|  |  |  |
| --- | --- | --- |
|  | ***Федеральное агентство по рыболовству***  ***Федеральное государственное бюджетное образовательное учреждение***  ***высшего образования***  ***«Астраханский государственный технический университет»***  **Система менеджмента качества в области образования, воспитания, науки и инноваций сертифицирована DQS**  **по международному стандарту ISO 9001:2015** | |
| Институт информационных технологий и коммуникаций  Направление подготовки 09.03.04 Программная инженерия  Профиль «Разработка программно-информационных систем»  Кафедра «Автоматизированные системы обработки информации и управления» | | |
| **КУРСОВОЙ ПРОЕКТ**  **“Учебно-демонстрационная программа по теме**  **“Электровакуумные лампы. Триод”**  по дисциплине «Компьютерное моделирование» | | |
| Допущен к защите  «\_\_» \_\_\_\_\_\_\_\_2020 г.  Руководитель  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Оценка, полученная на защите «\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_» | | Проект выполнен  обучающимся группы ДИПРб-21 Исмагуловым А.Г.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель  ст. преп. каф. АСОИУ Толасова В.В |
| Члены комиссии:  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |  |

**Астрахань - 2020**

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ПО РЫБОЛОВСТВУ**

**АСТРАХАНСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

|  |  |
| --- | --- |
| **УТВЕРЖДАЮ** | Кафедра «Автоматизированные системы  обработки информации и управления» |
| Заведующий кафедрой  к.т.н., доцент  С.В. Белов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

**ЗАДАНИЕ**

**на выполнение курсового проекта**

Обучающийся ***Исмагулов Азамат Гибадатович***

Группа ***ДИПРб-21***

Дисциплина ***Компьютерное моделирование***

Тема ***Учебно-демонстрационная программа по теме “Электровакуумные лампы. Триод”***

Дата получения задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.  
Срок представления обучающимся КП на кафедру «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Руководитель ***ст. преп. каф. АСОИУ*** \_\_\_\_\_\_\_\_\_***Толасова В.В.*** «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

должность, степень, званиеподписьФИО

Обучающийся \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ***Исмагулов А.Г.*** «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

подпись ФИО

**Задачи**

Разработать программу, которая

* предоставляет теоретический материал по теме “Электровакуумные лампы. Триод”;
* моделирует процесс работы электровакуумного триода;
* предоставляет возможность пройти тестирование по теме теоретического материала и решить расчётные задачи.

**Список рекомендуемой литературы**

* Калашников С.Г. Электричество: Учебное пособие – 6-е изд., стереотип. – М.: ФИЗМАТЛИТ, 2003. – 624 с.
* Шлее М. Qt 5.10. Профессиональное программирование на С++. – СПб.: БХВ – Петербург, 2018. – 1072 с.
* Лаптев В.В. С++. Объектно-ориентированное программирование: Учебное пособие. – СПб.: Питер, 2008. – 464 с.

|  |  |
| --- | --- |
| **УТВЕРЖДАЮ** | К заданию  на курсовой проект по дисциплине «Компьютерное моделирование» |
| Заведующий кафедрой  к.т.н., доцент  С.В. Белов\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г. |

**КАЛЕНДАРНЫЙ ГРАФИК**

курсового проектирования

|  |  |  |
| --- | --- | --- |
| Разделы, темы и их содержание, графический материал | Дата сдачи | Объем, % |
| Выбор темы | 24.02.2020 | 1 |
| Техническое задание | 09.03.2020 | 3 |
| Разработка модели, проектирование системы   * *введение,* * *технический проект,* * *программа и методика испытаний,* * *литература* | 13.04.2020 | 25 |
| Программная реализация системы   * *работающая программа,* * *рабочий проект* * *скорректированное техническое задание (при необходимости)* | 11.05.2020 | 40 |
| Тестирование и отладка системы, эксперименты   * *работающая программа с внесёнными изменениями,* * *окончательные тексты всех разделов* | 18.05.2020 | 50 |
| Компоновка текста  Подготовка презентации и доклада   * *пояснительная записка* * *презентация* * *электронный носитель с текстом пояснительной записки, исходным кодом проекта, презентацией и готовым программным продуктом* | 25.05.2020 | 59 |
| Защита курсового проекта | 15.06.2020 –21.06.2020 | 60-100 |

С графиком ознакомлен «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Исмагулов А.Г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, обучающийся группы ДИПРб-21

(фамилия, инициалы, подпись)

График курсового проектирования выполнен

без отклонений / с незначительными отклонениями / со значительными отклонениями

нужное подчеркнуть

Руководитель курсового проекта\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ст. преп. каф. АСОИУ Толасова В.В.

подпись, ученая степень, звание, фамилия, иниц **СОДЕРЖАНИЕ**

[Введение 5](#_Toc41237381)

[1 Технический проект 6](#_Toc41237382)

[1.1 Анализ предметной области 6](#_Toc41237383)

[1.1.1 Электронные лампы 6](#_Toc41237384)

[1.1.2 Триод 6](#_Toc41237385)

[1.1.3 Эмуляция работы триода 9](#_Toc41237386)

[1.1.4 Проверка знаний 10](#_Toc41237387)

[1.2 Технология обработки информации 11](#_Toc41237388)

[1.2.1 Форматы данных 11](#_Toc41237389)

[1.2.2 Форматы хранения данных ресурсов 14](#_Toc41237390)

[1.2.3 Алгоритм вывода теоретической информации 15](#_Toc41237393)

[1.2.4 Алгоритм определения режима работы триода 15](#_Toc41237394)

[1.2.5 Алгоритм составления вопросов теста 16](#_Toc41237395)

[1.2.6 Алгоритм оценивания результатов тестирования 16](#_Toc41237395)

[1.2.7 Алгоритм демонстрации работы триода 17](#_Toc41237395)

[1.3 Входные и выходные данные 18](#_Toc41237396)

[1.4 Системные требования 18](#_Toc41237397)

# ВВЕДЕНИЕ

Физическая наука, несмотря на всю свою сложность, вызывает огромный интерес у многих изучающих её студентов. Её привлекательность – в предмете изучения – окружающем мире, процесс познания которого есть теоретизация наблюдений. Однако для получения более глубокого понимания физики важно не только обладать обширными теоретическими знаниями, но и иметь практическое представление того, как протекают физические процессы. Вопрос наглядности познаваемого предмета является центральным для преподавателей физических дисциплин. По замыслам педагогов, лабораторные опыты, лежащие в основе практической части курса физики, должны обеспечить наглядность изучаемого материала. Однако из-за недостаточной оснащённости учебных заведений проводить все запланированные программой опыты невозможно. Восполнить пробелы образовательного процесса может помочь компьютерная симуляция физических явлений.

Подобная автоматизация учебной деятельности позволяет не просто заменить лабораторное оборудование, но и привнести большую наглядность предмету изучения. Цифровые технологии отрисовки придают детальность рассматриваемым явлениям, к примеру, незаметное глазу движение частиц, может быть явно продемонстрировано с помощью элементов графической анимации.

Особым образом должно быть построено обучение на прикладных специальностях, связанных с радиотехникой или управлением вычислительных машин. Постоянное взаимодействие с лабораторными приборами требует от специалистов знания мельчайших элементов их устройства и понимания процессов взаимодействия составных частей. Операторам ЭВМ, будущим архитекторам компьютерных устройств важно знать об истоках современной вычислительной техники – электровакуумных лампах использовавшихся в процессорной части первых компьютеров. Наглядно объяснить физический смысл использования триодов и диодов в вычислительных системах без использования оборудования на аудиторных занятиях не представляется возможным. На помощь преподавателю могут прийти цифровые образовательные инструменты – учебно-демонстрационные программы, разработанные специально для повышения качества представления материала и улучшения его репрезентативности.

Цель разработки учебно-демонстрационной программы “Триод”– автоматизация процесса обучения студентов по теме “Электровакуумные лампы”.

Назначение – снижение нагрузки на преподавателя, повышение качества знаний студентов, изучающих физику.

.

## 1 ТЕХНИЧЕСКИЙ ПРОЕКТ

## 1.1 Анализ предметной области

### 1.1.1 Электронные лампы

**Электровакуумный прибор –** устройство, предназначенное для генерации, усиления и преобразования [электромагнитной энергии](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BC%D0%B0%D0%B3%D0%BD%D0%B8%D1%82%D0%BD%D0%B0%D1%8F_%D1%8D%D0%BD%D0%B5%D1%80%D0%B3%D0%B8%D1%8F), в котором рабочее пространство освобождено от [воздуха](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D0%B7%D0%B4%D1%83%D1%85) и защищено от окружающей [атмосферы](https://ru.wikipedia.org/wiki/%D0%90%D1%82%D0%BC%D0%BE%D1%81%D1%84%D0%B5%D1%80%D0%B0_%D0%97%D0%B5%D0%BC%D0%BB%D0%B8) непроницаемой оболочкой.

**Электронная лампа –** электровакуумный прибор, работающий за счёт управления интенсивностью потока [электронов](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD), движущихся в вакууме или разрежённом газе между [электродами](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%B4). Электронные лампы имеют 2 и более электродов – анод, катод и сетку.

