMessage("Файл '" + file\_name + "' не существует");

# }

# void MainWindow::on\_save\_bd\_triggered()

# {

# if (current\_file == "") {

# MainWindow::on\_save\_as\_bd\_triggered();

# return;

# }

# QFile file(current\_file);

# if (file.open(QIODevice::WriteOnly)) {

# QTextStream out(&file);

# int len = ui->tableWidget->rowCount();

# for (int i = 0; i < len; ++i) {

# QString type = ui->tableWidget->item(i, 0)->text();

# QString cost = ui->tableWidget->item(i, 1)->text();

# QString specs = ui->tableWidget->item(i, 2)->text();

# out << type << ";" << cost << ";" << specs << "\n";

# }

# file.close();

# ui->statusbar->showMessage("Файл записан");

# }

# }

# void MainWindow::on\_save\_as\_bd\_triggered()

# {

# QString file\_name = QFileDialog::getSaveFileName(this, "Сохранить файл...", ".", "text files (\*.txt);;All files (\*.\*);;");

# if(!file\_name.endsWith(QString(".txt")))

# file\_name.append(QString(".txt"));

# if (file\_name != "") {

# current\_file = file\_name;

# MainWindow::on\_save\_bd\_triggered();

# }

# }

# void MainWindow::on\_edit\_single\_in\_bd\_triggered()

# {

# auto list = ui->tableWidget->selectedItems();

# if (list.size() == 3) {

# Radio single{};

# single.type = list.at(0)->text();

# single.cost = list.at(1)->text().toInt();

# single.specifications = list.at(2)->text();

# pop\_up window(nullptr, &single);

# window.setModal(true);

# window.exec();

# if (single.type != "" || single.cost != 0 || single.specifications != "")

# {

# list.at(0)->setText(single.type);

# list.at(1)->setText(QString::number(single.cost));

# list.at(2)->setText(single.specifications);

# array[list.at(0)->row()] = single;

# ui->statusbar->showMessage("Запись отредактирована");

# }

# }

# ui->statusbar->showMessage("Выделите всю строку");

# }

# void MainWindow::on\_help\_and\_tips\_triggered()

# {

# QMessageBox::information(this, "Help", "Подсказки не нужны\nИ так все просто и понятно");

# }

# void MainWindow::on\_merge\_bd\_triggered()

# {

# QString old\_file = current\_file;

# MainWindow::on\_open\_bd\_triggered(true);

# current\_file = old\_file;

# ui->statusbar->showMessage("Данные объединены");

# setWindowTitle("\*" + current\_file + " – Функции языка программирования");

# }

# Mainwindow.h

# #ifndef MAINWINDOW\_H

# #define MAINWINDOW\_H

# #include <QMainWindow>

# #include "Vector.hpp"

# #include "radio\_components.hpp"

# QT\_BEGIN\_NAMESPACE

# namespace Ui { class MainWindow; }

# QT\_END\_NAMESPACE

# class MainWindow : public QMainWindow

# {

# Q\_OBJECT

# public:

# MainWindow(QWidget \*parent = nullptr);

# ~MainWindow();

# private slots:

# void on\_exit\_\_triggered();

# void on\_add\_single\_to\_bd\_triggered();

# void on\_search\_in\_bd\_triggered();

# void on\_pushButton\_clicked();

# int on\_new\_bd\_triggered();

# void on\_author\_triggered();

# void on\_delete\_some\_triggered();

# void on\_close\_bd\_triggered();

# void on\_open\_bd\_triggered(bool);

# void on\_save\_bd\_triggered();

# void on\_save\_as\_bd\_triggered();

# void on\_edit\_single\_in\_bd\_triggered();

# void on\_help\_and\_tips\_triggered();

# void on\_merge\_bd\_triggered();

# private:

# Ui::MainWindow \*ui;

# QString current\_file = "";

# Vector<Radio> array;

# };

# #endif // MAINWINDOW\_H

# Mainwindow.ui

# <?xml version="1.0" encoding="UTF-8"?