**Введение**

На курсовое проектирование была поставлена задача, разработать программу на тему: Электронный учебный помощник «Состав ПК».

Цель курсового проекта заключается в знакомстве массового пользователя со системным блоком персонального компьютера и его компонентами.

Создаваемая программа будет рассчитана на любого рода пользователей. Применить данную программу смогут не только люди, которые плохо разбираются в персональных компьютерах, но также и люди, работающие за персональным компьютером, не зная его строение.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно- экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создается данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Литературе” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

1. **Анализ задачи**

**1.1 Постановка задачи**

Темой данного курсового проекта является «Состав ПК».

В наше время никак не обойтись без компьютерных технологий. Они играют решающую роль в современном мире, повышают эффективность работы, обеспечивают быстрый и удобный доступ к информации, упрощают коммуникацию и создают новые возможности во всех сферах жизни. Однако возникают и новые проблемы, такие как кибербезопасность и частная жизнь. Поэтому важно развивать информационные технологии, учитывая эти вызовы, и стремиться к их устойчивому и этичному использованию.

В период, начиная с появления первого компьютера в 1941 году и до наших дней, компьютерный мир очень сильно изменился в лучшую сторону. Но по сравнению с прошлым столетием, сегодня компьютер себе позволить могут многие.

Персональные компьютеры повышают производительность труда конкретных работников. Они удобны в использовании, а разработка приложений относительно быстрая и гибкая. Однако хотя один персональный компьютер недорог, все вместе они стоят столько же, сколько большой компьютер. Они требуют поддержки, соединения в сеть, наличия серверов, а также многих других составляющих инфраструктуры. Их экономическую эффективность трудно определить, так как они предназначены для удовлетворения потребностей индивидуумов, а не организации и невозможно прямо оценить прибыль от их использования.

Как же выбрать компьютер, который подойдет конкретному человеку?

Конечно же нет одного ПК, адаптированного под любого человека. Именно поэтому у каждого компьютера есть свои компоненты с разными характеристиками, влияющие на его работу. Для этого и создан программный продукт, а именно – изучение пользователем компонентов ПК и их предназначения.

Периодичность использования данного ПП не ограничена. Пользователь может в любое время установить, а также удалить приложение после полного изучения материала по данной теме и прохождения тестов.

Цель данного курсового проекта – разработать программный продукт, который будет сочетать в себе элементы обучения, проверки усвоенного материала и развлечения для людей, интересующихся компьютерами.

За основу программы следует взять из соответствующих источников (сайтов, статей) наиболее полную, достоверную информацию.

При реализации проекта разработчик должен стараться сделать его простым и максимально доступным для пользователей. Приложение должно располагать своим интерфейсом и функционалом к пользователю, чтобы ему было комфортно.

Данный проект должен стать общедоступным для всех пользователей. В поставленной задаче необходимо реализовать максимально простой пользовательский интерфейс, позволяющий использовать проект пользователю, не обладающему дополнительными знаниями электронно-вычислительной техники.

Разрабатываемый программный продукт должен позволять выполнять следующие действия:

* обеспечить возможность просмотра информации про каждый компонент системного блока;
* возможность сыграть в игру «Собери сам»;
* прохождения теста, в цели закрепления просмотренной информации;
* просмотр пункта «О программе»;
* просмотр справки.

В данном программном продукте будут использоваться три вида информации. К входной информации можно отнести вводимые пользователем значения, например ответы на вопросы при прохождении теста. К выходной – результаты теста, информация о компонентах системного блока. Постоянной информацией будут являться текстовые файлы, картинки, веб-страницы.

Программный продукт предоставляет функционал для следующего ряда пользователей: администратор – организовывает работу программой, гость.

**1.2 Инструменты разработки**

Для разработки данного проекта будет выбрана среда Delphi 7, так как это самое удобная и доступная среда разработки на данный момент. Delphi 7 – язык-программирования, относящийся к классу RAD- (Rapid Application Development – «Средство быстрой разработки приложений») средств CASE – технологии. Delphi 7 сделал разработку приложений для windows быстрым и приятным процессом. Теперь разрабатывать сложные и интересные проекты можно только одним человеком, использующим Delphi 7.

Delphi — высокоуровневый язык программирования, который сочетает в себе мощные средства разработки приложений, простоту и эффективность. Разработанный компанией Borland в 1995 году, Delphi быстро стал популярным выбором для создания программного обеспечения в различных сферах.

Язык программирования Delphi предлагает широкий набор инструментов и простоту разработки, что делает его привлекательным выбором для разработчиков. Однако, у него есть и свои ограничения. При выборе Delphi нужно учитывать требования проекта и доступные ресурсы, чтобы принять решение, соответствующее потребностям.

Также в программном продукте будут использованы и другие инструменты разработки.

**Help+Manual** – это программа от компании EC Software, способная генерировать help-файлы самых различных форматов. Главным преимуществом программы является ее универсальность. С ее помощью можно получить файл справочной информации в любом из наиболее распространенных на сегодняшний день форматов (CHM, HLP, HXS, HTML, PDF, RTF, EXE, XML). Интуитивно понятный интерфейс делает программу простой в освоении. Основной блок программы составляет текстовый редактор, мало отличимый от MS Word как по интерфейсу, так и по количеству возможностей.

**Smart Install Maker** – это мощное и удобное средство для создания инсталляторов. Программа обладает удобным и интуитивно понятным интерфейсом, а также полным набором необходимых функций для создания профессиональных инсталляторов с минимальным размером, высокой степенью сжатия файлов и приятным интерфейсом. Инсталляторы, созданные в среде Smart Install Maker имеют небольшой размер и поддерживают превосходный формат сжатия Cab. Все это позволяет создавать действительно компактные дистрибутивы, которые не создадут проблем при их распространении по интернету.

**Microsoft Word 2016** – редактор текста для написания документации.

**Microsoft Power Point 2016** – программа для создания презентации.

Разработка ведется на ноутбуке Machenike со следующими параметрами:

* процессор 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz;
* объем ОЗУ 8гб;
* объем места на HDD 500 гб;
* видеоадаптер NVIDIA GeForce RTX 3060 Laptop GPU;
* ОС – Windows 11.

**1.3 Требования к приложению**

Для разработки доступного любому пользователю программного продукта нужно подобрать оптимальный интерфейс. Красивый, но и в то же время без лишних элементов.