**Катод** – электрод, из которого “вытекает” поток заряженных частиц, полученный в результате термоэлектронной эмиссии.

**Анод** – электрод, принимающий поток заряженных частиц, выпущенный катодом.

**Сетка** – электрод, находящийся в потоке электронов между анодом и катодом и не перекрывающий его полностью. Внутри электровакуумного прибора сетка выполняет роль регулятора потока в соответствии с поданным на неё напряжением.

### 1.1.2 Триод

**1.1.2.1 Принцип работы триода**

**Триод** – электронная лампа состоящая из трёх электродов (анода, катода и сетки), которая может управлять током цепи с помощью небольших напряжений, подаваемых на управляющую сетку.

На рисунке 1.1. показано строение триода и его схематическое обозначение.

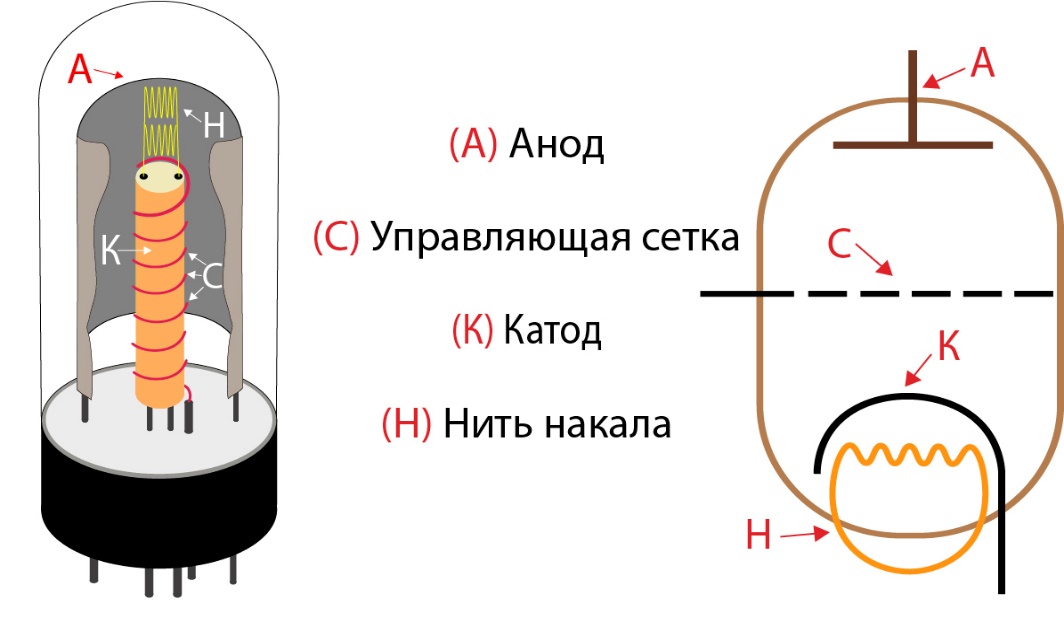


Рисунок 1.1 – Строение триода

Когда триод проводит ток, электроны, двигаясь от катода к аноду, вынуждены проходить через отверстия в управляющей сетке. Посредством подачи небольшого отрицательного напряжения на управляющую сетку, вокруг сетки образуется электронное облако, отталкивающее поток заряженных частиц, испускаемый катодом. В зависимости от величины подачи потенциала на сетке можно контролировать количество электронов, проходящих через электронное облако. Когда управляющая сетка блокирует протекание тока, говорят, что лампа “заперта”.

Усилить электрический сигнал, проходящий через триод можно, подав на ножку триода положительное напряжение. Тогда вокруг сетки образуется облако, проталкивающее заряженные частицы к отрицательному заряду на аноде. Ток, при котором все электроны, выпущенные катодом достигают анода называют **током насыщения**. При высоком входном напряжении, когда ток насыщения уже был достигнут, электроны могут скапливаться на сетке, создавая сеточный ток. В такой ситуации тока говорят, что лампа “отперта”

Электрический ток, при котором достигается наибольший производственный эффект, называется “током рабочей зоны” (или просто “рабочим током”).

На рисунке 1.2 продемонстрированы режимы работы триода при различных напряжениях, подаваемых на управляющую сетку.

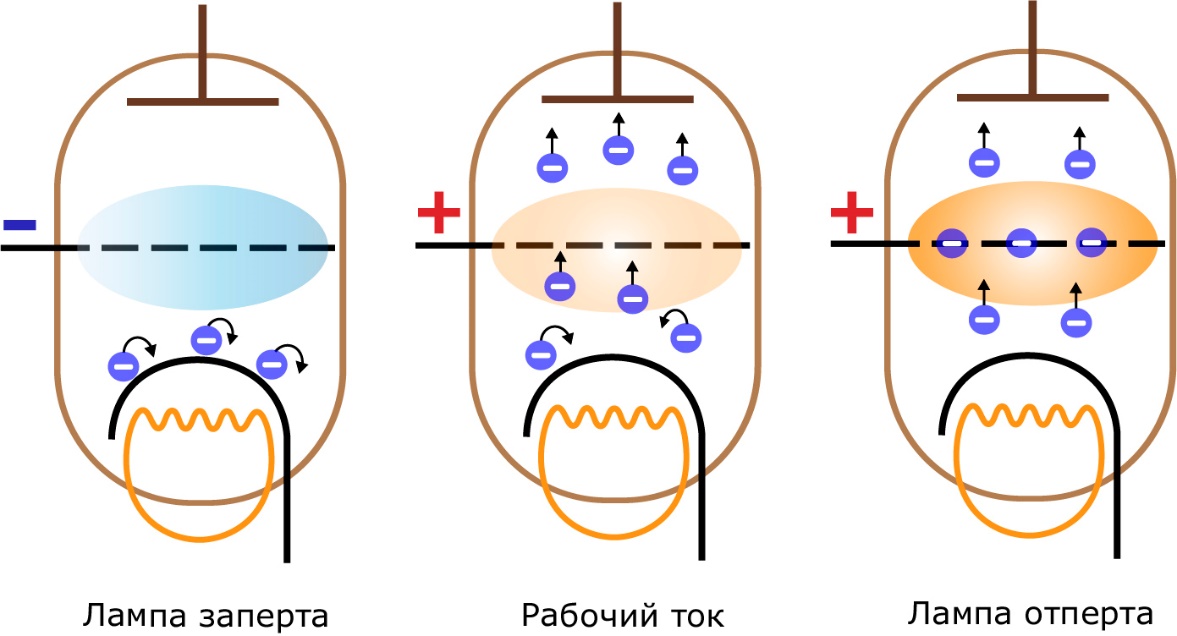


Рисунок 1.2 – Режимы работы триода

**1.1.2.2 Характеристики триода**

Чтобы измерить работу усилителя триода, необходима установка, состоящая из:

* источника анодного напряжения Ua;
* источника малого напряжения на сетке с двухполярным подключением Uc;
* миллиамперметра – для измерения силы анодного тока Ia;
* вольтметра – для измерения сеточного напряжения;

На рисунке 1.3 продемонстрирована схема электрической цепи для измерения физических характеристик триода.

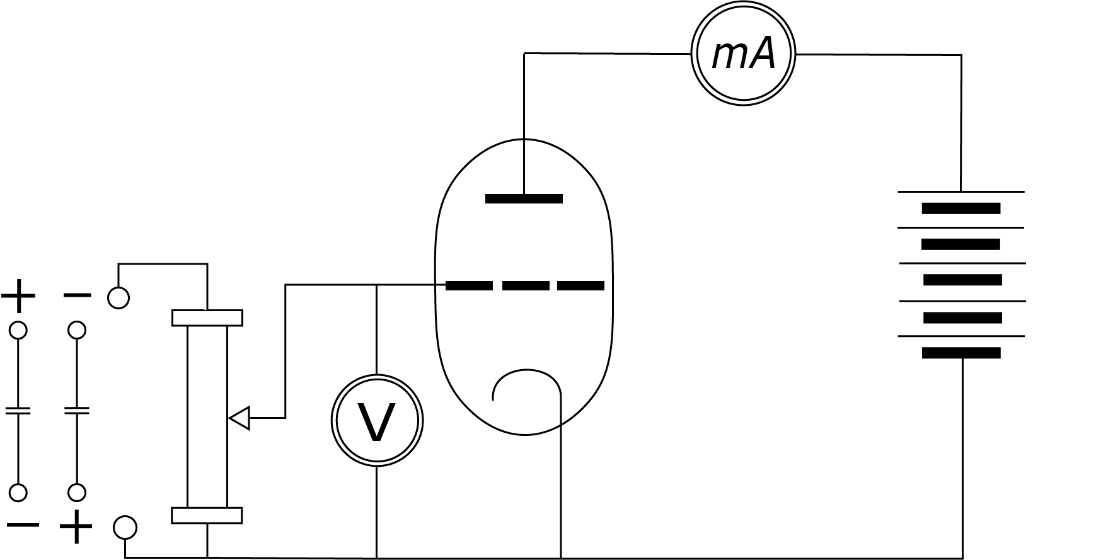


Рисунок 1.3 – Схема установки для измерения физических характеристик триода

Важнейшей характеристикой триода является анодно-сеточная. Она представляет собой график зависимости анодного тока от напряжения на сетке при неизменном напряжении на аноде лампы. На рисунке 1.4 представлен график анодно-сеточной характеристики.



