ФГБОУ ВО РГАТУ имени П.А. Соловьева

Авиационный колледж

УТВЕРЖДАЮ

МП

**ОТЧЕТ**

**О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ ПП.01  
(ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)**

Специальность ***09.02.03 Программирование в компьютерных системах***

**Место прохождения практики:** ООО НПО «Криста»

*(наименование организации полностью)*

**Руководитель практики от организации:**

Сапрыкин Максим Александрович

*(Ф.И.О., должность) (подпись)*

**Руководитель практики от колледжа:**

Кустова Екатерина Андреевна

преподаватель

*(Ф.И.О., должность) (подпись)*

**Работу выполнил студент группы** ПР-23

Чирков Михаил Дмитриевич

*(Ф.И.О.) (подпись)*

**Оценка за практику:**

Рыбинск, 2022 г.

**Рекомендации по оформлению отчета по практике**

*Требования к оформлению текста отчета:*

* При печатании следует руководствоваться СТП 1.01–2002, 1.02–2002, 1.03–2002.
* Отчет пишется:
* от первого лица (множественное число);
* оформляется на компьютере шрифтом Times New Roman;
* поля документа: левое – 3, верхнее, нижнее – 2, правое – 1,5;
* отступ первой строки – 1,25 см;
* размер шрифта — 14 пт;
* межстрочный интервал — 1,5;
* расположение номера страниц — в центре нижней части листа без точки;
* нумерация страниц на первом листе (титульном) не ставится;
* выравнивание текста по ширине;
* Графики и схемы, помещённые в тексте, выполняются с необходимыми пояснениями и входят в общую нумерацию страниц.
* Иллюстрации и таблицы, расположенные на отдельных листах, включают в общую нумерацию страниц. На все иллюстрации должны быть даны ссылки в работе.
* При делении текста отчёта на пункты и подпункты необходимо, чтобы каждый пункт содержал законченную информацию.
* Каждый отчет выполняется индивидуально.
* Содержание отчета формируется в папке-скоросшиватель.

*Требования к содержанию отчета:*

* Титульный лист по практике *(подписывается руководителем практики от организации + печать)*.
* Личная карточка инструктажа *(подписывается)*.
* Индивидуальное задание *(подписывается руководителем практики от организации + печать)*..
* Аттестационный лист по производственной практике *(подписывается руководителем практики от организации)*.
* Приложение к аттестационному листу по производственной практике *(подписывается руководителем практики от организации + печать)*.
* Характеристика на обучающегося по освоению общих компетенций. *(подписывается руководителем практики от организации + печать)*.
* Отзыв студента о прохождении производственной практики *(подписывается студентом)*.
* Замечания руководителя практики от колледжа.
* Календарно-тематический план *(подписывается руководителем практики от организации)*.
* Дневник прохождения производственной практики *(подписывается руководителем практики от организации + печать)*.
* Отчет о прохождении производственной практики
* Список использованных источников.
* Приложения (если есть).

**ЛИЧНАЯ КАРТОЧКА ИНСТРУКТАЖА ПО БЕЗОПАСНЫМ МЕТОДАМ РАБОТЫ И ПРОТИВОПОЖАРНОЙ БЕЗОПАСНОСТИ**

**1. Вводный инструктаж**

|  |  |
| --- | --- |
| Провёл инженер по охране труда и технике безопасности  \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *подпись, Фамилия И.О.*  \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г. | Инструктаж получил (а) и усвоил (а)    \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *подпись, Фамилия И.О.*  \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г. |

**2. Первичный инструктаж на рабочем месте**

|  |  |
| --- | --- |
| Провёл инженер по охране труда и технике безопасности  \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *подпись, Фамилия И.О.*  \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г. | Инструктаж получил (а) и усвоил (а)    \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *подпись, Фамилия И.О.*  \_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_\_г. |

**3. Разрешение на допуск к работе**

Разрешено допустить к самостоятельной работе

Начальник цеха (отдела)

*подпись Фамилия И.О.*

СОГЛАСОВАНО УТВЕРЖДАЮ

руководитель от организации председатель ПЦК «Программирования»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

МП

ФГБОУ ВО РГАТУ им. П.А. Соловьева

Авиационный колледж

***ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ***

НА ПРОИЗВОДСТВЕННУЮ ПРАКТИКУ ПП.01

(ПО ПРОФИЛЮ СПЕЦИАЛЬНОСТИ)

**09.02.03 Программирование в компьютерных системах**

в организации ООО НПО «Криста»

студенту Чиркову Михаилу Дмитриевичу

Начало практики 30.01 Окончание практики 11.02

В процессе производственной практики студент должен познакомиться с:

* организационной структурой предприятия;
* функциями подразделений и их взаимосвязью;
* правилами внутреннего распорядка предприятия;
* правилами охраны труда и противопожарной безопасности.

В процессе практики студент должен:

* разрабатывать алгоритм поставленной задачи и реализовать его средствами автоматизированного проектирования (постановка задачи: Разработка тестового программного модуля ведения документов по основной бюджетной классификации

);

* создавать программу по разработанному алгоритму как отдельный модуль;
* осуществлять разработку кода программного модуля на современных языках программирования (используемые языки программирования: Object Pascal

);

* разрабатывать код программного продукта на основе готовой спецификации на уровне модуля;
* выполнять отладку и тестирование программы на уровне модуля;
* использовать инструментальные средства на этапе отладки программного продукта;
* оформлять документацию на программные средства;
* использовать инструментальные средства для автоматизации оформления документации.

Руководитель практики:

От предприятия: / /

*(подпись) (дата) (фамилия, имя, отчество)*

От колледжа: / /

*(подпись) (дата) (фамилия, имя, отчество)*

**Аттестационный лист по производственной практике ПП.01**

Ф.И.О. Чирков Михаил Дмитриевич

Курс 4 группа ПР-23 .

Обучающийся(аяся) по специальности СПО

*09.02.03 Программирование в компьютерных системах*

*(код и наименование)*

прошел(ла) производственную практику по профессиональному модулю

*ПМ.01 Разработка программных модулей программного обеспечения для компьютерных систем* в объеме *72*  часа

с 30.01 по 11.02 20\_\_\_ г.

в организации ООО НПО «Криста»

*(наименование организации, юридический адрес)*

Виды и качество выполнения работ:

|  |  |
| --- | --- |
| **Виды работ, выполненные обучающимся во время практики** | **Качество выполнения работ в соответствии с технологией и (или) требованиями организации, в которой проходила практика**  *(выполнено/не выполнено)* |
| Прохождение инструктажа:   1. вводного, 2. на рабочем месте, 3. по технике безопасности |  |
| Ознакомительная экскурсия по предприятию. Изучить следующие вопросы:   * структура предприятия; * назначение и место каждого подразделения в производственном и управленческом процессах, их взаимосвязь; * правила внутреннего трудового распорядка; * функции главных специалистов предприятия; * перспективы развития производства; * план освоения новых технологий. |  |
| Информационные мероприятия по ознакомлению с оборудованием и технологией. |  |
| Выполнение заданий согласно специфики предприятия:   * разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования; * создание программы по разработанному алгоритму как отдельный модуль; * разработка кода программного модуля на современных языках программирования; * разработка кода программного продукта на основе готовой спецификации на уровне модуля; * выполнение отладки и тестирование программы на уровне модуля; * использование инструментальных средств на этапе отладки программного продукта; * оформление документации на программные средства; * использование инструментальных средств для автоматизации оформления документации. |  | |
| Групповые консультации с руководителем практики. |  | |

У обучающегося были сформированы/не сформированы профессиональные компетенции, отраженные в Приложении к аттестационному листу по производственной практике.

**ПРИЛОЖЕНИЕ**

**к аттестационному листу по производственной практике ПП.01**

У обучающегося/обучающейся Чиркова Михаила Дмитриевича были сформированы следующие профессиональные компетенции:

|  |  |  |
| --- | --- | --- |
| **Название**  **профессиональной (ПК) компетенции** | **Требования к умениям и практическому опыту** | **Оценка ПК**  *(освоена/*  *не освоена)* |
| ПК1.1 Выполнять разработку спецификаций отдельных компонент. | - разработка алгоритма поставленной задачи и реализации его средствами автоматизированного проектирования; | *освоена* |
| - осуществлять разработку кода программного модуля на современных языках программирования | *освоена* |
| ПК1.2 Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля. | - разработка кода программного продукта на основе готовой спецификации на уровне модуля | *освоена* |
| - создавать программу по разработанному алгоритму как отдельный модуль; | *освоена* |
| ПК1.3 Выполнять отладку программных модулей с использованием специальных программных средств. | - использование инструментальных средств на этапе отладки программного продукта; | *освоена* |
| - выполнять отладку и тестирование программы на уровне модуля; | *освоена* |
| ПК1.4 Выполнять тестирование программных модулей. | - проведение тестирования программного модуля по определенному сценарию; | *освоена* |
| - выполнять отладку и тестирование программы на уровне модуля | *освоена* |
| ПК1.5 Осуществлять оптимизацию программного кода модуля. | - использование инструментальных средств на этапе отладки программного продукта; | *освоена* |
| - выполнять отладку и тестирование программы на уровне модуля. | *освоена* |
| ПК1.6 Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций. | - оформлять документацию на программные средства; | *освоена* |
| - использовать инструментальные средства для автоматизации оформления документации. | *освоена* |

Производственная практика \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ *(проставляется руководителем практики от организации)*

*(дифференцированный зачет)*

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_20\_\_\_г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

*(подпись руководителя практики (расшифровка)*

*от организации)*

М.П.