>

# <ui version="4.0">

# <class>MainWindow</class>

# <widget class="QMainWindow" name="MainWindow">

# <property name="geometry">

# <rect>

# <x>0</x>

# <y>0</y>

# <width>1280</width>

# <height>720</height>

# </rect>

# </property>

# <property name="minimumSize">

# <size>

# <width>1280</width>

# <height>720</height>

# </size>

# </property>

# <property name="maximumSize">

# <size>

# <width>1280</width>

# <height>720</height>

# </size>

# </property>

# <property name="windowTitle">

# <string>MainWindow</string>

# </property>

# <widget class="QWidget" name="centralwidget">

# <widget class="QTableWidget" name="tableWidget">

# <property name="geometry">

# <rect>

# <x>50</x>

# <y>70</y>

# <width>1031</width>

# <height>511</height>

# </rect>

# </property>

# <property name="minimumSize">

# <size>

# <width>1031</width>

# <height>511</height>

# </size>

# </property>

# <column>

# <property name="text">

# <string>Тип</string>

# </property>

# </column>

# <column>

# <property name="text">

# <string>Цена</string>

# </property>

# </column>

# <column>

# <property name="text">

# <string>Характеристики</string>

# </property>

# </column>

# </widget>

# <widget class="QWidget" name="layoutWidget">

# <property name="geometry">

# <rect>

# <x>240</x>

# <y>20</y>

# <width>521</width>

# <height>27</height>

# </rect>

# </property>

# <layout class="QHBoxLayout" name="horizontalLayout">

# <item>

# <widget class="QLineEdit" name="lineEdit">

# <property name="text">

# <string/>

# </property>

# </widget>

# </item>

# <item>

# <spacer name="horizontalSpacer">

# <property name="orientation">

# <enum>Qt::Horizontal</enum>

# </property>

# <property name="sizeType">

# <enum>QSizePolicy::Minimum</enum>

# </property>

# <property name="sizeHint" stdset="0">

# <size>

# <width>40</width>

# <height>20</height>

# </size>

# </property>

# </spacer>

# </item>

# <item>

# <widget class="QPushButton" name="pushButton">

# <property name="text">

# <string>Искать</string>

# </property>

# </widget>

# </item>

# </layout>

# </widget>

# </widget>

# <widget class="QMenuBar" name="menubar">

# <property name="geometry">

# <rect>

# <x>0</x>

# <y>0</y>

# <width>1280</width>

# <height>22</height>

# </rect>

# </property>

# <widget class="QMenu" name="menu">

# <property name="title">

# <string>Файл</string>

# </property>

# <addaction name="new\_bd"/>

# <addaction name="open\_bd"/>

# <addaction name="save\_bd"/>

# <addaction name="save\_as\_bd"/>

# <addaction name="close\_bd"/>

# <addaction name="separator"/>

# <addaction name="exit\_"/>

# </widget>

# <widget class="QMenu" name="menu\_2">

# <property name="title">

# <string>Редактировать</string>

# </property>

# <addaction name="edit\_single\_in\_bd"/>

# <addaction name="add\_single\_to\_bd"/>

# <addaction name="delete\_some"/>

# </widget>

# <widget class="QMenu" name="menu\_3">

# <property name="title">

# <string>Вид</string>

# </property>

# <addaction name="search\_in\_bd"/>

# </widget>

# <widget class="QMenu" name="menu\_4">

# <property name="title">

# <string>Помощь</string>

# </property>

# <addaction name="help\_and\_tips"/>

# <addaction name="author"/>

# </widget>

# <addaction name="menu"/>

# <addaction name="menu\_2"/>

# <addaction name="menu\_3"/>

# <addaction name="menu\_4"/>

# </widget>

# <widget class="QStatusBar" name="statusbar"/>

# <widget class="QToolBar" name="toolBar">

# <property name="windowTitle">

# <string>toolBar</string>

# </property>

# <attribute name="toolBarArea">

# <enum>TopToolBarArea</enum>

# </attribute>

# <attribute name="toolBarBreak">

# <bool>false</bool>

# </attribute>

# <addaction name="new\_bd"/>

# <addaction name="open\_bd"/>

# <addaction name="save\_bd"/>

# <addaction name="save\_as\_bd"/>

# <addaction name="close\_bd"/>

# </widget>

# <widget class="QToolBar" name="toolBar\_2">

# <property name="windowTitle">

# <string>toolBar\_2</string>

# </property>

# <attribute name="toolBarArea">

# <enum>LeftToolBarArea</enum>

# </attribute>

# <attribute