

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ
ИМ. Б.Л. РОЗИНГА (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)**

Допущен к защите

Зав. отделением

(Подпись) Ю.В. Солодкая
(И.О. Фамилия)

«09» июня 2023 г.

**ДИПЛОМНЫЙ ПРОЕКТ
НА ТЕМУ**

РАЗРАБОТКА ПРИЛОЖЕНИЯ

«КОНФИГУРАТОР СБОРКИ ПК»

Л111. 23ДП00. 020 ПЗ

(Обозначение документа)

Рецензент	_____ (Подпись)	09.06.2023 (Дата)	З.В. Нехлебаев (И.О. Фамилия)
Руководитель	_____ (Подпись)	09.06.2023 (Дата)	Л.С. Хромова (И.О. Фамилия)
Дипломник	_____ (Подпись)	09.06.2023 (Дата)	Н.А. Шефов (И.О. Фамилия)
Консультант по оформлению	_____ (Подпись)	09.06.2023 (Дата)	Ю.В. Солодкая (И.О. Фамилия)
Консультант по экономической части	_____ (Подпись)	09.06.2023 (Дата)	С.Н. Короткова (И.О. Фамилия)

Архангельск 2023

СОДЕРЖАНИЕ

Введение.....	4
1 Анализ и разработка требований	6
1.1 Назначение и область применения.....	6
1.2 Постановка задачи	6
1.3 Описание алгоритма функционирования системы.....	7
1.4 Выбор состава программных и технических средств	7
2 Разработка приложения	10
2.1 Разработка БД.....	10
2.2 Проектирование причинно-следственной диаграммы.....	11
2.3 Проектирование DFD-диаграмм первого и второго уровней.....	12
2.4 Проектирование UML-диаграммы стереотипов и классов.....	14
2.5 Разработка классов приложения.....	15
3 Руководство пользователя	24
3.1 Установка БД.....	24
3.2 Установка приложения.....	24
3.3 Инструкция по работе.....	25
4 Тестирование приложения	29
4.1 Тестирование приложения методом черного ящика	29
5 Определение затрат на разработку приложения	33
6 Охрана труда и техника безопасности при работе с ПК	45
6.1 Общие требования безопасности	45
6.2 Требования безопасности перед началом работы	45
6.3 Требования безопасности во время работы	46
6.4 Требования охраны труда в аварийных ситуациях	46
6.5 Требования охраны труда по окончанию работы.....	47
Заключение	48
Список использованных источников	49
Приложение А	53

Приложение Б.....	67
-------------------	----

ВВЕДЕНИЕ

С повсеместным использованием информационных технологий, появляется необходимость в персональном компьютере. Как правило, приобретение готовой сборки ПК (персональный компьютер) дороже собранной самостоятельно, но при выборе комплектующих можно допустить множество ошибок, из-за которых ПК не будет работать.

В наше время популярностью пользуется сборка компьютера из самостоятельно подобранных комплектующих. Это хорошая возможность подобрать ПК под любые задачи: работа в офисе, игры, обучение, обработка графики и видео.

Производители выпускают сотни моделей материнских плат, процессоров, кулеров, других необходимых элементов стационарного ПК. Поиск оптимальных технических характеристик и совместимости каждой детали может занять много времени. Для многих пользователей сборка ПК – это пазл с головоломками.

Конфигураторы для сборки компьютера можно найти на сайтах онлайн-магазинов, посвящённых компьютерной технике, но в процессе проведения занятий не всегда существует возможность использовать онлайн-сервисы. Поэтому для учебного процесса очень важно иметь приложение «Конфигуратор сборки ПК», с помощью которого будет показан процесс конфигурирования автоматизированного рабочего места с проверкой совместимости компонентов.

Преимущества приложения:

- доступность обучения в любой момент времени,
- возможность сохранения конфигураций в разных форматах,
- возможность фильтрации компонентов,
- оперативность предоставления информации,
- комфортная среда обучения.

Целью дипломного проектирования является разработка приложения «Конфигуратор сборки ПК».