«\_\_\_\_\_\_\_» \_\_\_\_\_\_\_\_20\_\_\_г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

*(подпись руководителя практики (расшифровка)*

*от колледжа)*

**Характеристика на обучающегося**

**по освоению общих компетенций**

**в период прохождения производственной практики ПП.01**

Обучающийся(аяся) Чирков Михаил Дмитриевич программу производственной практики ПП.01 по ПМ.01 выполнил(а) в *полном/неполном* объеме;

все задания выполнил(а)

* *самостоятельно/с некоторой помощью,*
* *качественно/недобросовестно,*
* *в соответствии с установленными сроками/не в сроки.*

За время работы проявил(а) себя как

* *ответственный/безответственный,*
* *исполнительный/неисполнительный,*
* *коммуникабельный/замкнутый,*
* *доброжелательный/наглый сотрудник.*

У обучающегося были *сформированы/не сформированы* следующие общие компетенции:

|  |  |  |
| --- | --- | --- |
| **Название**  **общей (ОК) компетенции** | **Требования к умениям** | **Оценка ОК**  *(освоена/*  *не освоена)* |
| ОК 2.  Организовывать собственную деятельность, выбирать типовые методы и способы выполнения профессиональных задач, оценивать их эффективность и качество. | - выполнять отладку и тестирование программы на уровне модуля; | *освоена* |
| ОК 3.  Принимать решения в стандартных и нестандартных ситуациях и нести за них ответственность. | - создавать программу по разработанному алгоритму как отдельный модуль; | *освоена* |
| ОК 4.  Осуществлять поиск и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития. | - использовать инструментальные средства для автоматизации оформления документации; | *освоена* |
| ОК 5.  Использовать информационно-коммуникационные технологии в профессиональной деятельности. | - оформлять документацию на программные средства; | *освоена* |
| ОК 6.  Работать в коллективе и в команде, эффективно общаться с коллегами, руководством, потребителями. | - выполнять отладку и тестирование программы на уровне модуля. | *освоена* |
| ОК 7.  Брать на себя ответственность за работу членов команды (подчиненных), за результат выполненных заданий. | - осуществлять разработку кода программного модуля на современных языках программирования; | *освоена* |
| ОК 8.  Самостоятельно определять задачи профессионального и личностного развития, заниматься самообразованием, осознанно планировать повышение квалификации. | - осуществлять разработку кода программного модуля на современных языках программирования; | *освоена* |
| ОК 9.  Ориентироваться в условиях частой смены технологий в профессиональной деятельности. | - создавать программу по разработанному алгоритму как отдельный модуль; | *освоена* |

Дата «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_г.

Руководитель практики от организации

*(подпись)*

МП

**ОТЗЫВ СТУДЕНТА О ПРОХОЖДЕНИИ**

**ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ ПП.01**

Я, Чирков Михаил Дмитриевич, студент (ка) группы ПР-23 *,* проходил (а) практику в ООО НПО «Криста»*.*

Завершившая практика совпала с моими ожиданиями в том, что я получил практические навыки при выполнении заданий по практике, которые возможно пригодятся мне в будущем.

Своим главным достижением во время прохождения практики я считаю то, что я смог закрепить свои знания и применять их на практике.

Самым важным для формирования опыта практической деятельности было то, что я смог сам организовывать свой рабочий день, а также определять способы выполнения поставленных задач.

Прохождение производственной практики не повлияло на возможный выбор места работы в будущем, так как я до сих пор не определился кем хочу работать в будущем.

При выполнении выпускной квалификационной работы я хотел (а)/не хотел (а) бы получить возможность проходить преддипломную практику на данном предприятии, *так как* здесь очень дружный коллектив, который при возникновении каких либо сложностей всегда тебе поможет.

Студент (ка)

*(Ф. И.О.)*   *(подпись)*

**ЗАМЕЧАНИЯ РУКОВОДИТЕЛЯ ПРАКТИКИ ОТ КОЛЛЕДЖА**

|  |  |  |
| --- | --- | --- |
| Дата  проверки | Содержание замечаний | Должность и подпись  руководителя  практики |
| 1 | 2 | 3 |
|  |  |  |

**КАЛЕНДАРНО-ТЕМАТИЧЕСКИЙ ПЛАН ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ ПП.01**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Наименование темы | Сроки | Выполнено/ не выполнено |
| 1 | 2 | 3 | 4 |
|  | Инструктаж по технике безопасности, общий инструктаж по пожарной безопасности, а также инструктаж по правилам внутреннего распорядка и отдельнымособенностям режима работы организации. |  |  |
|  | Ознакомительная экскурсия по предприятию. Изучить следующие вопросы:   * структура предприятия; * назначение и место каждого подразделения в производственном и управленческом процессах, их взаимосвязь; * правила внутреннего трудового распорядка; * функции главных специалистов предприятия; * перспективы развития производства; * план освоения новых технологий. |  |  |
|  | Разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования; |  |  |
|  | Создание программы по разработанному алгоритму как отдельный модуль; |  |  |
|  | Разработка кода программного модуля на современных языках программирования; |  |  |
|  | Разработка кода программного продукта на основе готовой спецификации на уровне модуля; |  |  |
|  | Выполнение отладки и тестирование программы на уровне модуля; |  |  |
|  | Использование инструментальных средств на этапе отладки программного продукта; |  |  |
|  | Оформление документации на программные средства; |  |  |
|  | Использование инструментальных средств для автоматизации оформления документации |  |  |
|  | Итоговое собрание |  |  |

Руководитель практики

от организации

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Студент – практикант

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ДНЕВНИК ПРОХОЖДЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ ПП.01**

|  |  |  |
| --- | --- | --- |
| Дата | Содержание выполненной работы | Замечания и подпись руководителя от организации |
| **1** | **2** | **3** |
| **30.01** | Инструктаж по технике безопасности, общий инструктаж по пожарной безопасности, а также инструктаж по правилам внутреннего распорядка и отдельнымособенностям режима работы организации. |  |
| **31.01** | Ознакомительная экскурсия по предприятию. |  |
| **01.02** | Разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования |  |
| **02.02** | Разработка кода программного модуля на современных языках программирования |  |
| **03.02** | Создание программы по разработанному алгоритму как отдельный модуль |  |
| **04.02** | Создание программы по разработанному алгоритму как отдельный модуль |  |
| **06.02** | Разработка кода программного продукта на основе готовой спецификации на уровне модуля |  |
| **07.02** | Разработка кода программного продукта на основе готовой спецификации на уровне модуля |  |
| **08.02** | Выполнение отладки и тестирование программы на уровне модуля |  |
| **09.02** | Использование инструментальных средств на этапе отладки программного продукта |  |
| **10.02** | Оформление документации на программные средства |  |
| **11.02** | Оформление отчета |  |

М.П. Руководитель практики

от организации

**ОТЧЕТ О ПРОХОЖДЕНИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ ПП.01**

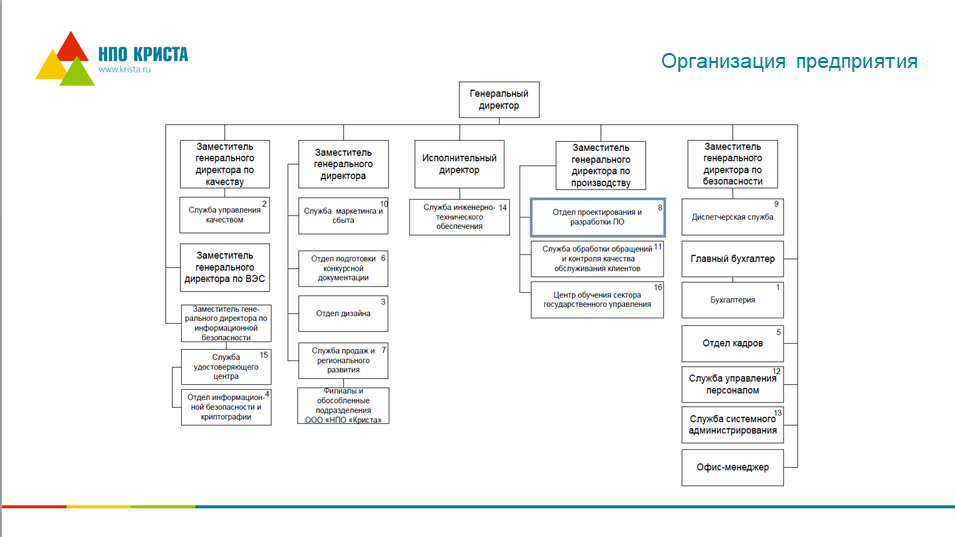
1. Инструктаж по пожарной безопасности

Работникам предприятия запрещается:

* загромождать мебелью, оборудованием и другими предметами двери, переходы и смежные секции и выходы;
* пользоваться поврежденными розетками, электрочайниками и другими электронагревательными приборами без подставки из негорючих материалов;
* оставлять без присмотра включенные в сеть электронагревательные приборы, радиоприемники и т. п.
* применять нестандартные (самодельные) нагревательные приборы;
* пользоваться электронагревательными приборами с открытой спиралью без автоматического отключения от сети при перегреве прибора;
* курение в помещениях и на территории предприятия;
* хранить в здании предприятия легковоспламеняющиеся, горючие жидкости и другие легковоспламеняющиеся материалы;
* снимать предусмотренные проектом двери коридоров, тамбуров и лестничных клеток;
* проводить огневые, сварочные и другие виды пожароопасных работ в здании предприятия при наличии в помещениях людей, а также без письменного разрешения или документа о разрешении производства таких работ;
* проводить уборку помещений с применением бензина, керосина и других легковоспламеняющихся и горючих жидкостей, а так же производить отогревание замерзших труб паяльными лампами и другими способами с применением открытого огня;
* крепление на электропроводке плакатов, схем и т.п.;
* оставление на ночь и хранение на столах стопок бумаг, папок;
* оставлять включенными компьютеры и другие электроприборы при покидании помещения;
* применение электроламп завышенной мощности;
* пользоваться переносными электролампами, шнуры которых полностью не заключены в резиновые шланги, а электролампы не защищены колпачками с укрепленной на рукоятке металлической сеткой;
* нарушать состояние электропроводки (заклеивать ее бумагой, обоями, материей, нарушать изоляцию, завешивать плакатами розетки, обертывать электролампы бумагой, тканью и другими материалами).
* закрывать рабочие кабинеты после окончания рабочего дня без предварительного осмотра помещения лицами, ответственными по приказу за пожарную безопасность;
* использовать средства пожаротушения не по назначению.

1. Ознакомительная экскурсия по предприятию

Иерархическая структура предприятия, во главе ген.директор, ему подчиняются заместители по различным направлениям(Качеству, ВЭС, по производству, по безопасности), они зачастую руководят группой отделов. Внутри отделов своя иерархия(по ответственности)

Рисунок 1 – Организация предприятия

 На этапе исполнения отдел разработки работает с АС бюджет

Рисунок 2 – Этапы бюджетного процесса

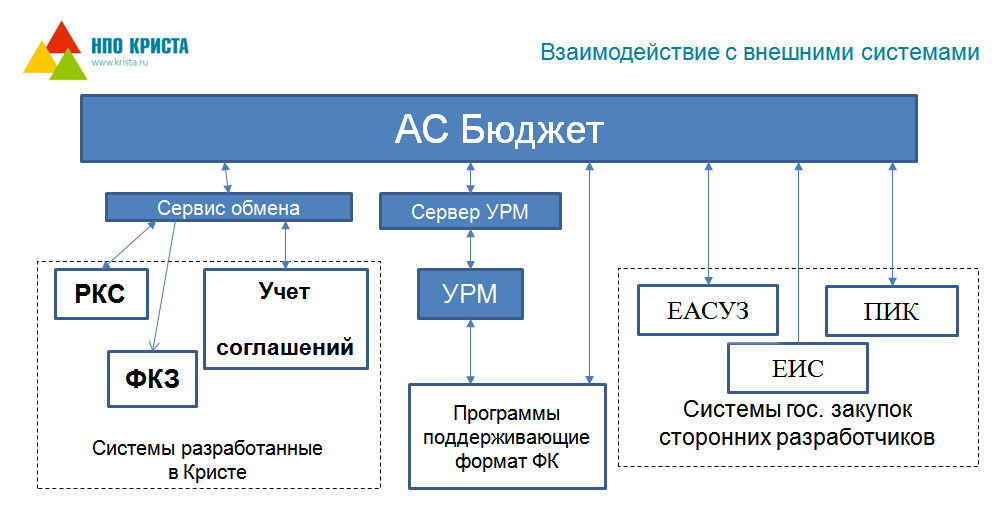
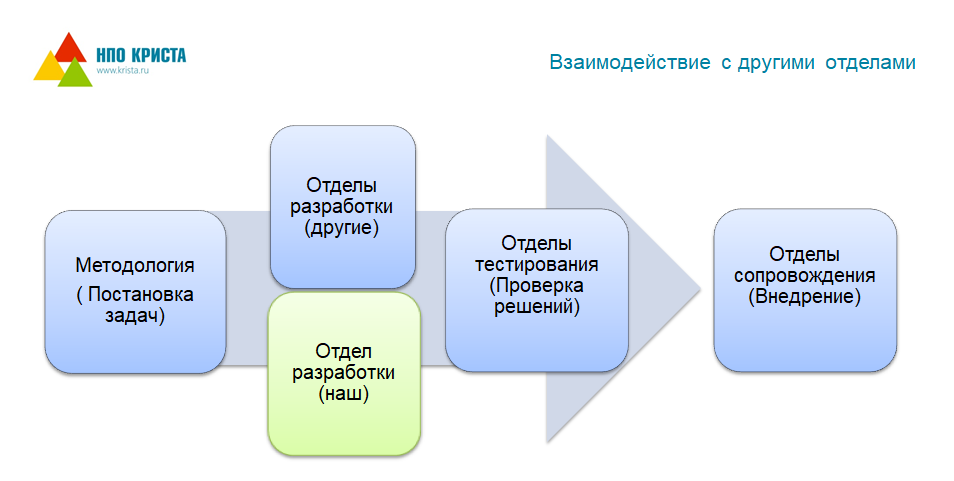
Рисунок 3 – Взаимодействие с внешними системами

Рисунок 4 – Взаимодействие с другими отделами

Правила внутреннего трудового распорядка — внутренний нормативный документ НПО «Криста». Регламентирующий в соответствии с Трудовым кодексом Российской Федерации и иными законодательными актами и нормативными документами:

* порядок приема и увольнения
* основные права,
* обязанности и ответственность сотрудников и администрации Предприятия,
* режим рабочего времени,
* время отдыха,
* применяемые к сотрудникам меры поощрения и взыскания
* иные вопросы трудовых отношений.

Функции главных специалистов предприятия – Руководитель департамента разработки:

* Планирование и координация работы отдела разработки;
* Разработка и контроль соблюдения календарных планов-графиков проектов;
* Оценка трудоемкости проектов и распределение задач на разработку среди программистов / разработчиков;
* Управление процессом разработки;
* Разработка технических заданий на модули программного обеспечения;
* Планирование и контроль исполнения бюджета отдела;
* Управление внешними ресурсами, привлекаемыми для разработки программного обеспечения;
* Разработка нормативной документации, регламентирующей работу отдела и порядок взаимодействия с функциональными подразделениями;
* Участие в разработке стратегии развития компании.

Ведущий инженер-программист-конструктор

* Делать код-ревью;
* Писать и рассматривать документацию по дизайну;
* разработка информационной модели;
* Планировать работу своих проектов;
* Делать сторонние проекты, которые приносят пользу компании;
* Быть в курсе, как проекты соотносятся с приоритетами компании.
* Согласование ТЗ

Ведущий инженер-системный программист

* Определяет необходимые системные и программные средства для разработки и отладки прикладного программного обеспечения
* Производит выбор операционной системы (ОС) и других системных компонентов,
* осуществляет подготовку задания на приобретение необходимой ОС на основе анализа задач, решаемых автоматизированной системой управления технологическими процессами (АСУТП), автоматизированной системой управления предприятием (АСУП), автоматизированной системой контроля (АСК), гибкими производственными системами (ГПС).
* Определяет возможности использования наиболее перспективных ОС на основе анализа и сравнения их аппаратно-технических характеристик.
* Осваивает средства проектирования и отладки ПО.
* Производит доработку компонентов ОС по результатам эксперимента.
* Определяет объем и содержание тестовых примеров, обеспечивающих наиболее полную проверку соответствия ОС задачам, решаемым в АСУТП, АСУП, АСК, ГПС.
* Обеспечивает модернизацию стандартных конфигураций ОС, устройств, сетей, протоколов и программ.
* Поддерживает в рабочем состоянии полный объем оперативной, накапливаемой и хранимой информации, обеспечивает защиту от несанкционированного доступа к информационным ресурсам.
* Обеспечивает работу локальной вычислительной сети.
* Разрабатывает инструкции по работе с ОС, оформляет необходимую техническую документацию.