**Требования к интерфейсу:** программа не должна иметь очень ярких (некомфортных глазу) цветов, а также легкодоступную и понятную навигацию. Следовательно, каждое окно программного продукта должно иметь не яркую цветовую гамму, а также расположение компонентов должно быть гармонично связано друг с другом и не загромождать другие элементы окна.

**Требования к надежности:** специальных требований к надежности не предъявляется. Время восстановления после отказа, вызванного сбоем электропитания технических средств (иными внешними факторами), не фатальным сбое (не крахом) операционной системы, не должно превышать 30-ти минут при условии соблюдения условий эксплуатации технических и программных средств.

Время восстановления после отказа, вызванного неисправностью технических средств, фатальным сбоем (крахом) операционной системы, не должно превышать времени, требуемого на устранение неисправностей технических средств и переустановки программных средств.

Отказы программы вследствие некорректных действий пользователя при взаимодействии с программой через графический интерфейс не должны влиять на конечный результат. ПП должен иметь средства фиксации всех действий в процессе изготовления программного продукта. Это необходимо для восстановления состояния процесса разработки на любом его этапе (при комплексной отладке).

**Требования к применению:** программный продукт помогает разобраться в устройстве системного блока персонального компьютера. Возможность просмотра информации и картинок компонентов персонального компьютера. Возможность проверить свои знания в виде теста.

**2 Проектирование задачи**

**2.1 Организация данных**

Необходимым условием для разработки данного программного продукта является описание организации данных, т.е. логическая и физическая структура данных в контексте среды разработки. В разрабатываемой программе будут использоваться три вида данных.

Первые данные – это данные, введенные разработчиком на этапе реализации. В программном продукте «Состав ПК» этими данными являются изображения (\*.jpg, \*.png, \*.bmp), текстовые файлы, html-страницы.

Вторым видом данных является вводимая пользователем информация. К входной информации программного продукта можно отнести ответы на вопросы в тесте.

Третьим видом данных является результат программы. В случае программы «Состав ПК» это результат прохождения игры «Собери сам» и результат теста, который пройдет пользователь.

**2.2 Процессы**

Для реализации разного вида задач будут использованы разного вида процедуры. Например, будет осуществляться загрузка html-страниц в компонент TWeb-Browser, создание игры и теста.

Тест будет сгенерирован на основе выбора пользователем его сложности в компоненте TRadioGroup. Вопросы для теста будут подгружаться в программу из txt-файлов. У пользователя будет возможность пройти тест заново, нажав на кнопку, сделанную в виде перезагрузки с помощью компонента TImage. В конце теста в отдельном поле будут показаны результаты его прохождения в процентах.

Игра собери сам сделана на основе свойства DragAndDrop. Пользователь имеет возможность включить подсказки, которые подсветят места компонентов системного блока, на которые нужно установить сами компоненты. После сборки нужно нажать на кнопку «Нажми, если собрал», которая реализована с помощью компонента TSpeedButton. После нажатия будет высвечена информация о том, собрал ли пользователь ПК полностью.

Окно для изучения информации будет реализовано с помощью множества компонентов TImage. После нажатия на один из компонентов пользователю будет предложено изучить полную информацию. Для перехода на следующее окно нужно нажать на кнопку «Узнать больше».

В информационном окне находится компонент TWeb-Browser, в который подгружается информация из html-файла.

**2.3 Описание внешнего пользовательского интерфейса**

Важной частью программного продукта является внешний пользовательский интерфейс. Внешний вид организован так, чтобы пользователю было понятно какие компоненты, для чего предназначены. Таким образом ключевые элементы должны быть выделены.

Для успешной работы всего программного продукта реализован интуитивно понятный интерфейс и управление кнопками, выделенных на фоне заднего плана форм.

Структура навигации по приложению представлена на рисунке 1.

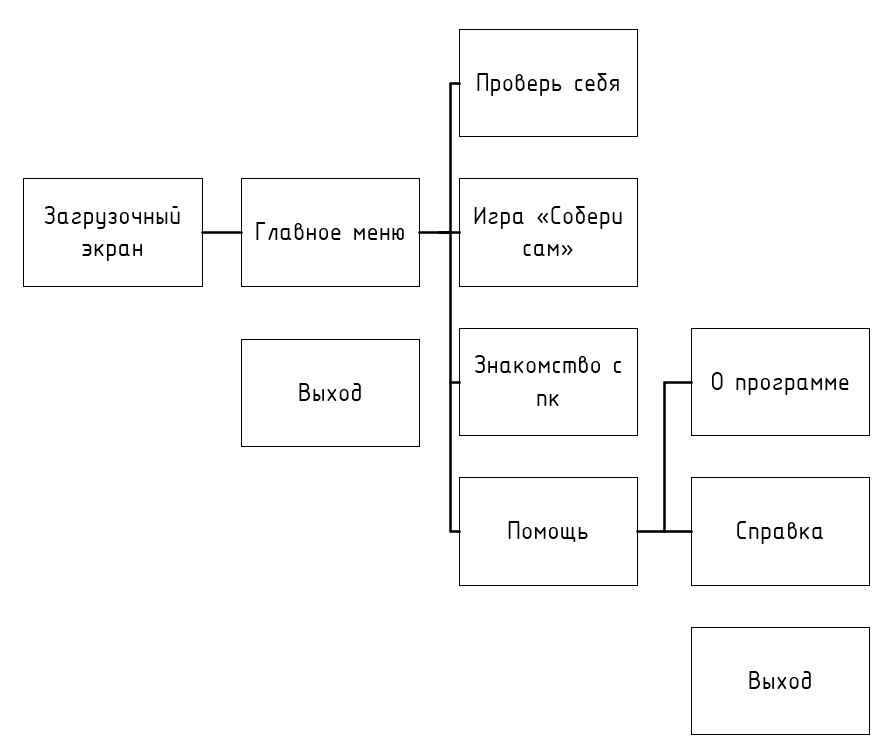


Рисунок 1 – Навигация между окнами программы

**3 Реализация**

**3.1 Структура программы**

Данный курсовой проект содержит 7 модулей . Далее рассмотрим назначение каждого модуля.

Модуль ScreenSaver – отвечает за загрузку приложения, загрузочный экран.

Модуль Computer – главный модуль программы. С его помощью осуществляется навигация по программному продукту. На этом модуле находится переход к справочному материалу и материалу о программе.