name="toolBarBreak">

# <bool>false</bool>

# </attribute>

# <addaction name="merge\_bd"/>

# <addaction name="add\_single\_to\_bd"/>

# <addaction name="edit\_single\_in\_bd"/>

# <addaction name="delete\_some"/>

# </widget>

# <action name="new\_bd">

# <property name="text">

# <string>Новая база данных</string>

# </property>

# </action>

# <action name="open\_bd">

# <property name="text">

# <string>Открыть базу данных</string>

# </property>

# </action>

# <action name="save\_bd">

# <property name="text">

# <string>Сохранить базу данных</string>

# </property>

# </action>

# <action name="close\_bd">

# <property name="text">

# <string>Закрыть базу данных</string>

# </property>

# </action>

# <action name="exit\_">

# <property name="text">

# <string>Выход</string>

# </property>

# </action>

# <action name="help\_and\_tips">

# <property name="text">

# <string>Подсказки по командам</string>

# </property>

# </action>

# <action name="author">

# <property name="text">

# <string>АФТОР!!!</string>

# </property>

# </action>

# <action name="view\_bd">

# <property name="text">

# <string>Просмотр б/д</string>

# </property>

# </action>

# <action name="add\_single\_to\_bd">

# <property name="text">

# <string>Добавить один элемент</string>

# </property>

# </action>

# <action name="merge\_bd">

# <property name="text">

# <string>Объеденить б/д</string>

# </property>

# </action>

# <action name="edit\_single\_in\_bd">

# <property name="text">

# <string>Изменить элемент</string>

# </property>

# </action>

# <action name="search\_in\_bd">

# <property name="text">

# <string>Поиск данных</string>

# </property>

# </action>

# <action name="save\_as\_bd">

# <property name="text">

# <string>Сохранить как</string>

# </property>

# </action>

# <action name="delete\_some">

# <property name="text">

# <string>Удалить выбранные</string>

# </property>

# </action>

# </widget>

# <resources/>

# <connections/>

# </ui>

# pop\_up.h

# #ifndef POP\_UP\_H

# #define POP\_UP\_H

# #include <QDialog>

# #include "radio\_components.hpp"

# namespace Ui {

# class pop\_up;

# }

# class pop\_up : public QDialog

# {

# Q\_OBJECT

# public:

# explicit pop\_up(QWidget \*parent = nullptr);

# explicit pop\_up(QWidget \*paretn = nullptr, Radio \* other = nullptr);

# ~pop\_up();

# private slots:

# void on\_pushButton\_clicked();

# private:

# Ui::pop\_up \*ui;

# Radio \* single;

# };

# #endif // POP\_UP\_H

# pop\_up.cpp

# #include "pop\_up.h"

# #include "ui\_pop\_up.h"

# #include "radio\_components.hpp"

# pop\_up::pop\_up(QWidget \*parent) :

# QDialog(parent),

# ui(new Ui::pop\_up)

# {

# ui->setupUi(this);

# }

# pop\_up::pop\_up(QWidget \*parent, Radio \* other) :

# QDialog(parent),

# ui(new Ui::pop\_up)

# {

# single = other;

# ui->setupUi(this);

# ui->lineEdit->setText(single->type);

# ui->lineEdit\_3->setText(QString::number(single->cost));

# ui->lineEdit\_2->setText(single->specifications);

# }

# pop\_up::~pop\_up()

# {

# delete ui;

# }

# void pop\_up::on\_pushButton\_clicked()

# {

# QString type = ui->lineEdit->text();

# QString cost = ui->lineEdit\_3->text();

# QString specs = ui->lineEdit\_2->text();

# type.replace(";",",").simplified();

# cost.replace(";",",").simplified();

# specs.replace(";",",").simplified();

# bool check = false;

# int x = cost.toInt(&check);

# if (!check) {

# ui->lineEdit\_3->setText("Это не число!!!");

# }

# this->single->type = type;

# this->single->cost = x;

# this->single->specifications = specs;

# if (check) {

# this->close();

# }

# }

# pop\_up.ui

# <?xml version="1.0" encoding="UTF-8"?