#### Перспективы развития производства – расширение количества рабочих мест, расширять сеть своих партнеров, создание большего числа продуктов, направленных на прозрачность деятельности органов власти

План освоения новых технологий **-** инновации в беспроводных технологиях. Компания может использовать несколько систем беспроводной передачи данных, чтобы расширить возможности в плане оптимизации бизнеса. Совмещение [Wi-Fi](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:Wi-Fi" \o "Wi-Fi), [5G](https://www.tadviser.ru/index.php/5G), [Bluetooth](https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:Bluetooth" \o "Bluetooth) и высокочастотной связи (HF) создаёт возможности для инноваций и помогает повысить эффективность работы. Инжиниринг платформ. Речь идёт о согласованности инструментов управления и различных компонентов инфраструктурных технологий, таких как управление ресурсами приложений (ARM), мониторинг производительности приложений (APM), мониторинг цифрового опыта (DEM) и инструменты управления цифровой платформой (DPC). Применение таких инструментов позволит повысить гибкость операций ввода-вывода, скорость, эффективность и безопасность.

1. Разработка алгоритма поставленной задачи и реализация его средствами автоматизированного проектирования

Начало

Конец

Документ создан

Модификация документа

удаляем

Изменяем

Ввод вручную

да

нет

да

нет

Сохранение данных

Рисунок 5 – Схема алгоритма

1. Разработка кода программного модуля на современных языках программирования

По заданию мне необходимо разработать код, который позволит вносить и записывать данные в базу данных Тест. Подключение к базе данных должно происходить через среду разработки Delphi. Листинг кода, который представлен ниже написан в текстовом редакторе Блокнот:

Листинг 1 – Тестовый код для подключения к базе данных

function TaxMasterTree.cgCaptionsACCCloseUpList(Sender: TObject;

List: TStrings; Index: Integer): Boolean;

begin

inherited;

// чистим колонку (а не поле), чтобы было ClearFields

if (Index = 0) then (Sender as TGridColumn).Text := '';

Result := True;

end;

procedure TaxMasterTree.ACCEdit(Sender: TObject;

Prefix: String);

var

OrgView : IOrgView;

Rel : IRelation;

DataSet : TStaticSet;

begin

DataSet := (TGridColumn(Sender).Grid.DataSet as TStaticSet);

// Запрашиваем интерфейс справочника организаций

if (Project.QueryInterface(IOrgView, OrgView) <> S\_OK) then

raise Exception.CreateFmt(errNoItf, ['IOrgView']);

Rel := OrgView.OrgHandbook(0);

if Rel <> nil then

with Rel do begin

DataSet.Edit;

DataSet.FieldByName(Prefix + '\_REF').AsInteger := Rel.SectorByName('ACC\_REF').AsInteger;

DataSet.FieldByName(Prefix).AsString := rel.SectorByName('ACC').AsString;

end;

end;

procedure TaxMasterTree.cgCaptionsPAYER\_ACCEditButtonClick(

Sender: TObject);

begin

inherited;

ACCEdit(Sender, 'PAYER\_ACC');

end;

procedure TaxMasterTree.cgCaptionsRECEIVER\_ACCEditButtonClick(

Sender: TObject);

begin

inherited;

ACCEdit(Sender, 'RECEIVER\_ACC');

end;

end.

1. Разработка кода программного продукта на основе готовой спецификации на уровне модуля

Готовый код программы разработан в среде разработки Delphi:

Листинг 2 – Готовый код для подключения к базе данных

unit uTax;

interface

uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,

ax\_Master, QueryMan, ImplQMan, MidasQMan, ABLImpl, Menus, SldCtrls, Db,

Statics, ImgList, ComCtrls, Tabs, ExtCtrls, StdCtrls, Mask, BtnEdit,

DBCtrls, SQLObj, CoolGrid, ToolWin, FormatFields, Elements, uInterfaces,

uBudgetConsts, SectorsX, krDBUtils, Columns, krImgList, krComCtrls;

type

TaxMasterTree = class(TaxMaster)

ssCaptionID: TIntegerField;

ssCaptionDOCNUMBER: TStringField;

ssCaptionDOCDATE: TNewDateField;

ssCaptionACCEPTDATE: TNewDateField;

ssCaptionPAYER\_ACC\_REF: TAccMasterField;

ssCaptionPAYER\_ACC: TStringField;

ssCaptionRECEIVER\_ACC\_REF: TAccMasterField;

ssCaptionRECEIVER\_ACC: TStringField;

ssCaptionSUMM: TExpressionField;

ssCaptionNOTE: TStringField;

ssCaptionKCSR: TClsField;

ssCaptionKESR: TClsField;

ssCaptionKFSR: TClsField;

ssCaptionKVR: TClsField;

ssCaptionKVSR: TClsField;

ssCaptionCREATERUSER: TStringField;

ssCaptionCREATEDATE: TDateTimeField;

ssCaptionUPDATERUSER: TStringField;

ssCaptionUPDATEDATE: TDateTimeField;

cgCaptionsDOCNUMBER: TGridColumn;

cgCaptionsDOCDATE: TGridColumn;

cgCaptionsACCEPTDATE: TGridColumn;

cgCaptionsPAYER\_ACC: TGridColumn;

cgCaptionsRECEIVER\_ACC: TGridColumn;

cgCaptionsSUMM: TGridColumn;

cgCaptionsNOTE: TGridColumn;

cgCaptionsKCSR: TGridColumn;

cgCaptionsKESR: TGridColumn;

cgCaptionsKFSR: TGridColumn;

cgCaptionsKVR: TGridColumn;

cgCaptionsKVSR: TGridColumn;

cgCaptionsCREATERUSER: TGridColumn;

cgCaptionsCREATEDATE: TGridColumn;

cgCaptionsUPDATERUSER: TGridColumn;

cgCaptionsUPDATEDATE: TGridColumn;

function cgCaptionsACCCloseUpList(Sender: TObject;

List: TStrings; Index: Integer): Boolean;

procedure ACCEdit(Sender: TObject; Prefix: String);

procedure cgCaptionsRECEIVER\_ACCEditButtonClick(Sender: TObject);

procedure cgCaptionsPAYER\_ACCEditButtonClick(Sender: TObject);

private

{ Private declarations }

public

{ Public declarations }

class procedure GetAxInfo(var GUID: TGUID);override;

end;

implementation

uses uSupportBudgetTree;

const

CLASS\_TaxMasterTree : TGUID = '{E84D285F-5064-49BE-8F1D-44908CD108BB}';

{$R \*.DFM}

class procedure TaxMasterTree.GetAxInfo(var GUID: TGUID);

begin

GUID := CLASS\_TaxMasterTree;

end;

function TaxMasterTree.cgCaptionsACCCloseUpList(Sender: TObject;

List: TStrings; Index: Integer): Boolean;

begin

inherited;

// чистим колонку (а не поле), чтобы было ClearFields

if (Index = 0) then (Sender as TGridColumn).Text := '';

Result := True;

end;

procedure TaxMasterTree.ACCEdit(Sender: TObject;

Prefix: String);

var

OrgView : IOrgView;

Rel : IRelation;

DataSet : TStaticSet;

begin

DataSet := (TGridColumn(Sender).Grid.DataSet as TStaticSet);

// Запрашиваем интерфейс справочника организаций

if (Project.QueryInterface(IOrgView, OrgView) <> S\_OK) then

raise Exception.CreateFmt(errNoItf, ['IOrgView']);

Rel := OrgView.OrgHandbook(0);

if Rel <> nil then

with Rel do begin

DataSet.Edit;

DataSet.FieldByName(Prefix + '\_REF').AsInteger := Rel.SectorByName('ACC\_REF').AsInteger;

DataSet.FieldByName(Prefix).AsString := rel.SectorByName('ACC').AsString;

end;

end;

procedure TaxMasterTree.cgCaptionsPAYER\_ACCEditButtonClick(

Sender: TObject);

begin

inherited;

ACCEdit(Sender, 'PAYER\_ACC');

end;

procedure TaxMasterTree.cgCaptionsRECEIVER\_ACCEditButtonClick(

Sender: TObject);

begin

inherited;

ACCEdit(Sender, 'RECEIVER\_ACC');

end;

end.