Модуль ComputerInfo – отвечает за выбор компонента, о котором нужно узнать информацию.

Модуль InformationInfo – отвечает за вывод текстовой информации про отдельные элементы системного блока персонального компьютера.

Модуль ComputerGame – отвечает за реализацию игры «Собери сам».

Модуль Test – отвечает за тестовую часть программы. Выбор варианта теста, ответ на вопросы, вывод результата.

Модуль AboutProgramm – содержит основную информацию о программе.

**3.2 Структура и описание процедур и функций пользователя**

Описание разработанных процедур находятся в таблице 1.

Таблица 1 – Процедуры и функции

|  |  |  |  |
| --- | --- | --- | --- |
| Имя процедуры (функции) | В каком модуле находится | За каким компонентом закреплена | Назначение |
| 1 | 2 | 3 | 4 |
| procedure TForm6.Timer1Timer(Sender: TObject); | ScreenSaver | Timer | Настройка таймера и полосы прогресса для загрузочного экрана |
| procedure TForm1.FormCreate(Sender: TObject); | Computer | TForm | Установка переменной для загрузочного экрана в значение false |
| procedure TForm1.FormShow(Sender: TObject); | Computer | TForm | Реализация загрузочного экрана |
| procedure TForm1.N2Click(Sender: TObject); | Computer | MainMenu | Открытие справочного материала |
| Продолжение таблицы 1 | | | |
| 1 | 2 | 3 | 4 |
| procedure TForm1.N4Click(Sender: TObject); | Computer | MainMenu | Закрытие формы |
| procedure TForm1.N5Click(Sender: TObject); | Computer | MainMenu | Открытие материала «О программе» |
| procedure TForm1.SpeedButton1Click(Sender: TObject); | Computer | TSpeedButton | Переход к модулю ComputerInfo |
| procedure TForm1.SpeedButton2Click(Sender: TObject); | Computer | TSpeedButton | Закрытие формы |
| procedure TForm1.SpeedButton3Click(Sender: TObject); | Computer | TSpeedButton | Переход к модулю ComputerGame |
| procedure TForm1.SpeedButton4Click(Sender: TObject); | Computer | TSpeedButton | Переход к модулю Test |
| procedure TForm2.Button2Click(Sender: TObject); | ComputerInfo | TButton | Переход к модулю с информацией и приведение свойств всех компонентов в начальное состояние |
| procedure TForm2.FormActivate(Sender: TObject); | ComputerInfo | TForm | Загрузка картинок из файла |
| procedure TForm2.Image3Click(Sender: TObject); | ComputerInfo | TImage | Выход из модуля и приведение свойств всех компонентов в начальное состояние |
| procedure TForm2.Image3MouseEnter(Sender: TObject); | ComputerInfo | TImage | Подсветка кнопки |
| procedure TForm2.Image3MouseLeave(Sender: TObject); | ComputerInfo | TImage | Отключение подсветки кнопки |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| procedure TForm2.PBlockClick(Sender: TObject); | ComputerInfo | TImage | Подсветка блока питания, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.PHardClick(Sender: TObject); | ComputerInfo | TImage | Подсветка жесткого диска, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.PKulerClick(Sender: TObject); | ComputerInfo | TImage | Подсветка кулера, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.PMotherBoardClick(Sender: TObject); | ComputerInfo | TImage | Подсветка материнской платы, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.PNetClick(Sender: TObject); | ComputerInfo | TImage | Подсветка сетевой карты, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.POpticalClick(Sender: TObject); | ComputerInfo | TImage | Подсветка оптического привода, появление краткой информации, загрузка информации в модуль InformationWindow |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| procedure TForm2.PRAMClick(Sender: TObject); | ComputerInfo | TImage | Подсветка оперативной памяти, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.PSoundClick(Sender: TObject); | ComputerInfo | TImage | Подсветка звуковой карты, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.PVideoClick(Sender: TObject); | ComputerInfo | TImage | Подсветка видеокарты, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.RadioGroup1Click(Sender: TObject); | ComputerInfo | TRadioGroup | Выбор нужного компонента, появление краткой информации, загрузка информации в модуль InformationWindow |
| procedure TForm2.Shape1MouseEnter(Sender: TObject); | ComputerInfo | TShape | Подсветка кнопки |
| procedure TForm2.Shape1MouseLeave(Sender: TObject) | ComputerInfo | TShape | Отключение подсветки кнопки |
| procedure TForm3.Image2Click(Sender: TObject); | InformationWindow | TImage | Возврат на прошлую форму |
| procedure TForm3.Image2MouseEnter(Sender: TObject); | InformationWindow | TImage | Подсветка кнопки |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| procedure TForm3.Image2MouseLeave(Sender: TObject); | InformationWindow | TImage | Отключение подсветки кнопки |
| procedure TForm3.Shape1MouseEnter(Sender: TObject); | InformationWindow | TShape | Подсветка кнопки |
| procedure TForm3.Shape1MouseLeave(Sender: TObject); | InformationWindow | TShape | Отключение подсветки кнопки |
| procedure TForm4.BlockPosDragDrop(Sender, Source: TObject; X, Y: Integer); | ComputerGame | TImage | Позиция для перетаскиваемого компонента, проверка правильной установки |
| procedure TForm4.BlockPosDragOver(Sender, Source: TObject; X, Y: Integer;  State: TDragState; var Accept: Boolean); | ComputerGame | TImage | Разрешение перетаскивания объекта |
| procedure TForm4.CPUPosDragDrop(Sender, Source: TObject; X, Y: Integer); | ComputerGame | TImage | Позиция для перетаскиваемого компонента, проверка правильной установки |
| procedure TForm4.CPUPosDragOver(Sender, Source: TObject; X, Y: Integer;  State: TDragState; var Accept: Boolean); | ComputerGame | TImage | Разрешение перетаскивания объекта |
| procedure TForm4.HardPosDragDrop(Sender, Source: TObject; X, Y: Integer); | ComputerGame | TImage | Позиция для перетаскиваемого компонента, проверка правильной установки |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| procedure TForm4.HardPosDragOver(Sender, Source: TObject; X, Y: Integer;  State: TDragState; var Accept: Boolean); | ComputerGame | TImage | Разрешение перетаскивания объекта |
| procedure TForm4.Image1Click(Sender: TObject); | ComputerGame | TImage | Выход из формы |
| procedure TForm4.Image1MouseEnter(Sender: TObject); | ComputerGame | TImage | Подсветка кнопки |
| procedure TForm4.Image1MouseLeave(Sender: TObject); | ComputerGame | TImage | Отключение подсветки кнопки |
| procedure TForm4.KulerPosDragDrop(Sender, Source: TObject; X, Y: Integer); | ComputerGame | TImage | Позиция для перетаскиваемого компонента, проверка правильной установки |
| procedure TForm4.KulerPosDragOver(Sender, Source: TObject; X, Y: Integer;  State: TDragState; var Accept: Boolean); | ComputerGame | TImage | Разрешение перетаскивания объекта |
| procedure TForm4.MotherBoardPosDragDrop(Sender, Source: TObject; X, Y: Integer); | ComputerGame | TImage | Позиция для перетаскиваемого компонента, проверка правильной установки |
| procedure TForm4.MotherBoardPosDragOver(Sender, Source: TObject; X, Y: Integer;  State: TDragState; var Accept: Boolean); | ComputerGame | TImage | Разрешение перетаскивания объекта |
| procedure TForm4.RAMPosDragDrop(Sender, Source: TObject; X, Y: Integer); | ComputerGame | TImage | Позиция для перетаскиваемого компонента, проверка правильной установки |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| procedure TForm4.RAMPosDragOver(Sender, Source: TObject; X, Y: Integer;  State: TDragState; var Accept: Boolean); | ComputerGame | TImage | Разрешение перетаскивания объекта |
| procedure TForm4.Shape14MouseEnter(Sender: TObject); | ComputerGame | TShape | Подсветка кнопки |
| procedure TForm4.Shape14MouseLeave(Sender: TObject); | ComputerGame | TShape | Отключение подсветки кнопки |
| procedure TForm4.SpeedButton1Click(Sender: TObject); | ComputerGame | TSpeedButton | Включение подсказок |
| procedure TForm4.SpeedButton2Click(Sender: TObject); | ComputerGame | TSpeedButton | Проверка установленных компонентов |
| procedure TForm4.VideoPosDragDrop(Sender, Source: TObject; X, Y: Integer); | ComputerGame | TImage | Позиция для перетаскиваемого компонента, проверка правильной установки |
| procedure TForm4.VideoPosDragOver(Sender, Source: TObject; X, Y: Integer;  State: TDragState; var Accept: Boolean); | ComputerGame | TImage | Разрешение перетаскивания объекта |
| procedure TForm5.Image3Click(Sender: TObject); | Test | TImage | Перезапуск теста |
| procedure TForm5.Image4Click(Sender: TObject); | Test | TImage | Выход из теста |
| procedure TForm5.Image4MouseEnter(Sender: TObject); | Test | TImage | Подсветка кнопки |
| procedure TForm5.Image4MouseLeave(Sender: TObject); | Test | TImage | Отключение подсветки кнопки |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| procedure TForm5.Image2Click(Sender: TObject); | Test | TImage | Работа с тестом |
| procedure TForm5.RadioGroup1Click(Sender: TObject); | Test | TRadioGroup | Загрузка теста из файла |
| procedure TForm5.Shape1MouseEnter(Sender: TObject); | Test | TShape | Подсветка кнопки |
| procedure TForm5.Shape1MouseLeave(Sender: TObject); | Test | TShape | Отключение подсветки кнопки |