>

# <ui version="4.0">

# <class>pop\_up</class>

# <widget class="QDialog" name="pop\_up">

# <property name="geometry">

# <rect>

# <x>0</x>

# <y>0</y>

# <width>588</width>

# <height>293</height>

# </rect>

# </property>

# <property name="windowTitle">

# <string>Dialog</string>

# </property>

# <layout class="QVBoxLayout" name="verticalLayout\_5">

# <item>

# <layout class="QVBoxLayout" name="verticalLayout\_3">

# <item>

# <widget class="QLabel" name="label">

# <property name="minimumSize">

# <size>

# <width>309</width>

# <height>32</height>

# </size>

# </property>

# <property name="text">

# <string>Тип</string>

# </property>

# </widget>

# </item>

# <item>

# <widget class="QLineEdit" name="lineEdit">

# <property name="minimumSize">

# <size>

# <width>309</width>

# <height>24</height>

# </size>

# </property>

# </widget>

# </item>

# <item>

# <widget class="QLabel" name="label\_2">

# <property name="minimumSize">

# <size>

# <width>309</width>

# <height>32</height>

# </size>

# </property>

# <property name="text">

# <string>Цена</string>

# </property>

# </widget>

# </item>

# <item>

# <widget class="QLineEdit" name="lineEdit\_3">

# <property name="minimumSize">

# <size>

# <width>309</width>

# <height>24</height>

# </size>

# </property>

# </widget>

# </item>

# <item>

# <widget class="QLabel" name="label\_3">

# <property name="minimumSize">

# <size>

# <width>309</width>

# <height>32</height>

# </size>

# </property>

# <property name="text">

# <string>Характеристики</string>

# </property>

# </widget>

# </item>

# <item>

# <widget class="QLineEdit" name="lineEdit\_2">

# <property name="minimumSize">

# <size>

# <width>309</width>

# <height>24</height>

# </size>

# </property>

# </widget>

# </item>

# <item>

# <spacer name="verticalSpacer">

# <property name="orientation">

# <enum>Qt::Vertical</enum>

# </property>

# <property name="sizeHint" stdset="0">

# <size>

# <width>20</width>

# <height>20</height>

# </size>

# </property>

# </spacer>

# </item>

# <item>

# <widget class="QPushButton" name="pushButton">

# <property name="minimumSize">

# <size>

# <width>309</width>

# <height>25</height>

# </size>

# </property>

# <property name="text">

# <string>Сохранить</string>

# </property>

# </widget>

# </item>

# <item>

# <spacer name="verticalSpacer\_2">

# <property name="orientation">

# <enum>Qt::Vertical</enum>

# </property>

# <property name="sizeHint" stdset="0">

# <size>

# <width>20</width>

# <height>20</height>

# </size>

# </property>

# </spacer>

# </item>

# </layout>

# </item>

# </layout>

# </widget>

# <resources/>

# <connections/>

# </ui>

# radio\_components.hpp

# #ifndef RADIO\_HPP

# #define RADIO\_HPP

# #include <QString>

# class Radio {

# public:

# Radio() : type(""), cost{}, specifications(""){};

# Radio(QString &t, int c, QString &s)

# : type(t), cost(c), specifications(s){};

# Radio(const Radio &other)

# : type(other.type),

# cost(other.cost),

# specifications(other.specifications){};

# Radio &operator=(const Radio &other) {

# type = other.type;

# cost = other.cost;

# specifications = other.specifications;

# return \*this;

# }

# QString type;

# int cost;

# QString specifications;

# };

# #endif

# Заключение

## Я научился работать с языком с++, библиотекой qt и редактором qt creator. Z Я разработал программу, которая выполняет все поставленные задачи: работает с базами данных (открытие, сохранение, добавление записи, удаление записей, редактирование записи и т. д.), работает на основе собственно реализованного вектора, а так же имеет интуитивно- понятное отображение всех возможностей программы.

# Список литературы

## Коробов С.А. Лекции по дисциплине "Объектно-ориентированное программирование".

## Справочные материалы QT Creator.