Рисунок 1.4 – График анодно-сеточной характеристики

По вертикали отложена сила анодного тока при различных напряжениях на сетке, причем анодное напряжение поддерживается постоянным. С изменением сеточного напряжения от отрицательного значения до нуля сила анодного тока возрастает до определенной величины. Вместе с тем, чем выше напряжение на аноде, тем больше сила анодного тока при данном напряжении на сетке.

К основным параметрам триода относятся: крутизна лампы, внутреннее сопротивление и коэффициент усиления.

**Крутизна** **лампы** S показывает, на сколько миллиампер изменяется сила анодного тока при изменении напряжения на сетке на 1 Вольт и постоянном анодном напряжении и определяется следующей формулой:

|  |  |
| --- | --- |
|  | (1) |

где *ΔIа* – изменение силы анодного тока, мА;

*ΔUc* – изменение напряжения на сетке, В.

Близко расположенная к катоду сетка воздействует на электроны гораздо сильнее, чем далеко расположенный анод. Поэтому изменить анодный ток на некоторую определенную величину можно либо соответствующим изменением анодного напряжения, либо во много раз меньшим изменением напряжения на сетке.

**Коэффициент усиления лампы** μ определяется отношением изменения анодного напряжения к изменению напряжения на сетке при постоянном анодном токе:

|  |  |
| --- | --- |
|  | (2) |

где *ΔUа* – изменение напряжения анодного тока, В;

*ΔUc* – изменение напряжения на сетке, В.

**Внутреннее сопротивление лампы** Ri показывает, на сколько вольт надо изменить напряжение на аноде при постоянном напряжении на сетке, чтобы анодный ток изменился на один ампер. Чем больше внутреннее сопротивление, тем меньше наклон анодной характеристики триода. Внутреннее сопротивление определяется формулой:

|  |  |
| --- | --- |
|  | (3) |

где *ΔUа* – изменение напряжения анодного тока, В;

*ΔIа* – изменение силы анодного тока, мА.

### 1.1.3 Эмуляция работы триода

Для понимания принципы работы триода в качестве усилителя электрического сигнала, необходимо продемонстрировать зависимость силы анодного тока от величины напряжения на сетке. Поэтому студенту следует дать возможность самому изменять сеточное напряжение. В схеме, продемонстрированной на рисунке 1.3, потенциал на сетку подаётся с помощью постоянного источника малых напряжений. Изменение потенциала производится за счёт изменения сопротивления на сетке Rс, для чего между источником тока и ножкой триода размещён реостат. Исходя из конструкции схемы, расчёт коэффициента усиления лампы лучше производить, используя величину Rc и крутизну лампы S:

|  |  |
| --- | --- |
|  | (4) |
| где *Rc* – сопротивление на сетке, В;  *S* – изменение напряжения на сетке, .  Таким образом, основными элементами управления должны стать слайдеры изменения сопротивления на сетке и напряжения тока в цепи. Изменение соответствующих величин должно отражаться на значениях количественных характеристик лампы – *S, , Ri,* и параметрах измеряемых вольтметром и амперметром – *Uc,**Ia*. | |

Чтобы продемонстрировать как отпирающее, так и запирающее действие триода, необходимо предоставить студенту выбор полярности подключения источника напряжения на сетке: “плюс-минус” – для создании положительного сеточного потенциала или “минус-плюс” – для отрицательного. Наглядность процессу работы триода может придать увеличенное изображение анодного тока в виде поточного движения заряженных частиц (см. рис. 1.2).

### 1.1.4 Проверка знаний

Раздел тестирования разработан для проверки знаний обучающегося по материалу, описанному в теоретическом блоке. На все вопросы теста можно дать верные ответы, внимательно изучив теоретический раздел.

Пользователю предоставлена возможность самостоятельного выбора количества вопросов теста. Однако тест не может содержать менее 5 вопросов. Максимальное количество вопросов в тесте - 10. Для объективной оценки знаний в тест включены 2 вида вопросов: с вариантами ответов и без них (открытый вопрос). В вопросах закрытого типа пользователю предлагаются 3 варианта ответа, из которых правильный только один, ответы на открытые вопросы вводятся пользователем в поле ввода.

Для переходов между вопросами пользователю необходимо использовать панель управления: предусмотрен возврат к предыдущим вопросам, пропуск вопросов и досрочное выполнение тестирования.

После выполнения теста на экран выводится оценка по пятибалльной шкале. 5 ставится за правильные ответы на 80 % вопросов, 4 - за 60%, 3 - за 50%, 2 - если были получены верные ответы менее чем на половину вопросов.

Все вопросы, на которые пользователем были даны неверные ответы, выводятся после проверки в завершающем окне тестирования.

## 1.2 Технология обработки информации

Анализ предметной области показал, что программа предусмотрена для одного пользователя. На рисунке 1.5 представлена диаграмма вариантов использования.



Рисунок 1.5 – Диаграмма вариантов использования

### 1.2.1 Форматы данных

Для удобства проектирования учебно-демонстрационной программы, триод, как материальный объект, следует разделить на две программные сущности: **ПарамТриод** – для хранения физических параметров триода и инкапсуляции функций расчёта анодно-сеточной характеристики и **РежимЛампы –** для демонстрации внутренних физических процессов, протекающих внутри триода.

Типданных **ПарамТриод** должен содержать следующие поля, отражающие физические характеристики лампы и представленные в виде целочисленных переменных:

* сопротивление на сетке;
* напряжение на сетке;
* крутизна лампы;
* коэффициент усиления лампы;
* внутреннее сопротивление лампы;

Для получения значений последних трёх полей необходимо реализовать соответствующие функции расчёта:

* расчёт крутизны лампы, возвращаемое значение – целое число;
* расчёт коэффициента усиления , возвращаемое значение – целое число;
* расчёт внутреннего сопротивления, возвращаемое значение – целое число;

Приведённое описание типа данных **ПарамТриод** отражено в диаграмме классов на рисунке 1.6.

Тип данных **РежимЛампы** должен хранить формат отображения триода при различных физических параметрах, его полями являются:

* цветная эллиптическая фигура, отражающая электронное облако вокруг сетки;
* фигура знака – плюс или минус, отражающий знак подаваемого на сетку напряжения;
* эллиптическая фигура, отражающая частицу электрического потока;

Для наглядной демонстрации поведения частиц и их взаимодействия с электронным облаком, возникающим вокруг сетки, при различных параметрах триода следует реализовать функции:

* движения частиц к аноду;
* вибрации частиц вокруг катода;
* смены знака потенциала;
* изменения цвета электронного облака.

Конкретный режим работы лампы характеризует определённое сочетание графических функций, поэтому тип данных **РежимЛампы** должен содержать методы, возвращающие лампу в одном из трёх рабочих положений:

* запирающее положение – облако синего оттенка, частицы вибрируют около катода;
* отпирающее положение – облако оранжевого оттенка, частицы стремятся к аноду, небольшое количество частиц скапливается на сетке;
* рабочее положение – облако светло-оранжевого цвета, частицы стремятся к аноду, некоторые возвращаются к катоду.

Описание типа данных **РежимЛампы** отражено в диаграмме классов на рисунке 1.6.

Хранить информацию о характеристиках тока цепи следует внутри объекта типа **ЭлектроЦепь**. Полями данного типа должны быть переменные, сохраняющие:

* напряжение цепи;
* силу тока цепи.

Установка и получение значений производятся с помощью соответствующих методов.

Приведённое описание типа данных **ЭлектроЦепь** показано на диаграмме классов на рисунке 1.6.

Для эмуляции физических процессов внутри триода будет создана графическая форма, на которой следует разместить все элементы управления и анимированные объекты. Класс **Демонстрация** инкапсулирует описанные данные исодержит следующие поля:

* форма пользовательского интерфейса;
* объект Триод типа данных **ПарамТриод**;
* объект Режим типа данных **РежимЛампы**;
* объект Цепь типа данных **ЭлектроЦепь**.

Чтобы взять входные данные и продемонстрировать результаты пользовательских действий на экране, требуется реализовать методы:

* выбора полярности подключения триода;
* изменения сопротивления на сетке;
* изменения напряжения в цепи;
* вывода пояснений к разделу;
* отрисовки графика;
* отрисовки лампы в нужном режиме.

Описание класса **Демонстрация** отражено на диаграмме классов на рисунке 1.6.

Подобно режиму эмуляции работы триода, для прочтения теоретического материала и прохождения тестирования, следует создать графические формы пользовательского интерфейса с необходимыми элементами управления. С целью инкапсуляции данных необходимо создать классы **Теория** и **Экзамен** и реализовать в них функции для работы с соответствующими графическими формами.

Класс **Экзамен** для обработки данных тестирования должен содержать функции:

* генерации вопросов;
* вывода вопросов на экран;
* проверки ответов пользователя;
* вывода результатов тестирования.

Описанные классы **Экзамен** и **Теория** показаны на диаграмме классов на рисунке 1.6.

Для упрощения ориентации в программе необходимо создать главное меню – диалоговое окно с элементами управления для перехода между разделами. Форму диалогового интерфейса вместе с сигнальными функциями нужно поместить в отдельный класс **Главное окно**.

На рисунке 1.6 представлена диаграмма классов разрабатываемой программы.