Для достижения поставленной цели требуется выполнить следующие задачи:

- провести анализ предметной области,
- проанализировать возможные подходы к поставленной задаче,
- проанализировать методы решения поставленной задачи с обоснованием выбранного метода,
- выбрать эффективные алгоритмы с учетом их устойчивости и точности,
- спроектировать модели, необходимые для разработки приложения,
- разработать базу данных [6],
- спроектировать интерфейс клиентского приложения,
- разработать клиентское приложение,
- реализовать экспорт данных в формате .xlsx, .docx, pdf,
- провести отладку кода приложения,
- провести тестирование приложения,
- проанализировать полученные в ходе тестирования и отладки результаты работы ПП,
- составить руководство оператора БД (база данных),
- составить руководство пользователя по установке и эксплуатации приложения.

1 Анализ и разработка требований

1.1 Назначение и область применения

Разрабатываемое приложение предназначено для аудиторной и самостоятельной работы обучающихся АКТ (ф) СПбГУТ по изучению раздела учебной практики по Техническому обслуживанию ПК «Подбор конфигурации автоматизированного рабочего места».

Приложение предоставит информацию о компонентах ПК, таких как процессор, материнская плата, корпус, ОП (оперативная память), видеокарта, система охлаждения процессора, блок питания и хранилище данных, функции сортировки, фильтрации и поиска компонентов, конфигурирования комплектующих, сохранения и экспорта сборки ПК.

1.2 Постановка задачи

Требуется спроектировать и разработать БД для хранения информации о комплектующих и приложение, предоставляющее и обрабатывающее информацию о комплектующих.

Создание приложения позволит конфигурировать комплектующих сборки ПК и отображать их характеристики, также экспортировать список комплектующих сборки ПК.

Для достижения этой цели необходимо создать приложение с удобным интерфейсом пользователя и БД.

Приложение должно обеспечивать выполнение следующих задач:

- отображение списков материнских плат, процессоров, корпусов, модулей ОП, систем охлаждения процессора, видеокарт, блоков питания и хранилищ данных,
- поиск комплектующих по имени, фильтрацию и сортировку по цене,

- конфигурирование комплектующих [5],
- экспорта комплектующих сборки ПК формате .pdf, .docx, .xlsx,
- сохранения списков, комплектующих сборки ПК, с возможностью переименовать и удалить [9].

1.3 Описание алгоритма функционирования системы

После запуска приложения перед пользователем отображается главное окно конфигуратора, предоставляющее следующие функции:

- переход к окну справки,
- переход к окну подробной информации о компоненте,
- экспорта сборки ПК в форматы .docx, .xlsx и .pdf,
- создания, изменения наименования и удаления сборки ПК,
- фильтрации, сортировки и поиска компонентов,
- конфигурирование процессора, материнской платы, корпуса, охлаждения процессора, модулей ОП, видеокарты, блока питания и хранилища данных [10].

Конфигурирование комплектующих происходит сразу же при выборе любого компонента и список зависимых комплектующих от выбранного компонента автоматически от фильтруется.

1.4 Выбор состава программных и технических средств

Согласно цели проектирования требуется создать приложение «Конфигуратор сборки ПК» для организации обучающего процесса на базе АКТ (ф) СПбГУТ.

Эксплуатироваться разрабатываемое приложение будет на персональных компьютерах с установленной ОС семейства Windows версии не ниже Windows 7.

В качестве системы управления базами данных выбрана СУБД Microsoft SQL Server 2019 Express, т.к. она является удобной в работе и имеет собственный язык запросов, который оптимален тем, что информацию из БД можно извлекать по любому критерию или совокупности критериев [11].

Приложение будет написано на языке программирования C# [17], т.к. в нем присутствует технология для доступа к данным БД Entity Framework и платформа пользовательского интерфейса Windows Presentation Foundation. Для разработки приложения будет использоваться интегрированная среда разработки программ Microsoft Visual Studio 2022, т.к. она позволяет достаточно быстро создавать приложения на языке программирования C#, проводить тестирование и отладку и создавать установочные файлы.