1. Создание программы по разработанному алгоритму как отдельный модуль

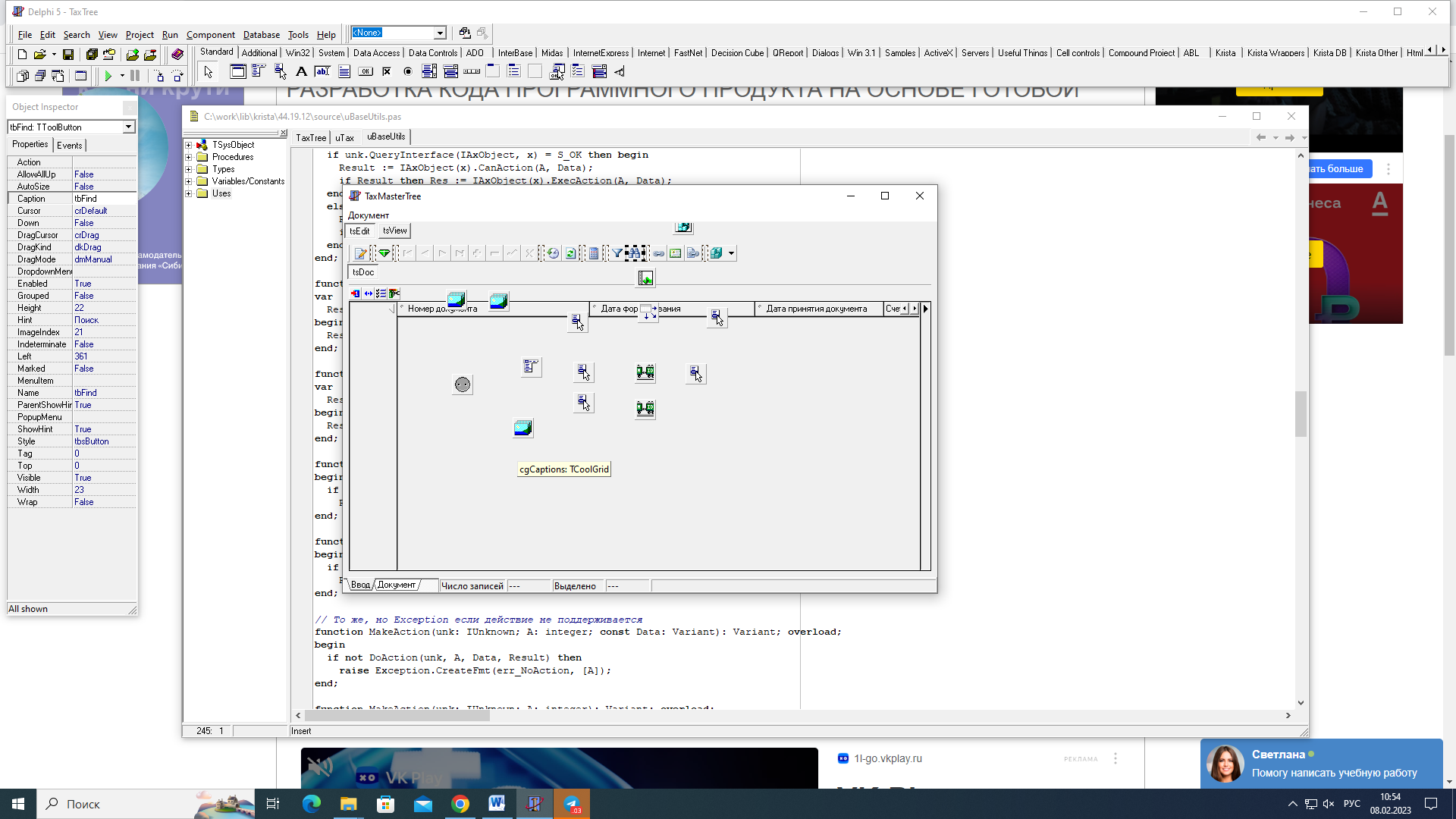


Рисунок 6 – Интерфейс приложения

Форма TaxMaster содержит:

* TDBNavigatorEx
* TCoolGrid
* 5 кнопок TGridToolButton
* TDataSource
* TMainMenu
* 5 полей TPopupMenu
* 10 кнопок TToolButton
* TFormSaver
* TStaticSet
* TImageList
* TImplementator
* TMidasQueryManager

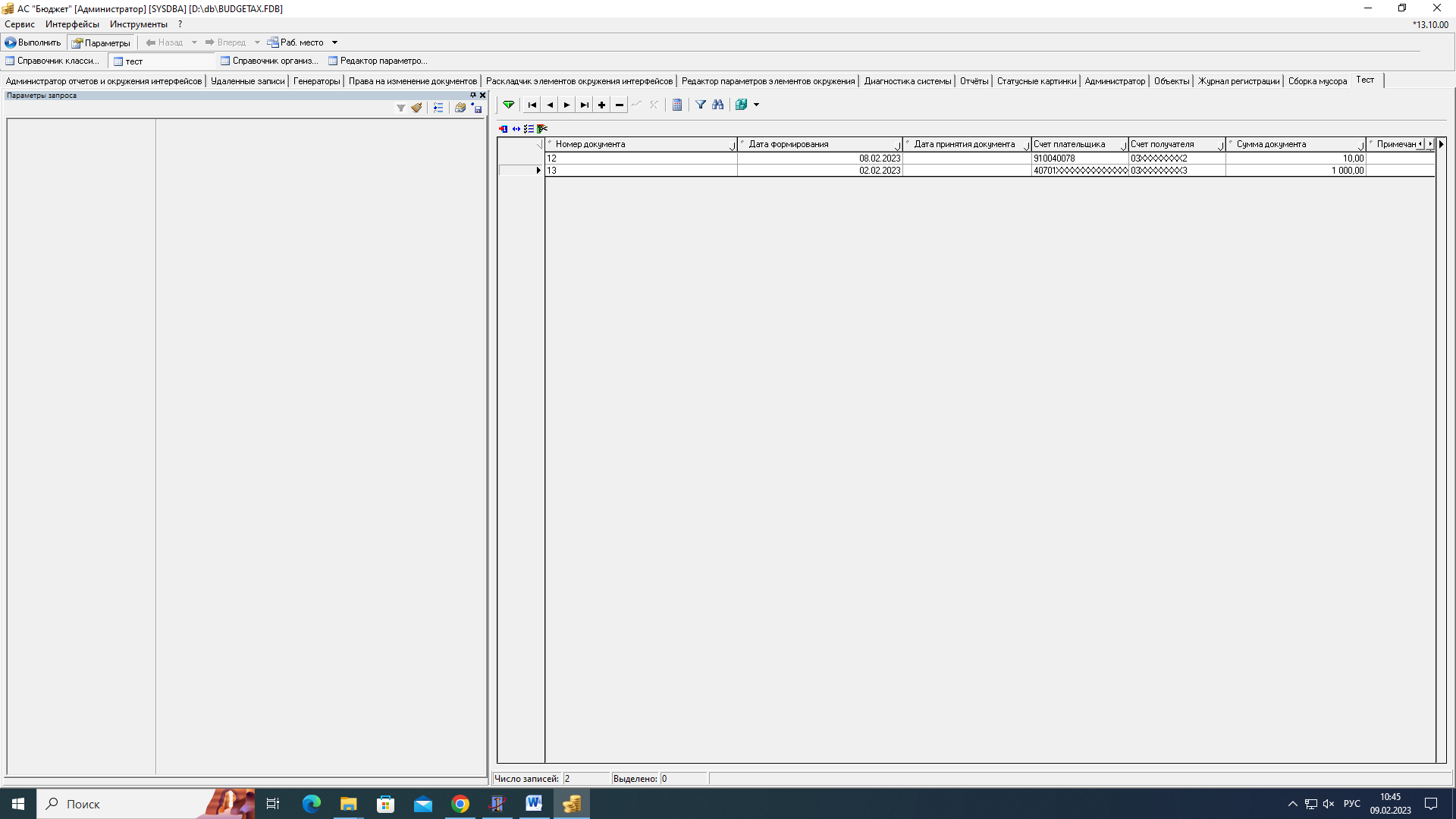


Рисунок 7 – Итоговый интерфейс программы

1. Выполнение отладки и тестирование программы на уровне модуля

Одна из самых распространенных задач отладки — выполнение программы шаг за шагом, по одной строке за раз для проверки правильности выполнения. При пошаговом прохождении кода отладчик выводит окно редактирования с выполняемой программой. Точка выполнения, показывающая следующую выполняемую строку программы, представляется в виде зеленой стрелки, расположенной слева от области исходного текста в окне редактирования.

После успешной компиляции модуля на полосе отладочной информации каждая строка кода, внесшая свой вклад в модуль, будет отмечена маленьким, синим кружком. Если же строка не помечена, значит, здесь поработал оптимизатор. Поскольку для таких строк выполняемый код не сгенерирован эти строки не будут помечены точкой выполнения.