**3.3 Описание использованных компонентов**

Описание использованных для разработки приложения компонентов приводится в таблице 2.

Таблица 2 – Использованные компоненты

|  |  |  |
| --- | --- | --- |
| Компонент | На какой форме расположен | Назначение |
| 1 | 2 | 3 |
| MainMenu | Computer | Используется для создания главного меню проекта |
| TImage | Computer, ComputerGame, ComputerInfo, ScreenSaver, Test, InformationWindow | Используется как фон, кнопки, отдельные элементы |
| TTimer | ScreenSaver | Используется для загрузочного экрана приложения |
| TProgressBar | ScreenSaver | Используется для создания полосы прогресса в загрузочном экране |
| TSpeedButton | Computer, ComputerGame | Используется для создания «невидимых» кнопок |
| TButton | ComputerInfo, AboutProgramm | Используется для перехода между формами |
| TPanel | ComputerInfo, Test | Используется как фон для текста |

Продолжение таблицы 2

|  |  |  |
| --- | --- | --- |
| 1 | 2 | 3 |
| TLabel | ComputerInfo, InformationWindow, Test | Используется для вывода текста на экран |
| TRadioGroup | ComputerInfo, Test, AboutProgramm | Используется для выбора одного из вариантов, а также для вывода информации |
| TShape | ComputerInfo, ComputerGame, InformationWindow, Test | Используется для создания декоративных кнопок, а также для подсказок в игре «Собери сам» |
| TWebBrowser | InformationWindow | Используется для вывода информации из html файлов |
| TListBox | AboutProgramm | Используется для создания списка |

**3.4 Спецификация программы**

Точное название проекта и его состав приводится в таблице 3.

Таблица 3 – Спецификация программы

|  |  |
| --- | --- |
| Имя файла | Назначение |
| 1 | 2 |
| PCHelper.exe | Исполняемый файл проекта |
| PCHelper.dpr | Связывает все файлы, из которых состоит проект |
| AboutProgramm.pas | Файл программного модуля для формы AboutProgramm |
| Computer.pas | Файл программного модуля для формы Computer |
| ComputerGame.pas | Файл программного модуля для формы ComputerGame |
| InformationWindow.pas | Файл программного модуля для формы InformationWindow |
| Screensaver.pas | Файл программного модуля для формы ScreenSaver |
| Test.pas | Файл программного модуля для формы Test |
| Setup.exe | Установочный файл |
| Справка.chm | Файл справки, содержит помощь по работе с программой |
| AboutProgramm.dfm | Форма с описанием программы и разработчика |
| Computer.dfm | Главная форма с навигацией по приложению |
| ComputerGame.dfm | Форма с игрой «Собери сам» |
| InformationWindow.dfm | Форма с информацией про основные компоненты системного блока |

Продолжение таблицы 3

|  |  |
| --- | --- |
| 1 | 2 |
| Screensaver.dfm | Форма с загрузочным экраном |
| Test.dfm | Форма с тестовой частью |
| \*.html | Файлы с учебным материалом |
| \*.txt | Тестовые файлы |
| \*.png, \*.jpg, \*.bmp | Файлы с картинками для форм |

**4 Тестирование**

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования предоставлен в таблице 4.