Для функционирования системы на стороне сервера достаточны следующие программные и технические средства:

- ОС Windows 10 TH1 1507 или выше, либо Windows Server 2016 или выше,
- сервер БД: Microsoft SQL Server версии не ниже 2019 года,
- программное обеспечение для конфигурирования, управления и администрирования MSSQL: SQL Server Management Studio 18 или выше,
- процессор Intel или совместимый процессор с тактовой частотой 1,4 ГГц и выше (рекомендуется 2 ГГц и выше),
- оперативная память минимум 512 МБ (рекомендуется 2 ГБ и выше),
- объем свободного места на жестком диске не менее 6 ГБ,
- манипуляторы: клавиатура и мышь,
- компьютерный монитор: LCD с диагональю не менее 21".

Для функционирования системы на стороне клиента достаточны следующие программные и технические средства:

- операционная система Windows 7 пакетом обновления 1 (SP1) или выше,
- пакет Microsoft Office,
- .Net Framework версии 4.8 и выше,

- процессор Intel или совместимый процессор с тактовой частотой 1 ГГц и выше (рекомендуется 2 ГГц и выше),
- оперативная память минимум 1024 МБ (рекомендуется 2 ГБ и выше),
- объем свободного места на жестком диске не менее 4 ГБ,
- компьютерный монитор: LCD с диагональю не менее 21",
- манипуляторы: клавиатура и мышь.

2 Разработка приложения

2.1 Разработка БД

В БД требуется хранить информацию в БД о процессорах, материнских платах, корпусах, видеокартах, модулях ОП, блоках питания и хранилищах данных [12]. У каждого комплектующего должны быть подробно описаны характеристики для конфигурирования сборки ПК [7].

Для достижения поставленных задач разработана БД, содержащая шестьдесят четыре таблицы [13]. На рисунке 1 показана часть физической модели модели предметной области, разработанная для СУБД Microsoft SQL Server Express 2019.