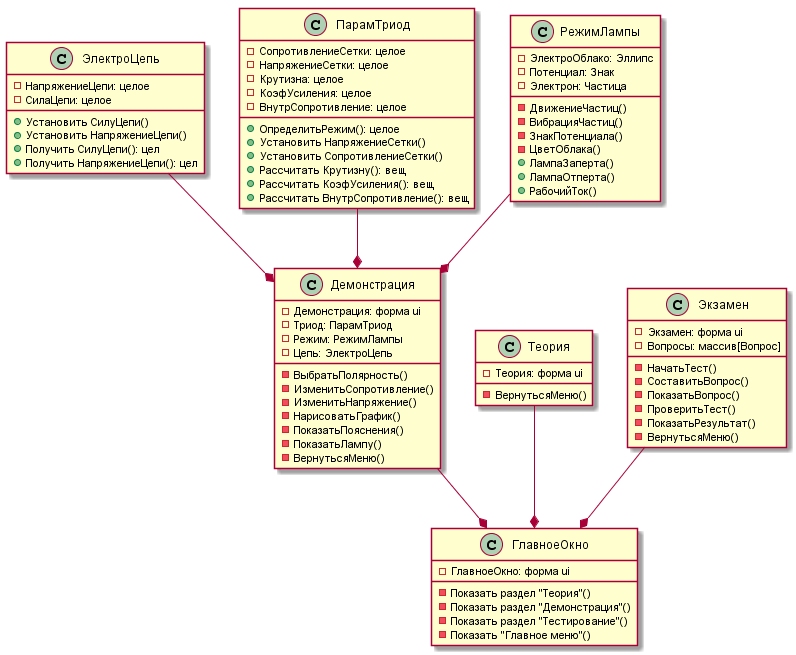


Рисунок 1.6 – Диаграмма классов

### 1.2.2 Форматы хранения данных ресурсов

### 1.2.2.1 Формат хранения данных ресурсов для теоретического материала

Информация теоретического раздела, выводимая в пользовательском окне, хранится в текстовом файле “theory\_text.txt”. Файл содержит текст теории, места хранения изображений и их названия. Файл полностью составлен из html-тегов, выводимых программой в удобной для чтения форме. Файл обрабатывается встроенным в программу текстовым браузером, который обрабатывает html-теги и выводит текст в форматированном виде.

Для доступа к изображениям в тексте теории прописывается их места хранения, названия, форматы: <img src="add/image2.png" alt="" />. Для обеспечения корректного вывода в начале файла указывается наименование кодировки: <charset=utf-8/>.

### 1.2.2.2 Формат хранения данных ресурсов для тестирования

Все вопросы теста находятся в одном текстовом файле. Вопросы выделены символом “?”, тип вопроса - открытый или закрытый указывается следующим символом: “1” - вопрос закрытого типа, “2” - вопрос открытого типа. Далее следует текст вопроса.

После формулировки вопроса закрытого типа следуют 3 варианта ответа. Варианты ответов и условие вопроса располагаются на отдельных строках. Первый вариант, представленный после вопроса, всегда является правильным.

Ответом на вопрос открытого типа является число, в файле ответ на хранится на следующей строки после текста вопроса.

* Структура вопроса закрытого типа :

*?1Текст вопроса*

*Вариант ответа 1 (ответ на вопрос)*

*Вариант ответа 2*

*Вариант ответа 3*

* Структура вопроса открытого типа :

*?2Текст вопроса*

*Ответ на вопрос.*

### 1.2.3 Алгоритм вывода теоретической информации

**Начало**

| создать *файловый поток*;

| считать информацию из *файла* с теоретическим материалом;

| создать элемент управления *ТекстовыйБраузер*;

| прикрепить *ТекстовыйБраузер* к диалоговой форме режима Теория;

| вывести в *ТекстовыйБраузер* содержание файлового потока;

**Конец**

### 1.2.4 Алгоритм определения режима работы триода

**Начало**

| определить *тип подключения* триода;

| считать *сопротивление* на сетке;

| провести расчёт *напряжения* на сетке

| **если** ( |*напряжение*| ≥ *минимального*)

| | **если** (*тип подключения* = *плюс*)

| | | **если** (*напряжения* ≥ *максимальный*)

| | | | вернуть “лампа отперта”;

| | | **конец**;

| | **иначе**

| | | **если** (*напряжения* ≥ *максимальный*)

| | | | вернуть “лампа заперта”;

| | **конец**;

| | вернуть “рабочий ток”;

| **конец**;

**Конец;**

### 1.2.5 Алгоритм составления вопросов теста

**Начало**

| считать в поток *stream* текстовый файл;

| **пока** (не конец файла)

| | занести в *строку* из файла - формулировка вопроса;

| | определить *тип вопроса*;

| | **выбрать** (от *тип**вопроса*)

| | | **случай** (*закрытый**вопрос***)**:

| | | | присвоить *первому**варианту**ответа* статус верного;

| | | | считать *три**следующие**строки* – варианты ответов;

| | | | занести содержимое предыдущих четырёх строк в экземпляр *структуры**вопрос*;

| | | **завершить**;

| | | **случай** **(***открытый* *вопрос*)

| | | | считать строку из потока – *ответ* *на* *вопрос*;

| | | | занести ответ в экземпляр *структуры**ответ*;

| | | **завершить**;

| | **конец**;

| **конец**;

| занести экземпляр *структуры* *вопрос* в *массив* *вопросов*;

**Конец**;

#### 1.2.6 Алгоритм оценивания результатов тестирования

**Начало**

| **целое** *число правильных* = 0;

| **булево** *правильность ответа* = true;

| для (**целое** *i* = 0; *i* < *число вопросов*; ++*i*)

| | **целое** *номер вопроса*;

| | **строка** *текст ответа*;

| | | **выбрать** (от *тип вопроса*)

| | | | **случай** (*закрытый вопрос*):

| | | | | *ответ* = *ответы. закрытый вопрос*[*i*];

| | | | | **если** (*ответ верный*)

| | | | | | ++*количество верных*;

| | | | | | *correct*[*i*] = *истина*;

| | | | | **конец**;

| | | | **завершить;**

| | | | **случай** (*открытый вопрос)*:

| | | | | *ответ*  = *ответы. открытый вопрос* [*i*];

| | | | | **если** (*ответ верный*)

| | | | | | ++ *число правильных*;

| | | | | | *правильные*[*i*] = истина;

| | | | | **конец**;

| | | | | **завершить**;

| | | | **вернуть** *число правильных*;

| | | **конец**;

| | **конец**;

| **конец**;

**Конец**

#### 1.2.7 Алгоритм демонстрации работы триода

**Начало**

| **пока** (не нажата кнопка возврата)

| | определить *тип подключения* триода;

| | **если** (пояснения включены)

| | | вывести пояснения;

| | **конец**;

| | определить сопротивление на сетке и напряжение цепи;

| | установить значения сопротивления и напряжения в *Триоде*;

| | произвести расчёт характеристик триода, вызвав функции *Триода*;

| | определить *Режим* работы лампы;

| | вывести анимированное изображение *Лампы* на экран;

| | отрисовать график анодно-сеточной характеристики;

| **конец**;

**Конец**

## 1.3 Входные и выходные данные

Входные данные:

* пункт меню – выбор полярности подключения триода;
* пункт меню – выбор функции вывода пояснений;
* целое число – количество вопросов теста;
* целое число – ответ на вопрос теста;
* целое число – напряжение в цепи;
* целое число – сопротивление на сетке.

Выходные данные:

* строки – теоретический материал;
* строки – вопросы теста;
* строки – результаты тестирования;
* числовые данные – результаты расчётов;
* графическая информация – физические процессы в триоде.

## 1.4 Системные требования

Аппаратные требования:

* Intel-совместимый процессор с частотой не менее 1,6 ГГц.
* Не менее 512 Mb ОЗУ.
* Не менее 20 Mb свободного места на жестком диске.
* Средства ввода: клавиатура, мышь, монитор.

Операционная система: Windows 7 и более новые, разрядность ОС значения не имеет.

# 2 РАБОЧИЙ ПРОЕКТ

## 2.1 Общие сведения о работе системы

Программный продукт разработан в интегрированной среде Qt (версии 5.14.2) на языке С++ (стандарт 11) с использованием компилятора MinGW (7.3.0). Программа работает под управлением операционной системы Windows 7 Professional и более поздними.

## 2.2 Функциональное назначение программного продукта

Разработанный программный продукт предназначен для ознакомления пользователя со структурой данных «список с пропусками», предоставляющий возможность изучить теоретический материал, рассмотреть выполнение основных операций в режиме демонстрации, пройти тест на знание алгоритмов работы списков с пропусками. Программа имеет следующие функциональные возможности:

* предоставление теоретического материала;
* демонстрация выполнения основных операций со списками (вставка узла, удаление узла, поиск по ключу);
* переход между шагами демонстрации;
* предоставление объяснения текущего шага визуализации;
* прохождение тестирование на знание теоретического материала;
* прекращение тренировки по желанию пользователя, либо при окончанию тестовых вопросов.

Программа имеет следующие функциональные ограничения

* при работе в демонстрационном режиме необходимо ввести от 3 до 5 узлов списка;
* значениями узлов могут быть только целые числа в диапазоне от -99 до 99.
* для работы с теоретическим и демонстрационным режимами используются файлы questions.txt, theory\_text.txt и insert.txt, delete.txt, find.txt;
* количество вариантов ответа для теста не превышает 3.

## 2.3 Инсталляция и выполнение программного продукта

Для выполнения программы необходимо:

1. Скопировать на жёсткий диск установщик Setup.exe.
2. Запустить установщик, выбрать место расположения программы.
3. Запустить программу.