Таблица 4 – Отчет о результатах тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| № теста | Тест | Ожидаемый результат | Физический результат |
| Т1 | Проверка входа пользователя в систему | Открытие главного окна | Открытие главного окна |
| Т2 | Проверка кнопки главного меню «Помощь» | Развернутый список с кнопками «О программе», «Выход», «Справка» | Развернутый список с кнопками «О программе», «Выход», «Справка» |
| Т3 | Проверка кнопки «О программе» | Переход на форму с информацией о программе и разработчике | Переход на форму с информацией о программе и разработчике |
| Т4 | Проверка кнопки «Справка» | Открытие справочного файла | Открытие справочного файла |
| Т5 | Проверка кнопки «Выход» | Закрытие приложения | Закрытие приложения |
| Т6 | Проверка кнопки «Знакомство с ПК» | Открытие формы ComputerInfo | Открытие формы ComputerInfo |
| Т7 | Проверка кнопки «Игра «Собери сам»» | Открытие формы ComputerGame | Открытие формы ComputerGame |
| Т8 | Проверка кнопки «Проверь себя» | Открытие формы Test | Открытие формы Test |
| Т9 | Проверка кнопки «Выход» | Закрытие приложения | Закрытие приложения |
| Т10 | Нажатие на компонент системного блока | Подсветка компонента и появление краткой информации на форме | Подсветка компонента и появление краткой информации на форме |
| Т11 | Проверка кнопки «Назад» на форме Computer | Возврат на главную форму | Возврат на главную форму |
| Т12 | Проверка кнопки «Узнать больше» | Переход на форму InformationWindow | Переход на форму InformationWindow |

Продолжение таблицы 4

|  |  |  |  |
| --- | --- | --- | --- |
| Т13 | Проверка переключателя между «Кулер» и «Процессор» | Изменение картинки и текста | Изменение картинки и текста |
| Т14 | Проверка кнопки «Назад» на форме InformationWindow | Возврат на форму ComputerInfo | Возврат на форму ComputerInfo |
| Т15 | Проверка перетаскивания элементов в игре | Установленный элемент | Установленный элемент |
| Т16 | Проверка кнопки «Включить подсказки» | Подсветка установочных мест в системном блоке | Подсветка установочных мест в системном блоке |
| Т17 | Проверка кнопки «Нажми, если собрал» | Появление сообщения «Молодец, ты его собрал!» в случае установки всех деталей; появление сообщения «Не все детали установлены!» в случае установки не всех деталей | Появление сообщения «Молодец, ты его собрал!» в случае установки всех деталей; появление сообщения «Не все детали установлены!» в случае установки не всех деталей |
| Т18 | Проверка кнопки «Назад» на форме ComputerGame | Возврат на главную форму | Возврат на главную форму |
| Т19 | Проверка переключателя между вариантами теста | Появление разных вариантов | Появление разных вариантов |
| Т20 | Проверка кнопки ответа на вопрос | Появление текста «Верно/Неверно», при завершении теста появление результата | Появление текста «Верно/Неверно», при завершении теста появление результата |
| Т21 | Проверка кнопки перезапуска теста | Сброс вариантов сложности и опустошение компонента с вопросами | Сброс вариантов сложности и опустошение компонента с вопросами |
| Т22 | Проверка кнопки «Назад» на форме с тестом | Возврат на главную форму | Возврат на главную форму |
| Т23 | Проверка кнопки «ОК» на форме AboutProgramm | Возврат на главную форму | Возврат на главную форму |

При разработке программного продукта было решено множество проблем, например, не осуществлялась подсветка элементов при нажатии на них. Элементы программы были проверены. Все они работают корректно.

**5 Руководство пользователя**

**5.1 Общие сведения о программном продукте**

Цель данного проекта заключается в знакомстве пользователя с компонентами системного блока персонального компьютера, их функционалом, размещением.

Создаваемый программный продукт будет рассчитан на любого рода пользователя, который интересуется компьютерными технологиями.

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилось на разных классах ЭВМ и работать с данной программой было комфортно. Программа разработана на ПК со следующими характеристиками:

* процессор 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz;
* объем ОЗУ 8гб;
* объем места на HDD 500 гб;
* видеоадаптер NVIDIA GeForce RTX 3060 Laptop GPU;
* ОС – Windows 11.

**5.2 Инсталляция**

Для установки данного программного продукта достаточно запустить файл «Setup.exe». Затем откроется окно установки. Далее нужно проследовать инструкции, выбрать папку для установки и нажать «Установить».

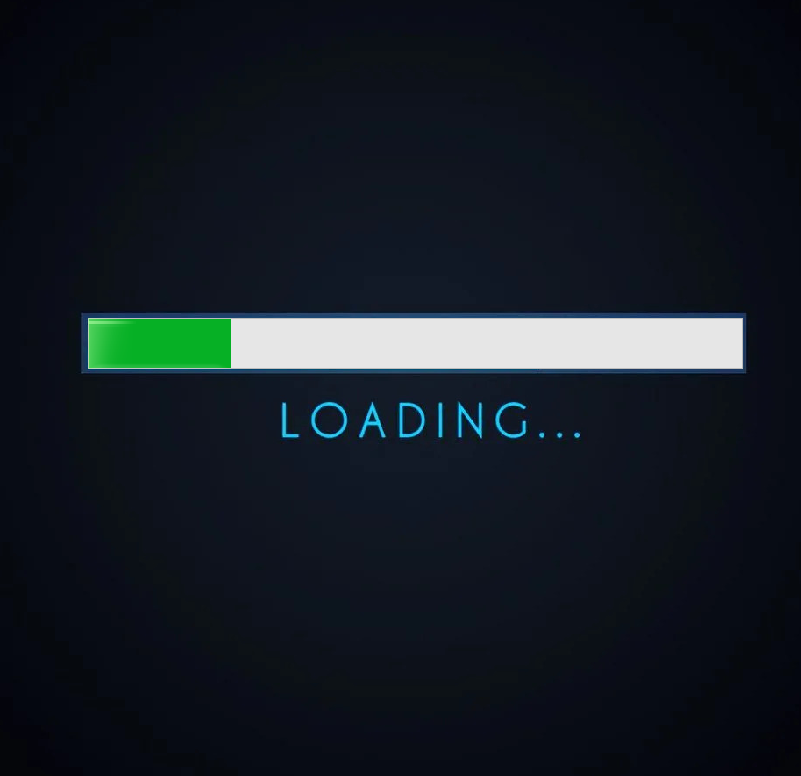
**5.3 Выполнение программы**

Данную программу можно запустить различными способами.