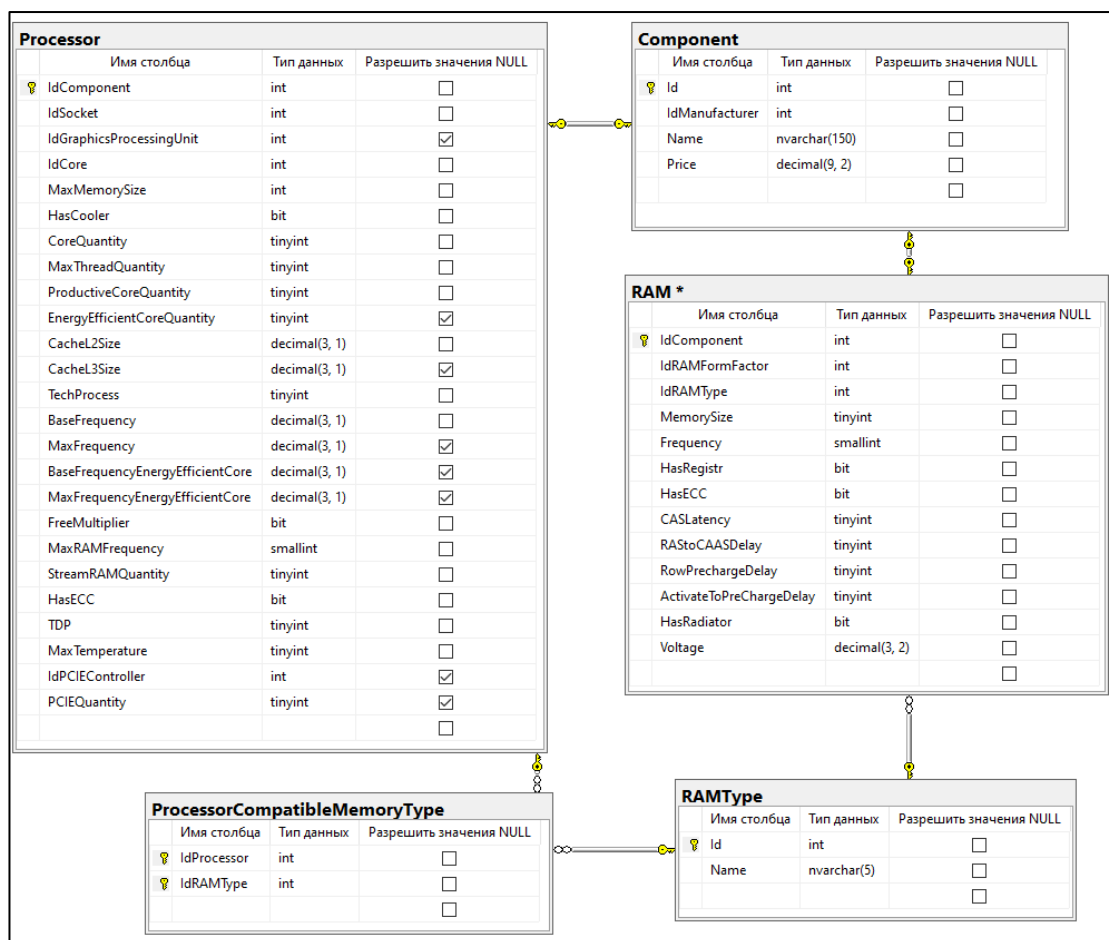


Рисунок 1 – Часть физической модели

Также для редактирования записей в БД разработано руководство оператора БД (Приложение А).

2.2 Проектирование причинно-следственной диаграммы

Причинно-следственная диаграмма – это графический способ, позволяющий выявить наиболее существенные причины, влияющие на конечный результат. Необходимо составить причинно-следственную диаграмму для наглядного отображения функционала разрабатываемого продукта и лучшего понимания исследуемого процесса [15]. На рисунке 2 представлена причинно-следственная диаграмма приложения «Конфигуратор сборки ПК».