Интегрированная среда Delphi предоставляет пользователю несколько команд пошаговой отладки доступных в меню Run

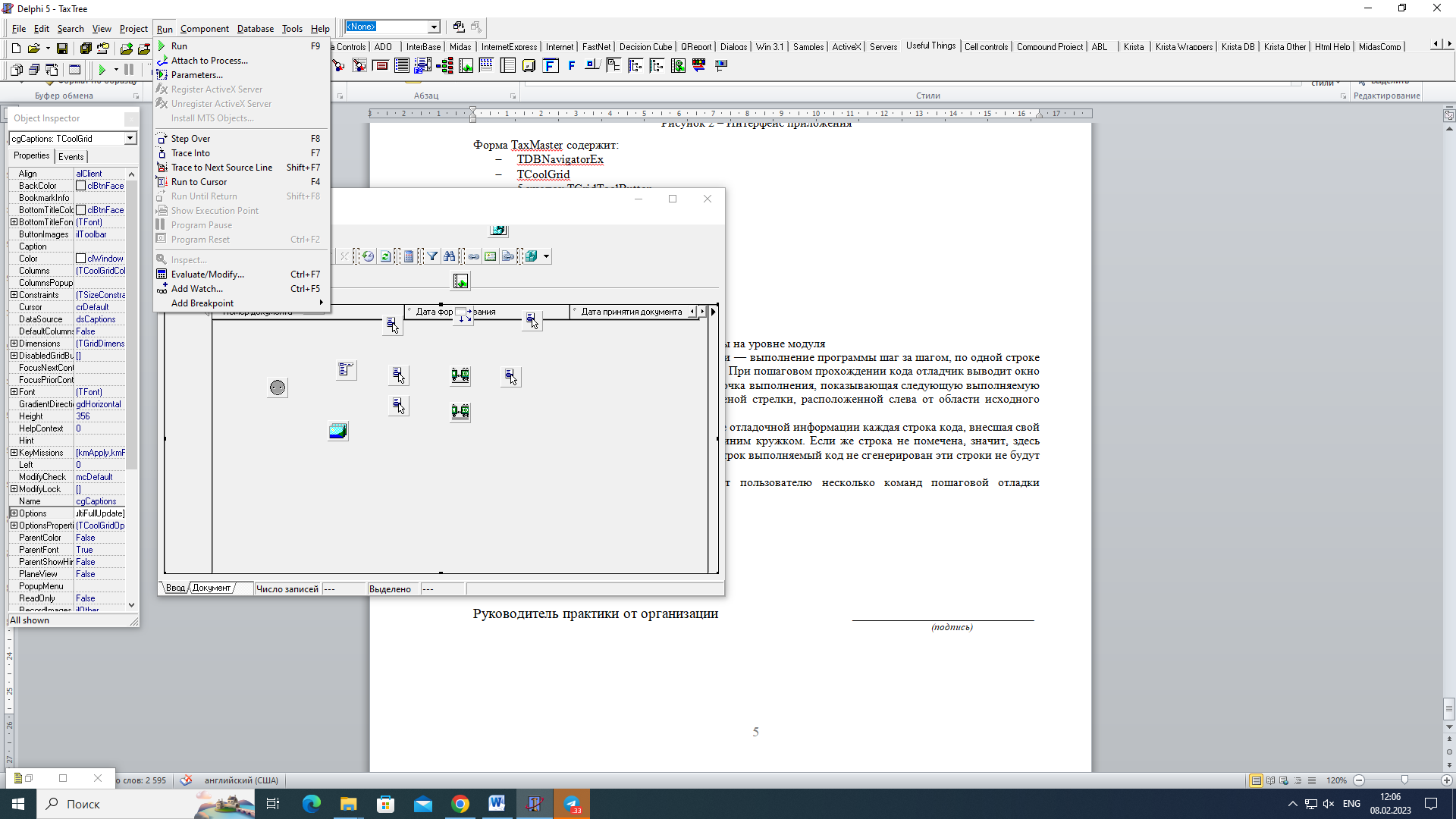


Рисунок 8 – Компоненты отладки программы

Run. Выбор этой команды запускает приложение на выполнение в обычном режиме. Вы можете использовать ее как для запуска приложения, так и для продолжения его работы после какого-либо прерывания выполнения (например, по точке останова). Если включена опция Break on Exception, используйте команду для продолжения работы после получения сообщения об исключительной ситуации

Step Over. Когда точка выполнения находится на строке содержащей вызов процедуры или функции, используйте эту команду для выполнения строки, включая вызовы в один шаг, без прохождения отдельных строк вызываемых функций. Точка выполнения перемещается при выполнении на следующую строку

Trace Into. В отличие от предыдущей команды, эта опция отработает пошаговую отладку вызываемых процедур и функций. Другими словами, если, например, в строке вызывается некая процедура, то при выполнении этой команды точка выполнения перейдет на первую строку процедуры. Однако если в строке нет таких вызовов, значит, последние две команды идентичны. Будьте осторожны при пошаговой трассировке обработчика события OnPaint. Поскольку при пошаговой отладке окно редактора размещается поверх других окон, требуется перерисовка окна приложения, для чего вызывается обработчик события OnPaint… Вы попадаете в замкнутый круг, точнее — в бесконечный цикл вызовов одного и того же обработчика. Тем не менее, стоит лишь проследить, чтобы окна приложения и редактора не перекрывались, и проблема разрешится сама собой.

Trace to Next Source Line. Иногда ваш код вызывает другой код косвенно, например, при вызове функции, которая запускает обработчик события, или при вызове функции Windows API, которая, в свою очередь, запускает функцию косвенного вызова. Поскольку такие вызовы косвенные, отладчик не видит вызова и не отслеживает пошагового выполнения таких вызовов. Однако использование описываемой команды приводит к отслеживанию таких вызовов и останову отладчика на первой строке вызываемой таким образом функции или процедуры.

I Run to Cursor. Зачастую вам вовсе не хочется в поисках ошибки, местоположение которой с какой-то точностью вам известно, пошагово добираться до нужного места через сотни, а то и тысячи строк кода. В таком случае просто поместите курсор на нужную вам строку программы в окне редактирования и используйте команду Run to Cursor. Эти действия эквивалентны временному помещению точки останова в необходимую вам строку программы, и после выполнения предшествующего строке кода работа программы приостанавливается.

Show Execution Point. Эта команда заставляет среду разработки открыть окно редактора и показать выполняемую в настоящее время строку программы. Она полезна в случаях, когда вы, например, закрыли или свернули окно редактора во время отладки (обычно при нормальном состоянии окна отладчик делает это автоматически).

Program Pause. Выбор этой команды немедленно останавливает выполнение программы. Она особенно полезна при зацикливании программы.

Program Reset. Если вы достаточно «наотлаживались» и хотите завершить работу своей программы или запустить ее заново, используйте эту команду. Она немедленно прекратит выполнение программы и вернет вас в среду разработчика. Многие команды имеют связанные с ними комбинации клавиш, например <F9 для Run. Однако назначения клавиш могут быть изменены во вкладке Editor диалогового окна Options. Например, команде Step Over назначена клавиша <F8, но при выборе назначений клавиш в стиле редактора BRIEF назначенной комбинацией клавиш становится <Ctrl+F11.

1. Использование инструментальных средств на этапе отладки программного продукта

Точка останова (breakpoint) — своеобразный знак STOP для отладчика (на полосе слева в окне редактора она и выглядит как маленький красный значок). Когда ваше приложение запущено под отладчиком и доходит до строки, в которой находится точка останова, оно прекращает работу и ждет ваших дальнейших распоряжений. Такие точки могут быть условными и безусловными. Отладчик всегда останавливается на точке безусловного останова и может останавливаться в точке условного останова, когда выполнено условие. Интегрированный отладчик Delphi поддерживает два типа условий — логическое и по количеству проходов. Ниже рассмотрены оба типа.

Установить точки останова можно следующими способами:

Поместите курсор редактирования на выбранную строку программы и нажмите клавишу команды Toggle Breakpoint (по умолчанию это клавиша <F5>) для установки или удаления точки останова в этой строке. То же самое можно выполнить и с помощью контекстного меню.

Выберите команду Run/Add Breakpoint, и откроется диалоговое окно Edit breakpoint. Для установки простейшей точки останова просто щелкните на кнопке New. Вы также можете использовать поля Filename и Line Number для установки точек останова в другом файле или строке за пределами текущей позиции курсора. Поля Condition и Pass count используются для установки точки условного останова. После установки одной или нескольких точек останова можете использовать окно Breakpoint List для управления ими. Для вызова окна Breakpoint List выберите команду View/Breakpoints. В этом окне можете щелкнуть на строке конкретной точки правой кнопкой мыши и в контекстном меню отключить точку останова с помощью команды Disable (вновь включить точку останова можно с помощью команды Enable) или удалить ее с помощью команды Delete. Команды View Source и Edit Source активизируют окно с текущим файлом исходного текста, при этом команда Edit Source устанавливает курсор в строку с точкой останова. Команда Properties выводит диалоговое окно Edit breakpoint, позволяя тем самым изменять параметры точки останова.