Первым из них является запуск с помощью ярлыка на рабочем столе. Необходимо дважды щелкнуть левой кнопкой мыши на ярлыке с названием “PCHelper”.

Вторым способом является запуск из каталога, в который устанавливалось приложение (по умолчанию C:\Program Files (x86)\Company\PCHelper.exe.

После запуска приложения на экране нас встречает загрузочное окно.

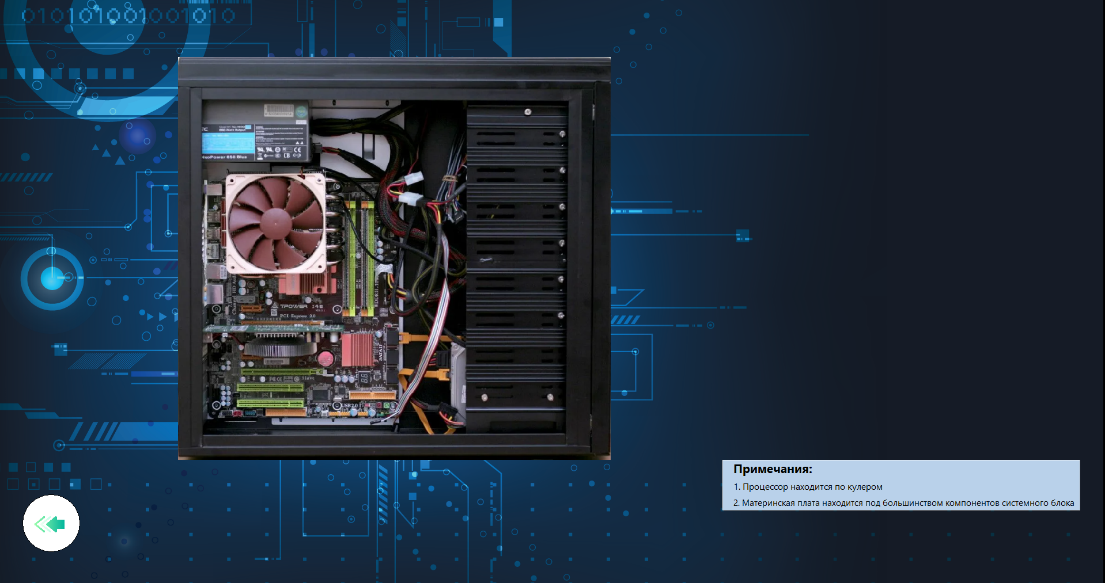
****Рисунок 2 – Загрузочный экран приложения

Далее появляется главное меню программного продукта.



Рисунок 3 – Главное меню

При нажатии на кнопку «Знакомство с ПК» будет осуществлен переход на форму с открытым системным блоком (Рисунок 4), где по нажатию на каждый элемент будет появляться краткая информация о нем и кнопка «Узнать больше» (Рисунок 5).

Рисунок 4 – Открытый системный блок

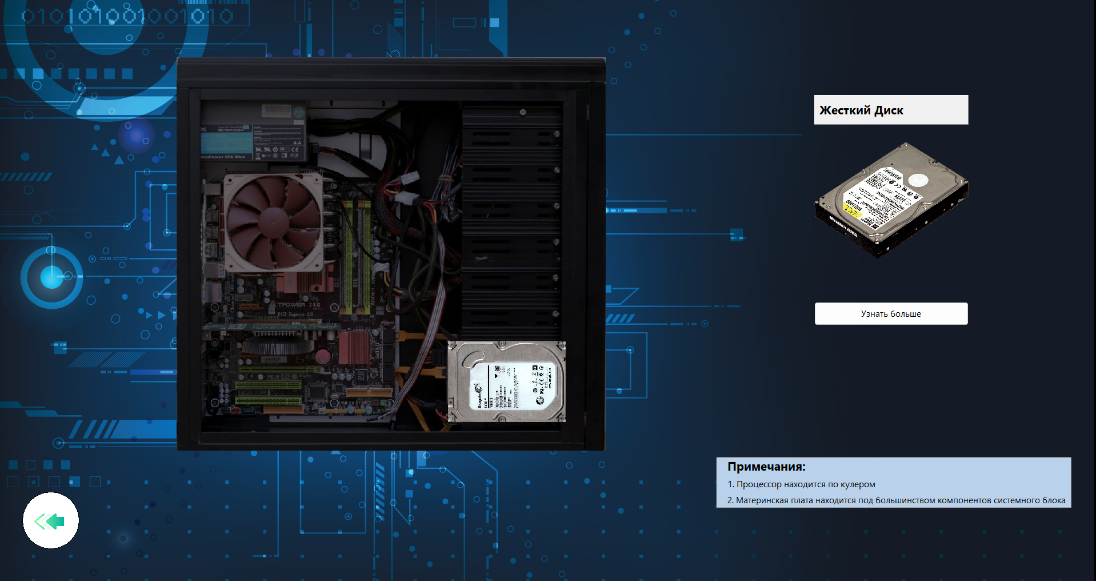


Рисунок 5 – Появление картинки и кнопки

При нажатии на кнопку «Узнать больше» будет осуществлен переход на форму с подробным описанием каждого компонента в отдельности. В зависимости от ранее выбранного компонента откроется соответствующая информация о нем. (Рисунок 6). При желании можно вернуться на предыдущую форму, нажав кнопку со стрелочкой.

Рисунок 6 – Окно с информацией

При нажатии на кнопку «Игра» будет осуществлен переход на форму с игрой «Собери сам». (Рисунок 7). В данном окне пользователь может перетаскивать элементы в системный блок, чтобы установить их на нужное место. При нажатии на кнопку «Включить подсказки» будут подсвечены установочные места в системном блоке. (Рисунок 8). При нажатии на кнопку «Нажми, если собрал» будет выведено сообщение об успехе либо неуспехе поставленной задачи. (Рисунок 9). При желании можно вернуться на главную форму, нажав на кнопку со стрелочкой.