## 2.4 Общий алгоритм программного продукта

Программный продукт состоит из 5 модулей: демонстрационного, теоретического, модуля тестирования, главного меню и модуля, описывающего алгоритмы работы списков с пропусками.

Модуль теории устроен довольно просто, в главном пользовательском окне демонстрируется теоретическая выдержка по теме. Окно можно прокручивать, текст раздела состоит из строкового материала и изображений. В таблице 2.1 приведены сигналы и слоты, используемые в модуле теоретического материала.

Таблица 2.1 – Сигналы и слоты модуля чтения теории

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| void on\_menuPage\_clicked() | Нажата кнопка возврата в главное меню |
| void back\_to\_menuPage() | Эмитирован сигнал возврата в меню |

В таблице 2.2 приведены функции и процедуры, используемые в модуле демонстрации.

Таблица 2.2 – Функции и процедуры модуля демонстрации

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| void highlight(steps step) | Выделение элемента узла, над которым было произведено действие |
| void all\_clear() | Установка стартовых настроек: очистка форма, удаление списка и атрибутов демонстрации |
| void createList() | Создание списка с пропусками на основе чисел, введённых пользователем в меню демонстрации |
| void drawNumbers() | Вывод первого уровня списка - значений узлов |
| void drawCells() | Отрисовка прямоугольных блоков - уровней списка |

В таблице 2.3 приведены сигналы и слоты, используемые в модуле демонстрации.

Таблица 2.3 – Сигналы и слоты модуля демонстрации

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| void on\_pbInsert\_clicked(); | Выбран показ операции вставки узла в список |
| void on\_pbDelete\_clicked(); | Выбран показ операции удаления узла из списка |
| void on\_pbFind\_clicked(); | Выбран показ операции поиска узла в списке |
| void on\_pbToMenu\_clicked(); | Нажата кнопка возврата в меню демонстрации |

Продолжение таблицы 2.3

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| void on\_back\_clicked(); | Переход к следующему шагу операции |
| void on\_next\_clicked(); | Возврат к предыдущему шагу демонстрации |
| void on\_toMenuPage\_clicked(); | Возврат в главное меню |
| void on\_newValue\_textChanged(const QString &arg1); | Ввод значения вставляемого узла |
| void on\_key\_textChanged(const QString &arg1); | Ввод ключа вставляемого узла |

В таблице 2.4 приведены важнейшие переменные, используемые в модуле демонстрации.

Таблица 2.4 – Важнейшие переменные модуля демонстрации

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| cur\_step | size\_t | Текущий шаг операции |
| skiplist | Skiplist | Список с пропусками, над которыми выполняются операции |
| skiplistDemo | QPixmap | Поле для отрисовки |
| painter | QPainter | Инструмент отрисовки графических примитивов и текста на форме |
| startDemo | QPoint | Стартовая точка вывода узлов списка |
| endDemo | QPoint | Конечная точка вывода узлов списка |
| fontDemo | QFont | Характеристики текстового шрифта на форме |
| cellsize | QSize | Размер клетки |
| cell | QRect | Прямоугольная клетка - уровень списка |

В таблице 2.5 приведены функции и процедуры, используемые в модуле тестирования.

Таблица 2.5 – Функции и процедуры модуля тестирования

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| std::string read\_testFile(const std::string &filename); | Считывание текста вопросов из файла |
| void begin\_test(); | Начало выполнение теста |
| void make\_question(); | Создание вопроса из текстового файла |

Продолжение таблицы 2.5

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| void display\_question(); | Вывод вопроса с вариантами ответа на экран |
| void display\_result(); | Вывод результатов тестирования |
| int check(std::vector<bool> &correct); | Вектор безошибочных ответов, данных пользователем |

В таблице 2.6 приведены важнейшие переменные, используемые в модуле тестирования.

Таблица 2.6 – Важнейшие переменные модуля тестирования

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| questionnaire | string | Текст вопросов из файла |
| test\_questions; | vector<question> | Вопросы тестирования |
| answers | answer\_types | Ответы на вопросы теста |
| number\_of\_questions | size\_t | Количество вопросов |
| step | size\_t | Номер вопроса/номер ответа |

В таблице 2.7 приведено описание структуры ответа, используемого в модуле тестирования.

Таблица 2.7 – Описание структуры ответа

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| condition | QString | Текст варианта ответа |
| type | bool | Правильность ответа |

В таблице 2.8 приведено описание структуры вопроса, используемого в модуле тестирования.

Таблица 2.8 – Описание структуры вопроса

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| text | QString | Формулировка вопроса |
| type | int | Тип вопроса (закрытый/открытый) |
| variants | answer | Варианты ответов на вопросы теста |

В таблице 2.9 приведено описание структуры Skipnode - узла списка с пропусками, используемого в модуле алгоритма работы списка с пропусками.

Таблица 2.9 – Важнейшие переменные модуля работы списков с пропусками

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| key | size\_t | Ключ узла списка |
| value | int | Значение узла списка |
| next | vector <Skipnode\*> | Уровни списка, содержащие указатели на соответствующие уровни узлов-наследников |

В таблице 2.10 приведены важнейшие переменные, используемые в модуле алгоритмов работы списков с пропусками.

Таблица 2.10 – Важнейшие переменные модуля работы списков с пропусками

|  |  |  |
| --- | --- | --- |
| **Имя** | **Тип** | **Назначение** |
| level | size\_t | Уровень узла списка |
| head | Skipnode\* | Указатель на заголовок списка |
| next | vector <Skipnode\*> | Уровни списка с указателями на соответствующие уровни узлов-наследников |
| random\_engine | mt19937 | Устройство генерации случайных чисел |

В таблице 2.11 приведены функции и процедуры, используемые в модуле алгоритмов работы со списками пропусков.

Таблица 2.11 – Функции и процедуры основных операций над списками пропусков

|  |  |
| --- | --- |
| **Прототип** | **Назначение** |
| void initialize(size\_t key, int value) | Инициализация списка с пропусками |
| void insert(size\_t key, int value) | Вставка в список с пропусками |
| bool erase(size\_t key) | Удаление из списка с пропусками |
| int find(size\_t key) | Поиск в списке с пропусками |
| void clear() noexcept | Очистка списка с пропусками |
| size\_t size() const | Количество узлов в списке с пропусками |
| size\_t top\_level\_of\_new\_node() | Генерация нового уровня в списке с пропусками |

## 2.5 Разработанные интерфейсы

После запуска программы на выполнение первым будет выведена форма стартового меню, представленная на рисунке 2.1

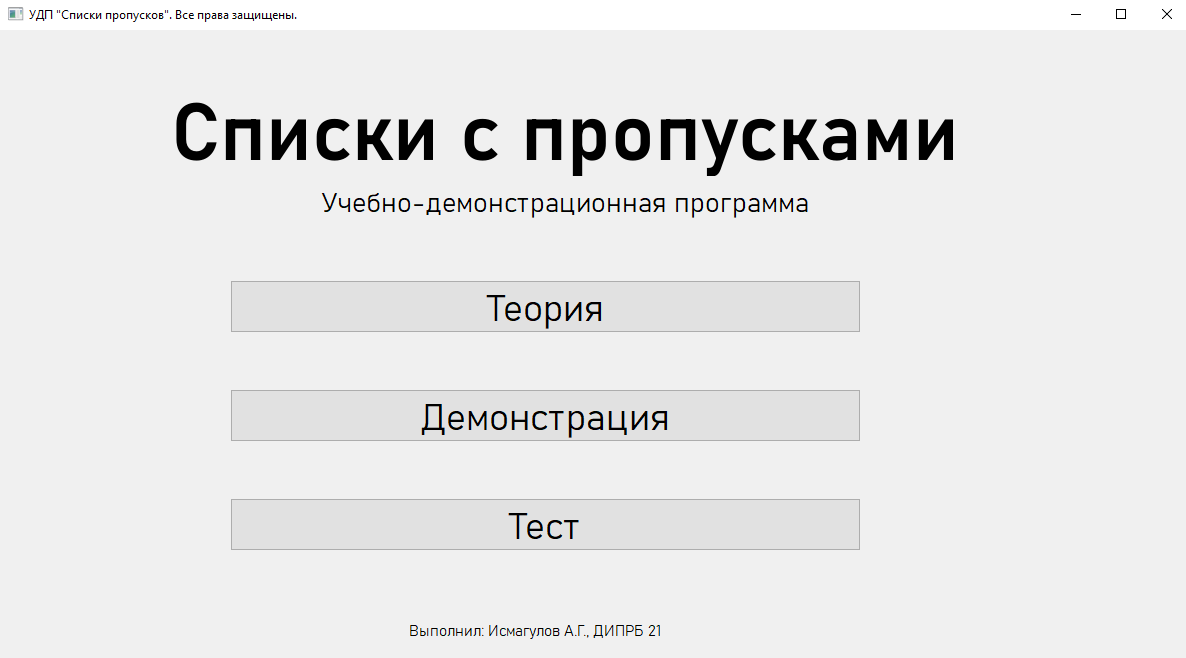


Рисунок 2.1 – Стартовая страница программы

В меню доступны три кнопки: «Теория», «Демонстрация», «Тест». По нажатию на каждый откроется соответствующий раздел.