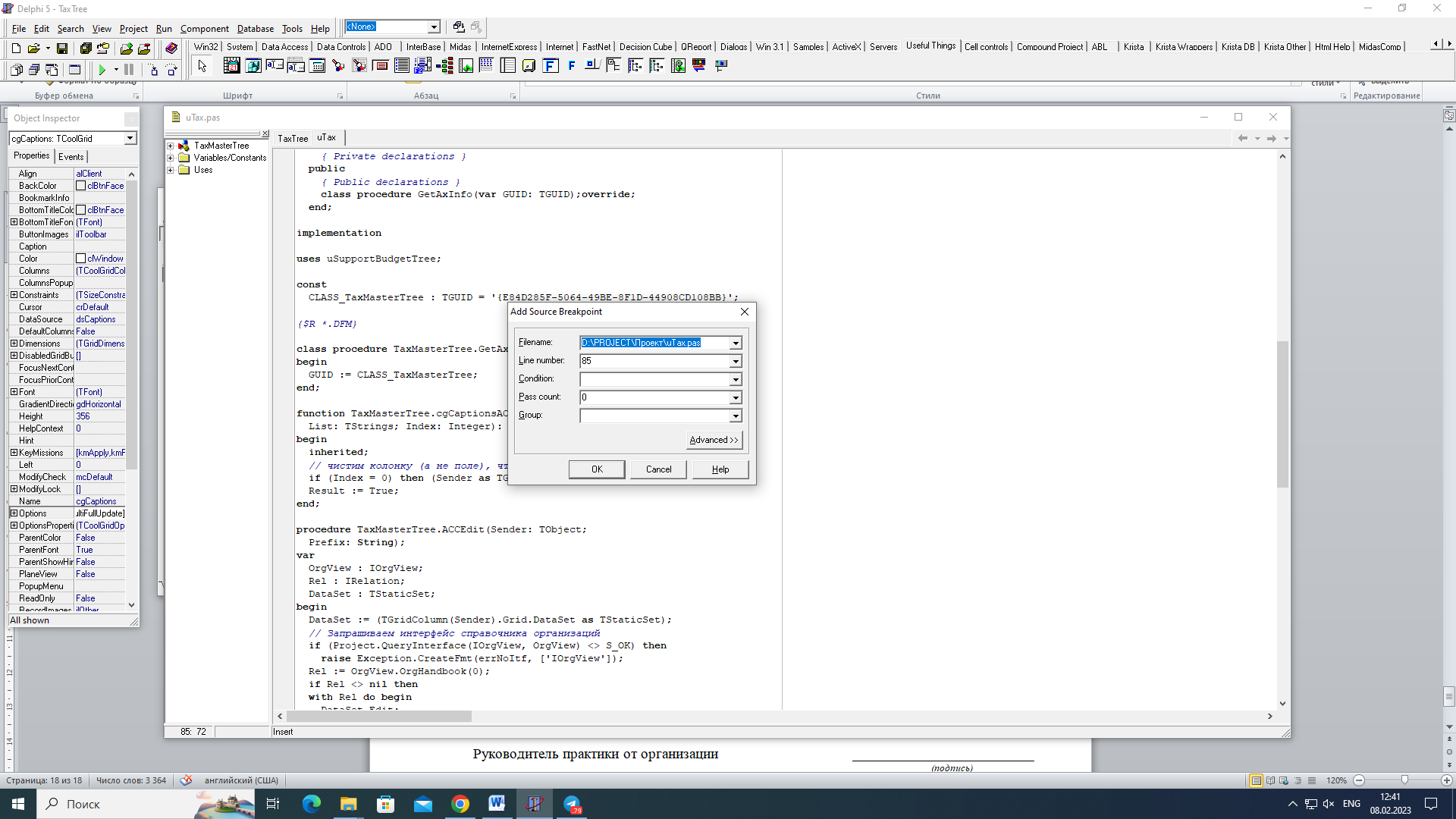


Рисунок 9 – Окно для установки точки останова

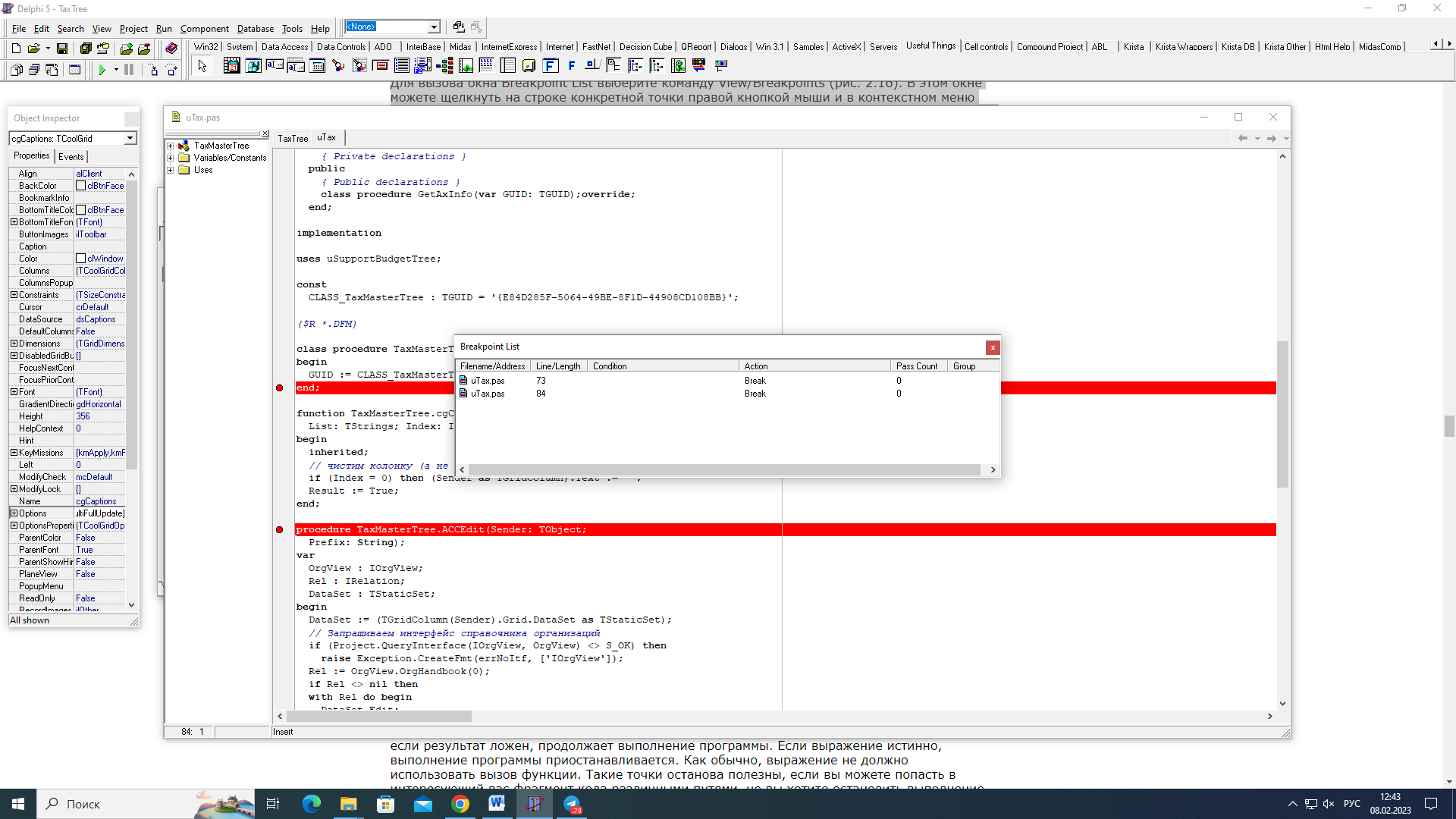


Рисунок 10 – Окно для управления точками останова

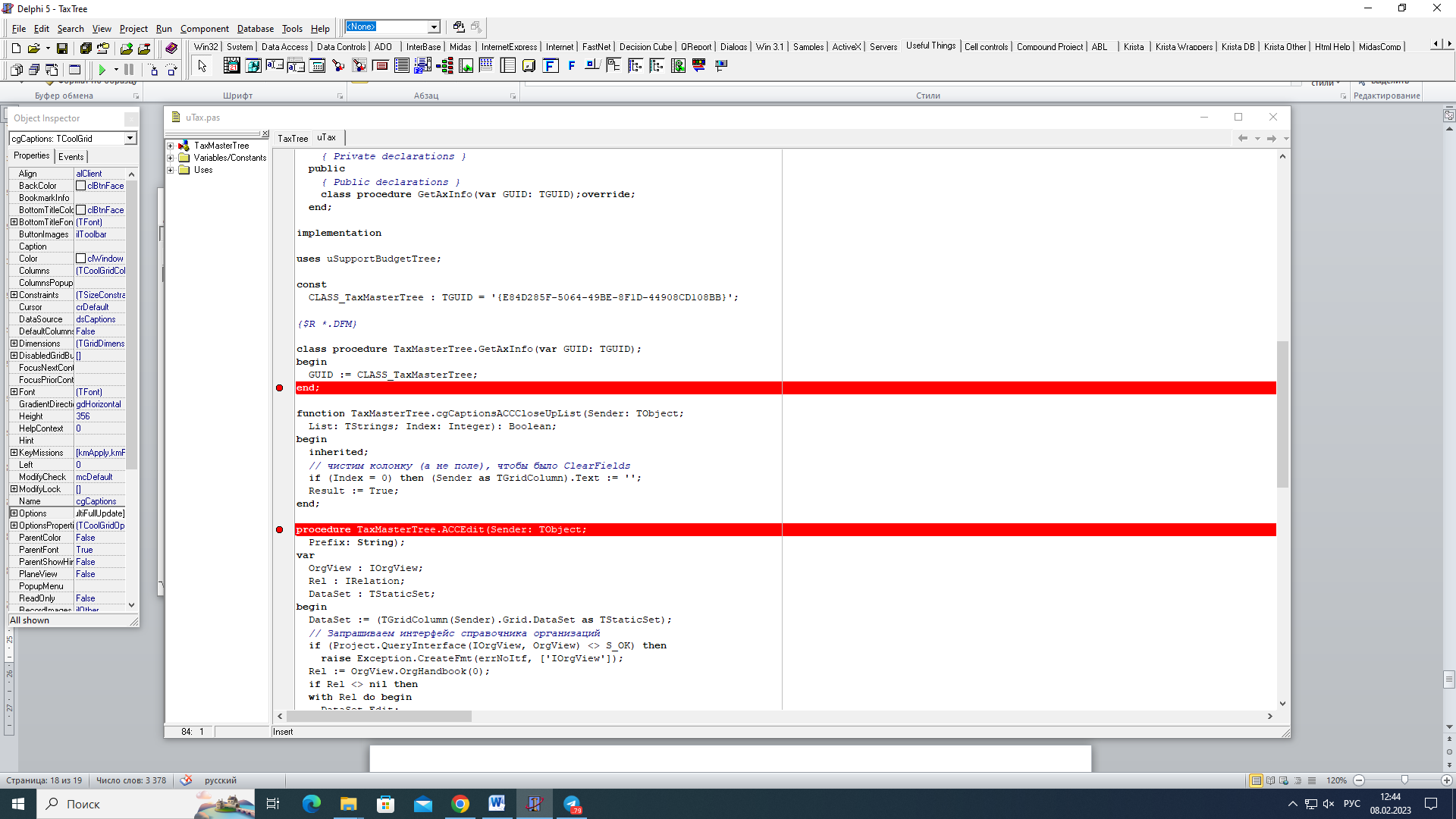


Рисунок 11 – Точки останова

После щелчка правой кнопкой мыши в окне при не выбранной точке останова выводится контекстное меню, в котором команда Add служит для добавления новой точки, Delete All удаляет все точки останова, а команды Disable All и Enable All отключают или включают все точки останова в списке.

Для превращения безусловной точки останова в условную, необходимо вызвать диалоговое окно Edit breakpoint и ввести условное выражение или количество проходов в соответствующие поля.

Условное выражение, введенное в поле Condition, может быть любым логическим выражением. По достижении точки останова отладчик вычисляет значение выражения и, если результат ложен, продолжает выполнение программы. Если выражение истинно, выполнение программы приостанавливается. Как