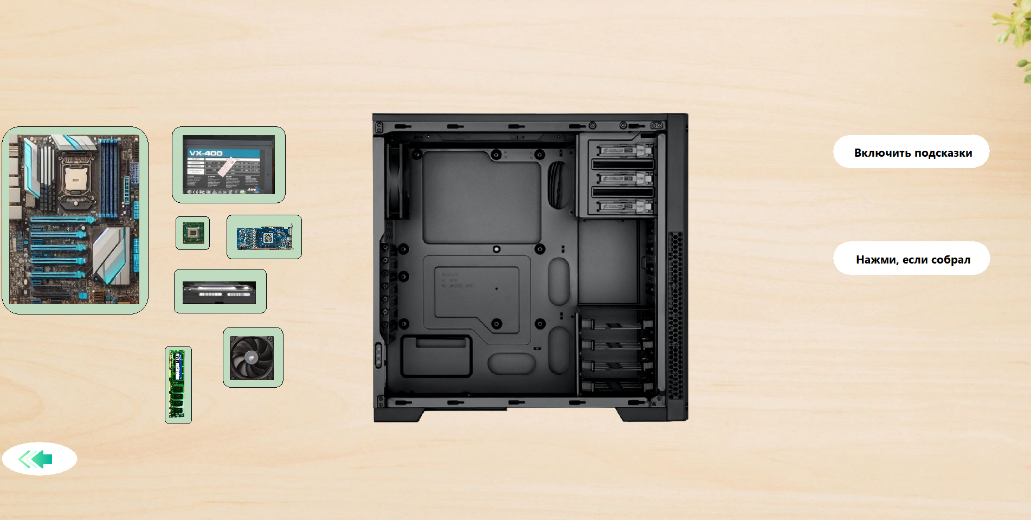


Рисунок 7 – Игра «Собери сам»

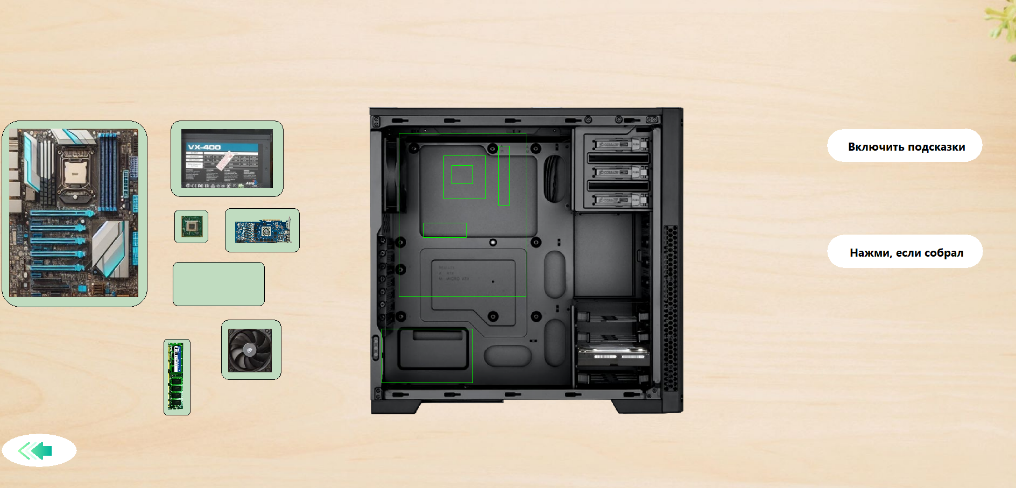
Рисунок 8 – Включение подсказок



Рисунок 9 – Появление сообщения

При нажатии на кнопку «Проверь себя» будет осуществлен переход на форму с тестом. (Рисунок 10). В тесте можно выбрать сложность. (Рисунок 11). После выбора правильного ответа нужно нажать на кнопку с «галочкой». Для перезапуска теста нужно нажать на кнопку с перезагрузкой. После прохождения теста будет выведен результат в процентах. (Рисунок 12).

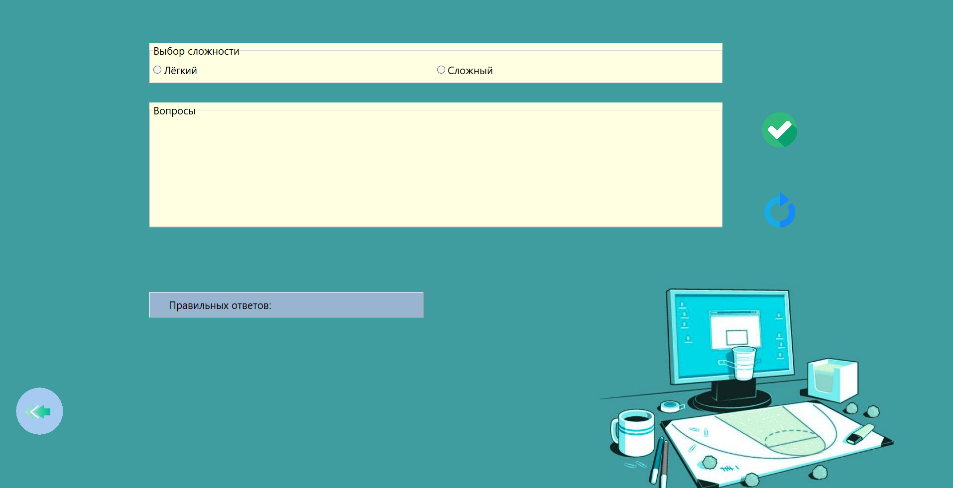
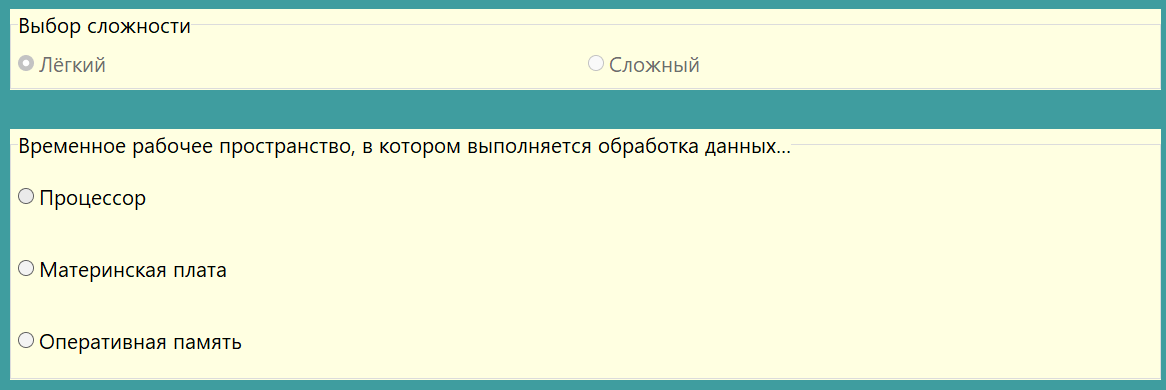