При нажатии на кнопку «Теория» будет отображено окно с текстовым материалом. В правом верхнем углу окна расположена кнопка «Вернуться в меню» для перехода в главное меню (см. рис. 2.1).

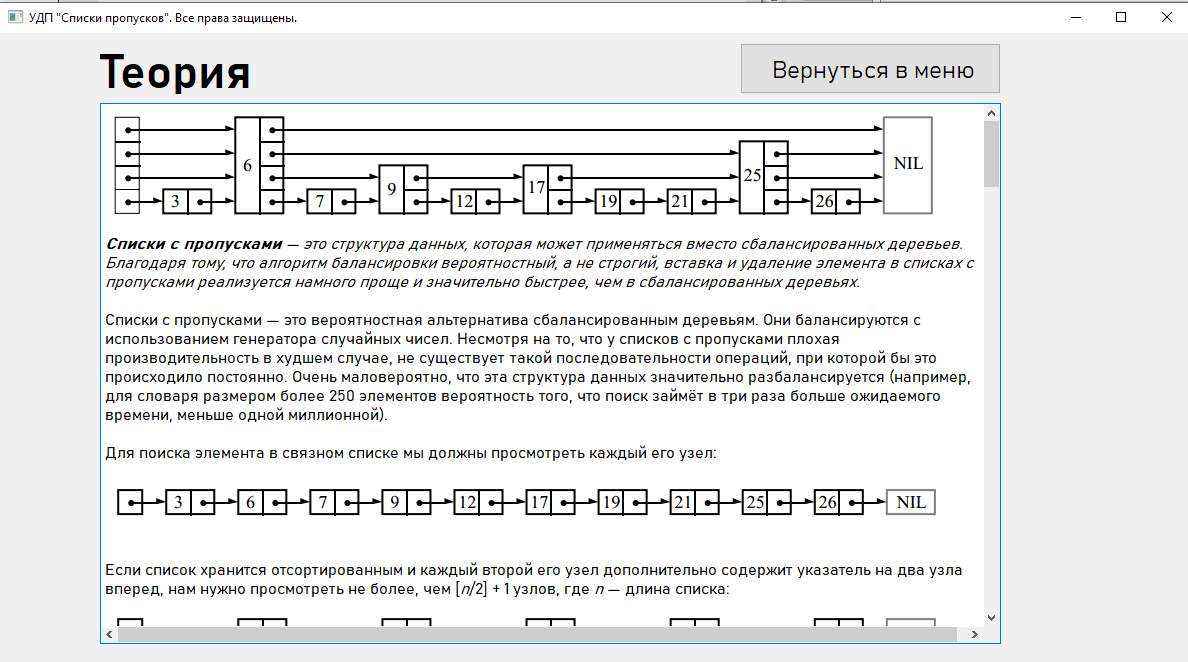


Рисунок 2.2 – Раздел с теоретическим материалом

При нажатии на кнопку «Демонстрация» произойдёт переход в меню раздела. представленное на рисунке 2.3. Кнопки «Вставка элемента», «Удаление элемента», «Поиск элемента» будут недоступны, так как не введены атрибуты: ключ и значение. Для создания списка пользователю требуется ввести элементы в поле ввода, расположенному под кнопками операций.

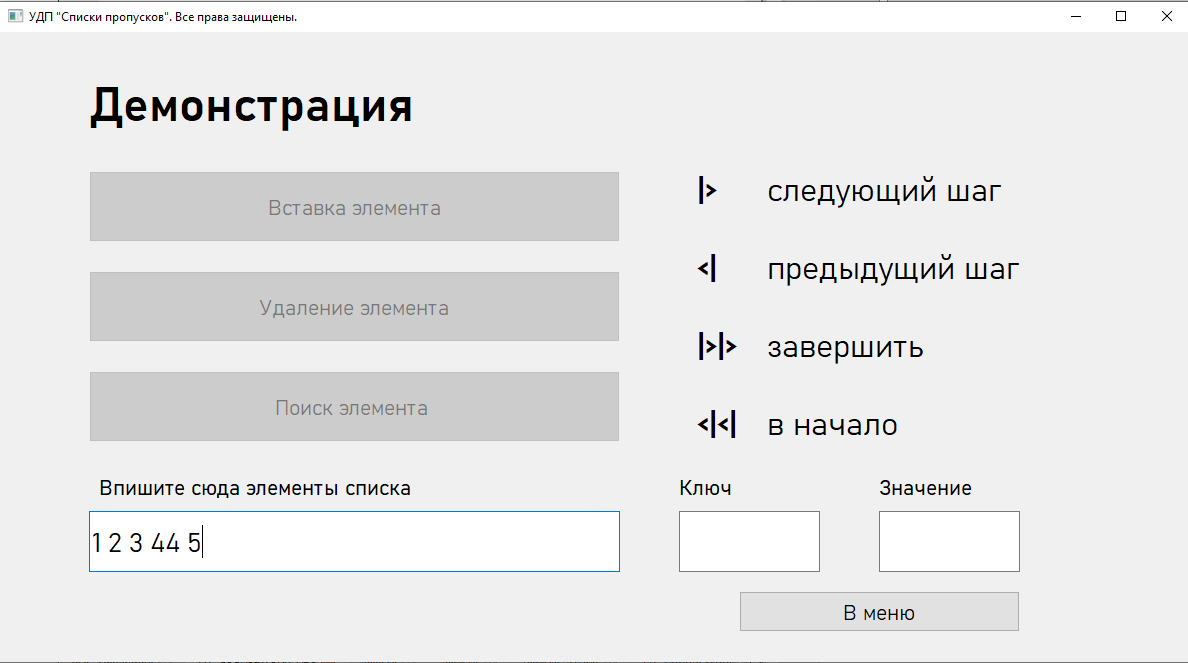


Рисунок 2.3 – Меню раздела «Демонстрация»

После ввода соответствующих атрибутов и нажатия на кнопку «Вставка элемента» на экране появится окно с демонстрацией выбранной операции, показанное на рисунке 2.4.

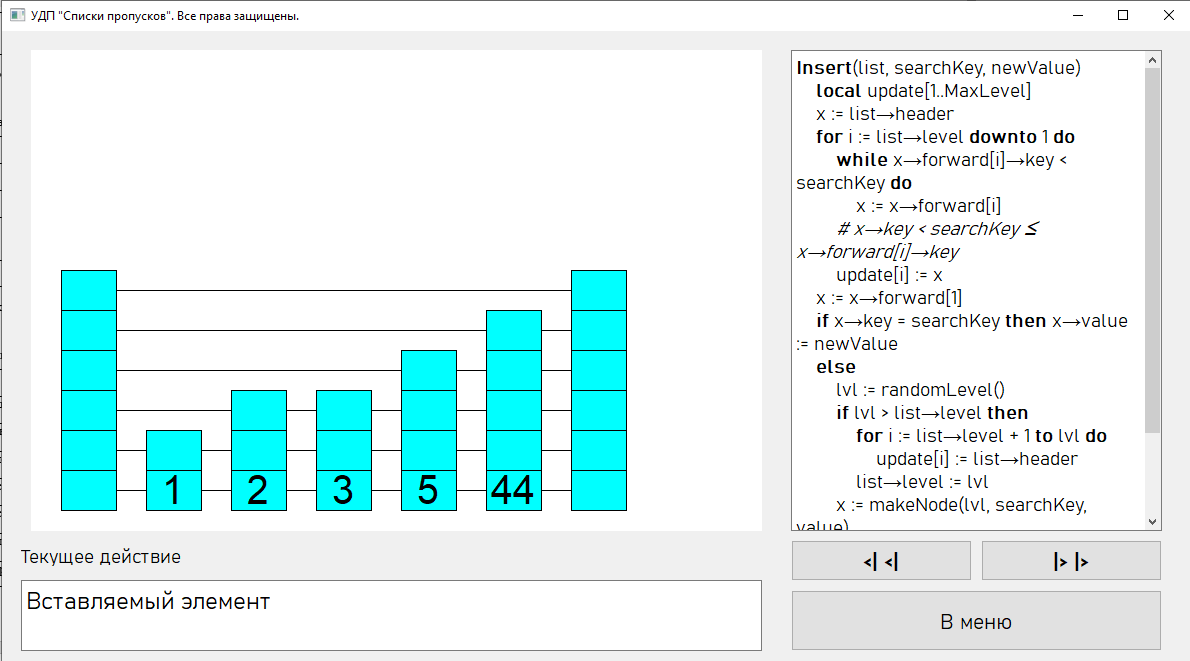


Рисунок 2.4 – Демонстрация операции над списком с пропусками

Для перехода между шагами операции следует нажать на кнопки перехода, расположенным в правом нижнем углу окна. При нажатии на кнопку «Вернуться в меню» произойдет переход в окно, представленное на рисунке 2.3. При ещё одном нажатии на клавишу произойдет переход в главное меню, которое представлено на рисунке 2.1.

Описание текущего шага выводится в поле текстового браузера. На рисунке 2.4 поле описания названо «Текущее действие».

При нажатии в главном меню на кнопку «Тест» произойдет переход в раздел тестирования. На рисунке 2.5 представлена форма окна тестирования.