обычно, выражение не должно использовать вызов функции. Такие точки останова полезны, если вы можете попасть в интересующий вас фрагмент кода различными путями, но вы хотите остановить выполнение программы только при достижении каких-либо конкретных условий.

Ненулевое значение, введенное в поле Pass count, дает отладчику задание продолжать выполнение программы при прохождении через точку останова, пока через нее не будет выполнено соответствующее количество проходов. При каждом проходе через точку останова отладчик уменьшает значение счетчика на единицу и по достижении нулевого значения программа приостанавливается. Такой метод полезен при работе с циклами, особенно если вы знаете, что ошибка происходит после определенного количества циклов.

1. Оформление документации на программные средства

Я составил руководство пользователя к программному продукту

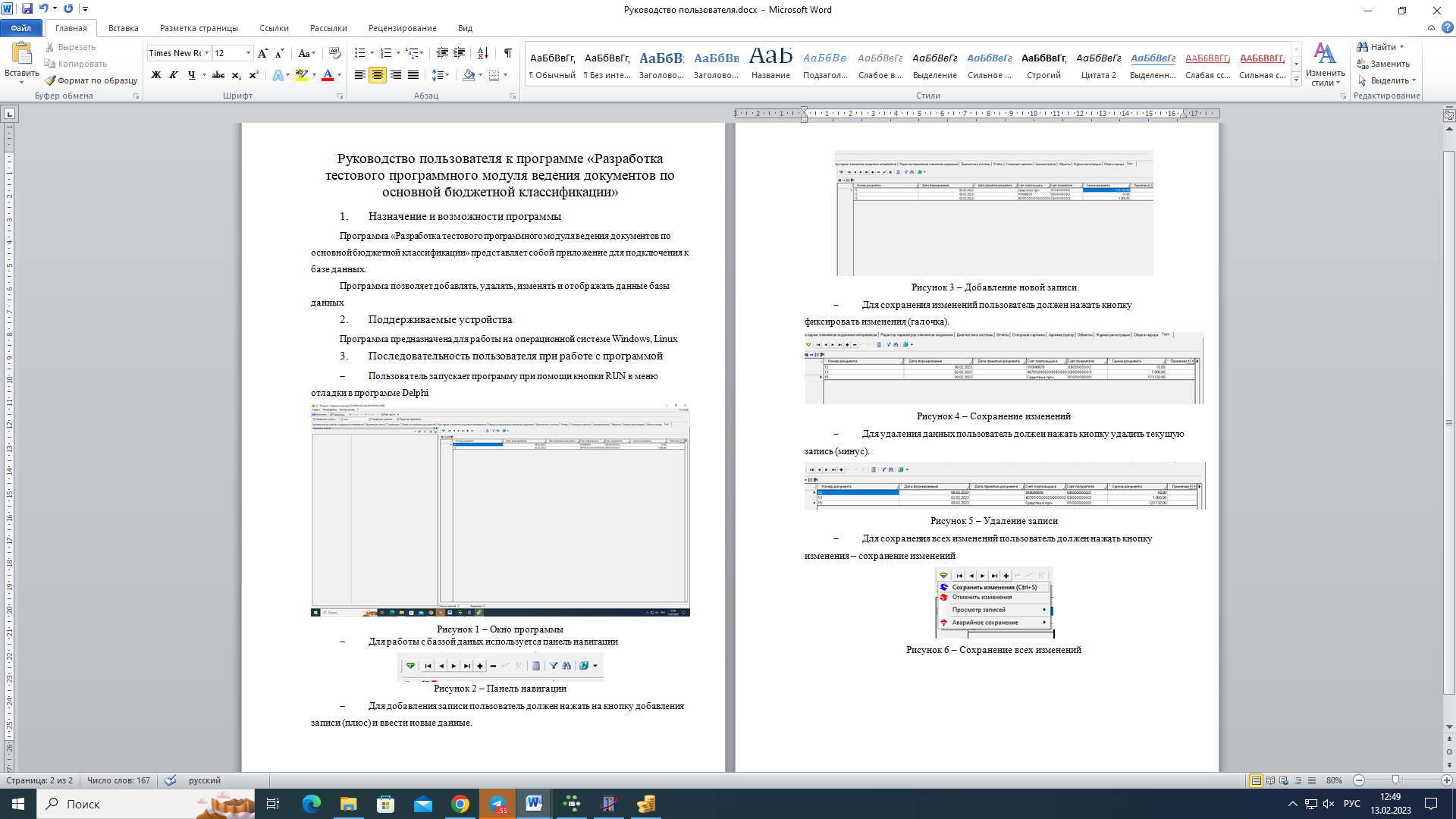


Рисунок 12 – Руководство пользователя

Руководитель практики от организации

*(подпись)*