Рисунок 10 – Окно с тестом

Рисунок 11 – Выбор сложности

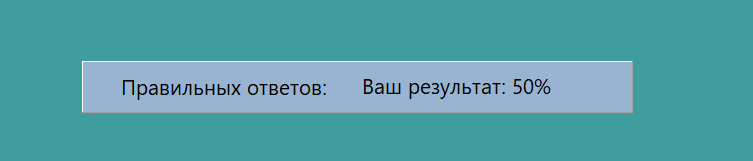


Рисунок 12 – Вывод результата

Также, на главной форме реализована кнопка “Справка”, нажав на которую открывается справочная система. (Рисунок 13).

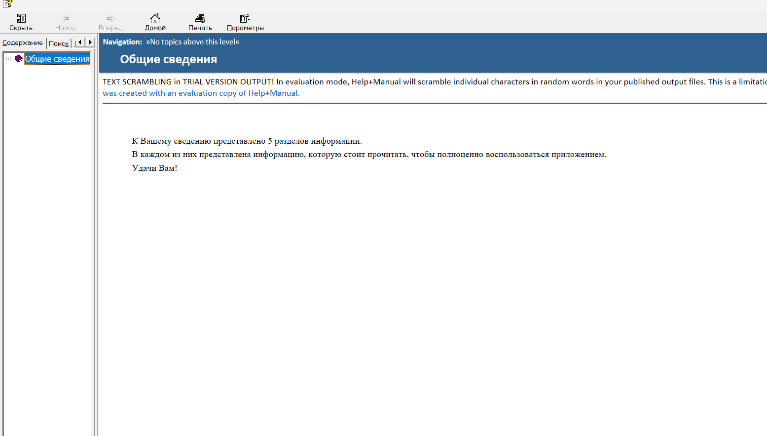
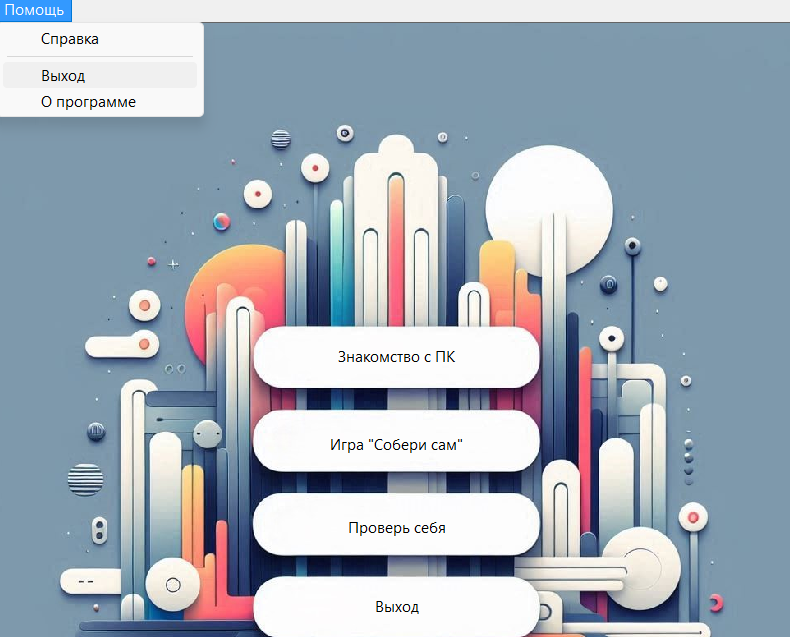


Рисунок 13 – Справка

Завершить работу с программой можно двумя способами:

1. нажав на кнопку «Выход» на главном экране;
2. нажав на пункт «Помощь», далее нажать на пункт «Выход». (Рисунок 14).

Рисунок 14 – Выход из программы

**Заключение**

В данном курсовом проекте была поставлена задача создать программный продукт «Состав ПК».

Задача выполнялось с помощью среды разработки Delphi 11.

Поставленная задача была выполнена успешно. С помощью созданного программного продукта удалось донести до пользователя всю информацию о составе системного блока персонального компьютера.

В дальнейшем эту программу можно развивать, добавляя различные элементы интерфейса, усовершенствуя игру.

Вся информация была взята из достоверных источников. Для зрительного запоминания были использованы картинки с компонентами системного блока. Для усвоения информации о том, как расположены компоненты в системном блоке была создана игра со сборкой. Для усвоения и закрепления тестовой информации был реализован тест с вопросами по каждому из компонентов.

Приложение было разработано для ряда пользователей начиная от людей, не использовавших ПК никогда, до начинающих программистов.

На данном этапе уже можно сказать, что программный продукт полностью готов к использованию.

**Список использованных источников**

1. Учебный курс по программированию «Основы алгоритмизации и программирования»;
2. IT community [Электронный ресурс]: – Электронные данные. – Режим доступа: https://itc.ua. – Дата доступа 24.06.2024;
3. Wondershare [Электронный ресурс]: – Электронные данные. – Режим доступа: https://recoverit.wondershare.com.ru. – Дата доступа 24.06.2024;
4. NurKZ [Электронный ресурс]: – Электронные данные. – Режим доступа: https://www.nur.kz/technologies/device. – Дата доступа 24.06.2024;
5. Dzen [Электронный ресурс]: – Электронные данные. – Режим доступа: https://dzen.ru. – Дата доступа 24.06.2024;
6. Wikipedia [Электронный ресурс]: – Электронные данные. – Режим доступа: https://ru.wikipedia.org/wiki. – Дата доступа 24.06.2024;
7. Sostav\_pc [Электронный ресурс]: – Электронные данные. – Режим доступа: http://kefahelp.h1n.ru/page\_1sovet/case/doc/sostav\_pc\_1.pdf. – Дата доступа 24.06.2024;
8. Derudo [Электронный ресурс]: – Электронные данные. – Режим доступа: https://derudo.ru/ustroystvo\_kompyutera.html. – Дата доступа 24.06.2024.