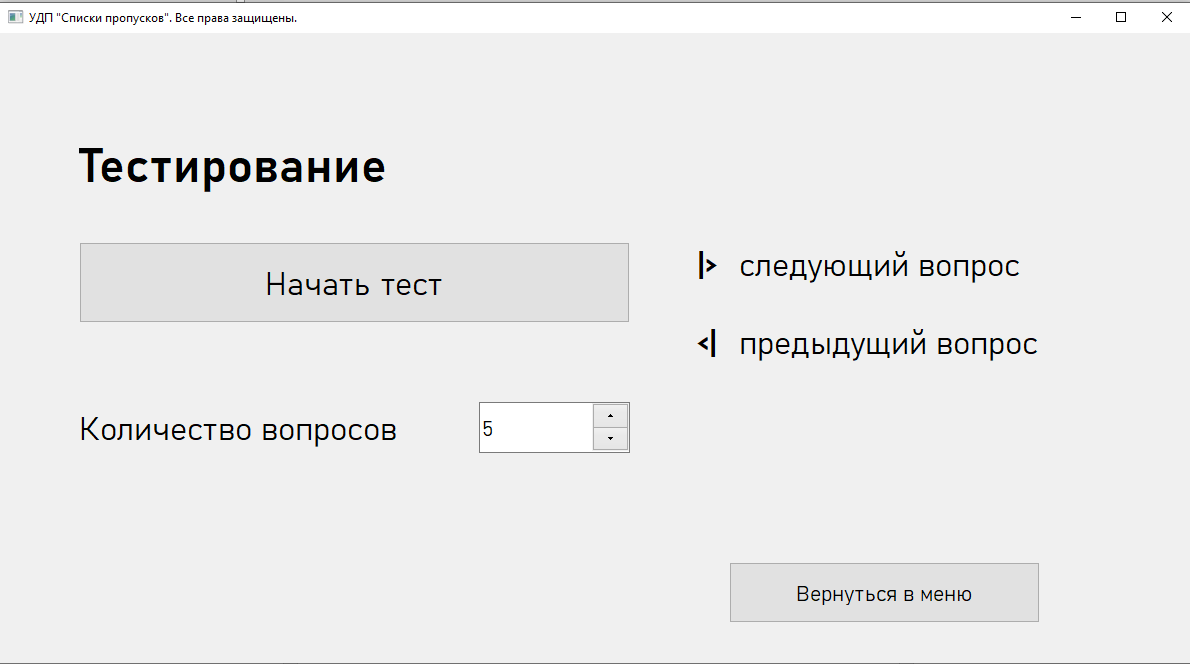


Рисунок 2.5 – Стартовая страница раздела тестирования

Пользователю необходимо выбрать количество вопросов в тесте и нажать кнопку «Начать тест». Далее произойдёт переход в окно для ответа на вопросы. На рисунке 2.6 показан вариант вопроса закрытого типа.

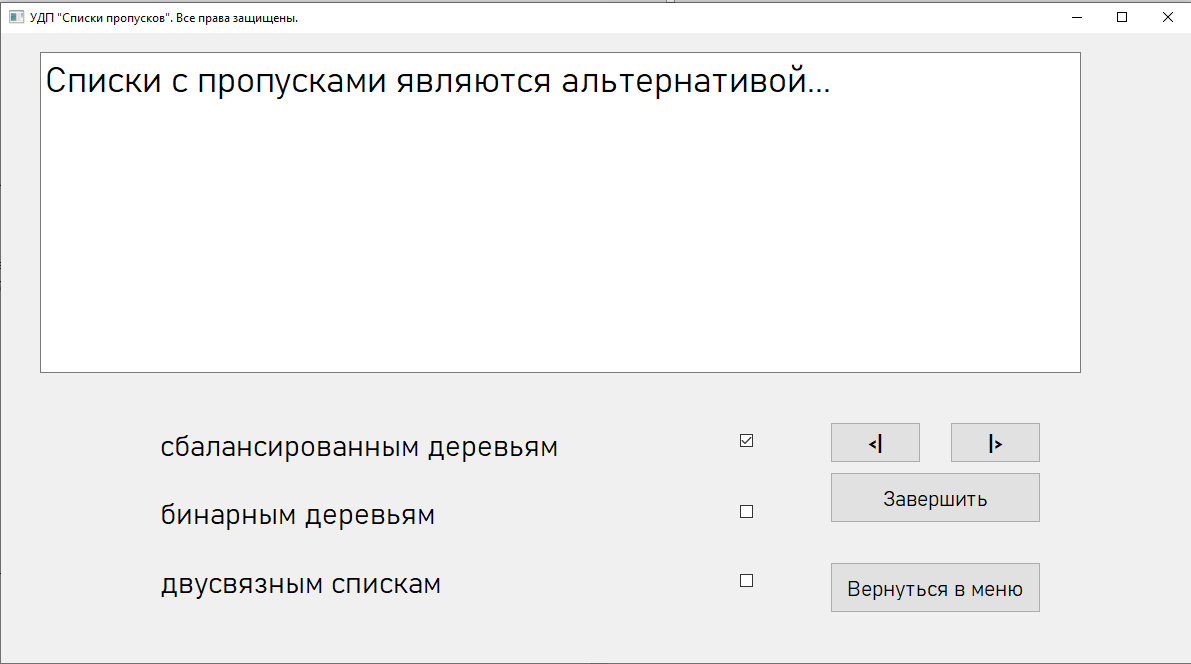


Рисунок 2.6 – Вопрос с вариантами ответов

Для перехода между вопросами необходимо нажать на кнопки в правом нижнем углу окна. Чтобы досрочно сдать тест следует нажать кнопку «Завершить».

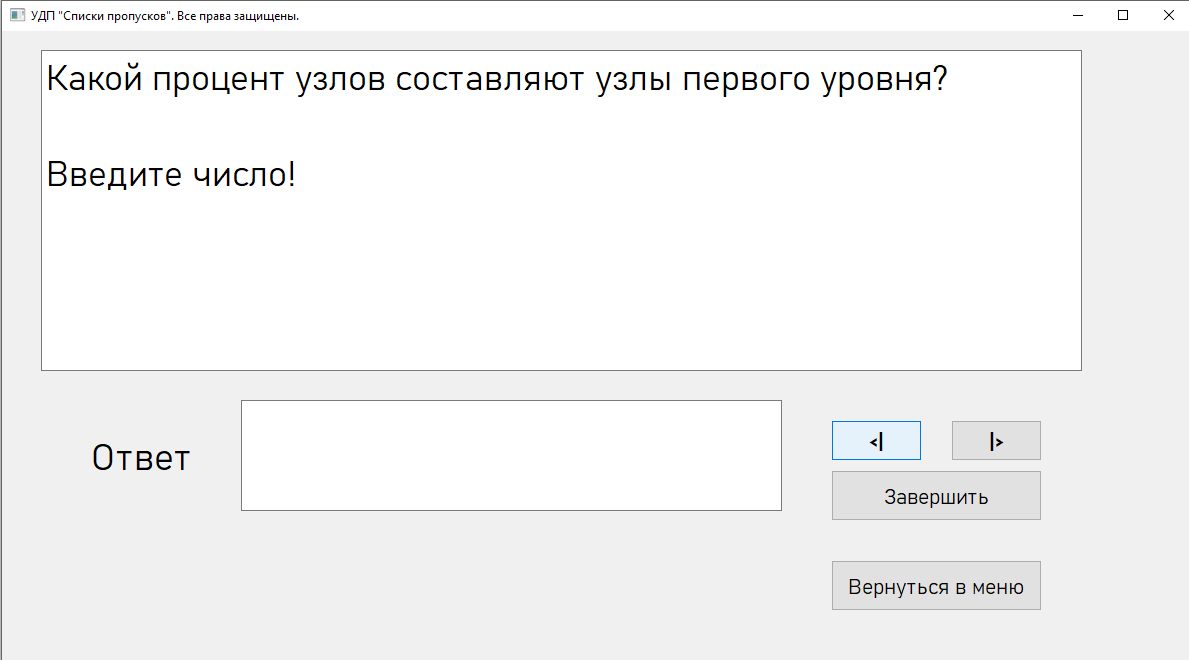
Пользователю следует выбрать один из вариантов ответа и перейти к следующему вопросу. Чтобы отметить вариант ответа, необходимо выбрать одно из полей checkbox. Ответ на вопрос без вариантов записывается в поле ввода, обозначенное названием «Ответ». На рисунке 2.7 представлен вопрос открытого типа.

Рисунок 2.7 – Вопрос открытого типа

После ввода ответов на все вопросы теста появляется окно с результатами. Все вопросы, на которые были даны неверные ответы будут выведены в специальном окне, что показано на рисунке 2.8.

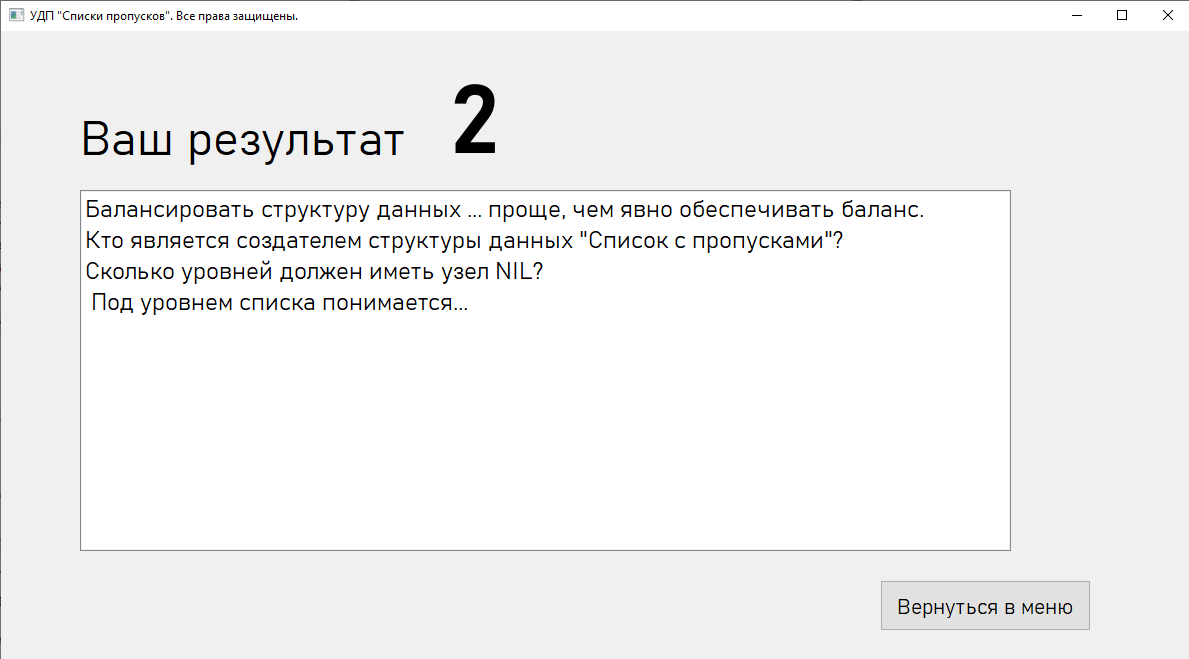


Рисунок 2.8 – Результаты тестирования

# 3 ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

1. Запустить файл skiplist.exe.
2. Убедиться в открытии формы представленной на рисунке 2.1.
3. Нажать на клавишу «Открыть теоретический материал».
4. Убедиться в открытии формы представленной на рисунке 2.2.
5. Нажать на кнопку «Вернуться в меню».
6. Убедиться в открытии формы на рисунке 2.1.
7. Нажать на кнопку «Демонстрация».
8. Убедиться в открытии формы на рисунке 2.3.
9. Ввести данные в форму.
10. Нажать на клавишу «Вставка элемента».
11. Убедиться в открытии формы представленной на рисунке 2.4.
12. Нажать на кнопку перехода к следующему шагу.
13. Убедиться, что в выводе нового шага операции.
14. Пролистать шаги сортировки до конца и убедиться в правильности вывода.
15. Нажать на кнопку «Вернуться в меню».
16. Убедиться в открытии формы на рисунке 2.1.
17. Нажать на клавишу «Тест».
18. Убедиться в открытии формы на рисунке 2.5.
19. Нажать на клавишу «Начать тест».
20. Ответить на вопросы.
21. Получить по результатам прохождения теста оценку - окно с формой на рисунке 2.8.
22. Нажать на клавишу «Вернуться в меню».
23. Убедиться в открытии формы представленной на рисунке 2.1.
24. Завершить работу программы, нажав на крестик в правом верхнем углу окна программы.

# ЗАКЛЮЧЕНИЕ

В результате курсового проектирования была разработана «Учебно-демонстрационная программа «Списки с пропусками», позволяющая получить теоретический материал о структуре данных «списки с пропусками», наглядно рассмотреть пошаговое выполнение основных операций в режиме демонстрации, пройти тест на знание алгоритмов работы со списками пропусков.

Программа отвечает всем поставленным требованиям и может быть использована для обучения студентов.

#### СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кормен Т.Х. Алгоритмы: построение и анализ, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1296 с. : ил. – Парал. тит. англ.
2. Лаптев В.В. С++. Объектно-ориентированное программирование: Учебное пособие. – СПб.: Питер, 2008. – 464 с.
3. Cеджвик Р. Фундаментальные алгоритмы на C++. Анализ / Структуры данных / Сортировка / Поиск: Пер. с англ. /- К.: Издательство «ДиаСофт», 2001.- 688

# ПРИЛОЖЕНИЕ 1

**Техническое задание**

**на разработку учебно-демонстрационной программы**

**«Списки с пропусками»**

по дисциплине «Алгоритмы и структуры данных**»**

Направление 09.03.04 Программная инженерия

Исполнитель: студент гр. ДИПРб21 **Исмагулов А.Г.**

1. **Назначение, цели и задачи разработки**

**Цель разработки** – Разработка и реализация учебно-демонстрационной программы «Cписки пропусков»

**Назначение разработки:** предоставление теоретической информации о списке пропусков, демонстрация основных операций над списками пропусков, предоставление возможности пройти тест по основным теоретическим сведениям о работе списков с пропусками.

**Основные задачи,** решаемые разработчиком в процессе выполнения курсового проекта:

* + анализ предметной области;
  + разработка программного продукта в соответствии с требованиями;
  + документирование проекта в соответствии с установленными требованиями.

1. **Характер разработки:** прикладная квалификационная работа.
2. **Основания для разработки**
   * Учебный план направления 09.03.04 «Программная инженерия» 2018 года набора.
   * Рабочая программа дисциплины «Алгоритмы и структуры данных».
   * Распоряжение по кафедре АСОИУ №\_\_\_\_ от «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2019 г.
3. **Плановые сроки выполнения –** зимний семестр 2019/20 учебного года:

Начало «15» сентября 2019 г.

Окончание «30» декабря 2019 г.

1. **Требования к проектируемой системе**

**5.1 Требования к функциональным характеристикам**

* предоставление теоретического материала;
* демонстрация выполнения основных операций со списками (вставка узла, удаление узла, поиск по ключу);
* переход между шагами демонстрации;
* предоставление объяснения текущего шага визуализации;
* прохождение тестирование на знание теоретического материала;
* прекращение тренировки по желанию пользователя, либо при окончанию тестовых вопросов.

Программа имеет следующие функциональные ограничения

* при работе в демонстрационном режиме необходимо ввести от 3 до 5 узлов списка;
* значениями узлов могут быть только целые числа в диапазоне от -99 до 99.
* для работы с теоретическим и демонстрационным режимами используются файлы questions.txt, theory\_text.txt и insert.txt, delete.txt, find.txt;
* количество вариантов ответа для теста не превышает 3.

**5.2 Требования к эксплуатационным характеристикам**

Программа не должна аварийно завершаться при любых действиях пользователя.

Время реакции программы на действия пользователя не должно превышать 10 секунд.

**5.3 Требования к программному обеспечению:**

Средства разработки: интегрированная среда Qt (версия 5.14.2), язык C++ (стандарт С++ 11 и выше). Операционная система: Windows XP (x86) с пакетом обновления 3 (SP3), либо Alt Linux 7.0.

**5.4 Требования к аппаратному обеспечению:**

Рекомендуемая конфигурация:

* + Intel-совместимый процессор с частотой не менее 1,6 ГГц;
  + не менее 512 МБ ОЗУ;
  + не менее 50 КБ свободного места на диске;
  + дисковод CD-ROM/DVD-ROM.

1. **Стадии и этапы разработки**
2. **Эскизный проект (ЭП)**
   * Анализ предметной области.
   * Подготовка проектной документации.
3. **Технический проект (ТП)**
   * Разработка структур и форм представления данных.
   * Разработка структуры программного комплекса.
   * Подготовка пояснительной записки.
4. **Рабочий проект (РП)**
   * Программная реализация.
   * Тестирование и отладка программы.
   * Подготовка программной и эксплуатационной документации.
5. **Эксплуатация (Э)**

Описание и анализ результатов проведенного исследования.

1. **Требования к документированию проекта**

К защите курсового проекта должны быть представлены следующие документы:

* + Пояснительная записка к курсовому проекту:
  + Презентация доклада.
  + Программа, презентация и пояснительная записка к курсовому проекту на оптическом носителе.

Требования к структуре документов определены соответствующими стандартами ЕСПД.

Требования к оформлению определены соответствующими методическими указаниями.

1. **Порядок контроля и приемки**

Контроль выполнения курсового проекта проводится руководителем поэтапно в соответствии с утвержденным графиком выполнения проекта.

На завершающем этапе руководитель осуществляет нормоконтроль представленной исполнителем документации и принимает решение о допуске (недопуске) проекта к защите.

Защита курсового проекта проводится комиссией в составе не менее двух человек, включая руководителя проекта.

В процессе защиты проекта исполнитель представляет документацию, делает краткое сообщение по теме разработки и демонстрирует ее программную реализацию.

При выставлении оценки учитывается:

* + степень соответствия представленной разработки требованиям технического задания;
  + качество программной реализации, документации и доклада по теме проекта;
  + соблюдение исполнителем графика выполнения курсового проекта.

1. **Литература**
2. Кормен Т.Х. Алгоритмы: построение и анализ, 2-е издание.: Пер. с англ. – М.: Издательский дом «Вильямс», 2005. – 1296 с. : ил. – Парал. тит. англ.
3. Лаптев В.В. С++. Объектно-ориентированное программирование: Учебное пособие. – СПб.: Питер, 2008. – 464 с.
4. Cеджвик Р. Фундаментальные алгоритмы на C++. Анализ / Структуры данных / Сортировка / Поиск: Пер. с англ. /- К.: Издательство «ДиаСофт», 2001.- 688

# ПРИЛОЖЕНИЕ 2