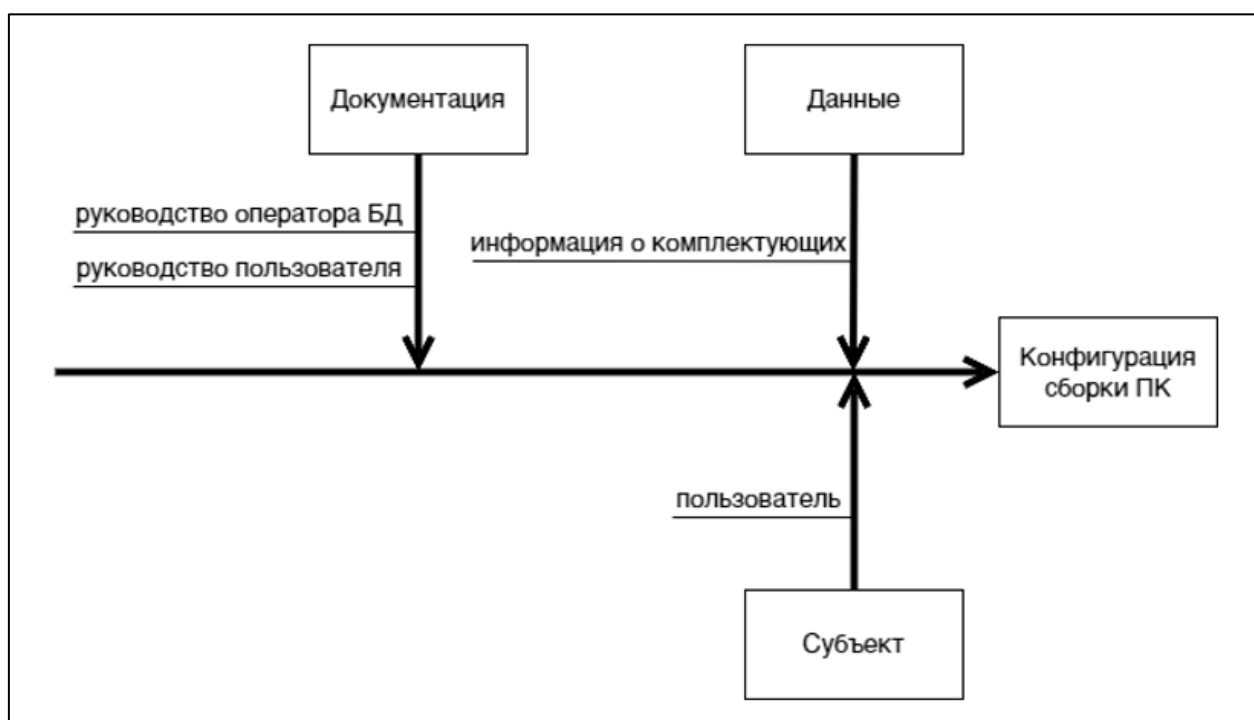


Рисунок 2 – Причинно-следственная диаграмма

Диаграмма показывает, что разрабатываемое приложение предусматривает только пользовательскую роль. Для описания приложения

предусмотрена пояснительная записка, руководство оператора БД и руководство пользователя. Основной проблемой разрабатываемого приложения является изучение конфигурирования сборки ПК. Данная проблема изображена на диаграмме длинной стрелкой, а остальные стрелки, примыкающие к основной, усугубляют проблему.

2.3 Проектирование DFD-диаграмм первого и второго уровней

DFD-диаграммы представляют собой иерархию функциональных процессов, связанных потоками данных. Целью такой диаграммы является продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

В ходе разработки спроектированы DFD-диаграммы первого и второго уровня. DFD-диаграмма первого уровня приложения «Конфигуратор сборки ПК» показана на рисунке 3.

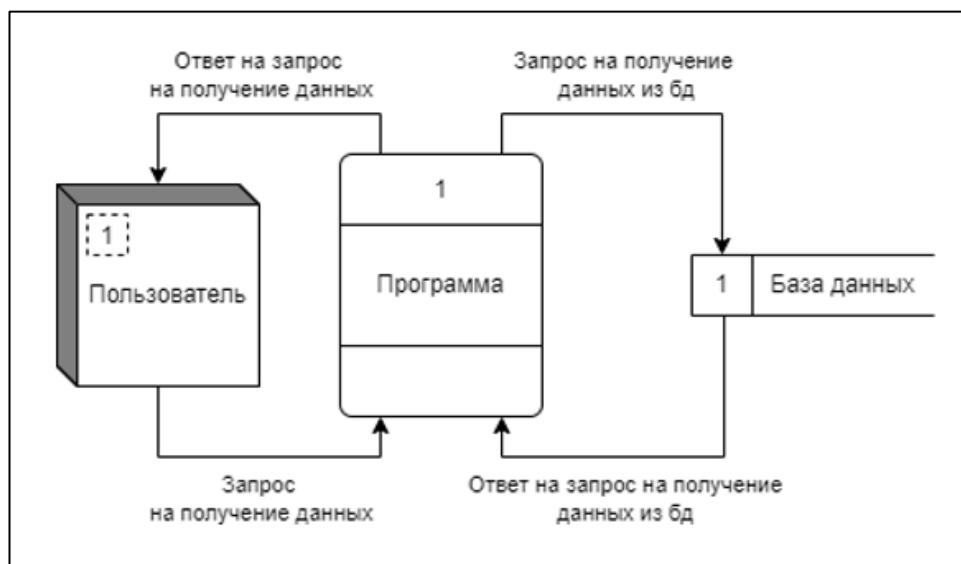


Рисунок 3 – DFD-диаграмма первого уровня

DFD-диаграмма первого уровня показывает, как пользователь взаимодействует с программой, отправляя запрос на получение данных и получая доступ к данным из БД.

DFD-диаграмма второго уровня ПМ «Конфигуратор сборки ПК» показана на рисунке 4.

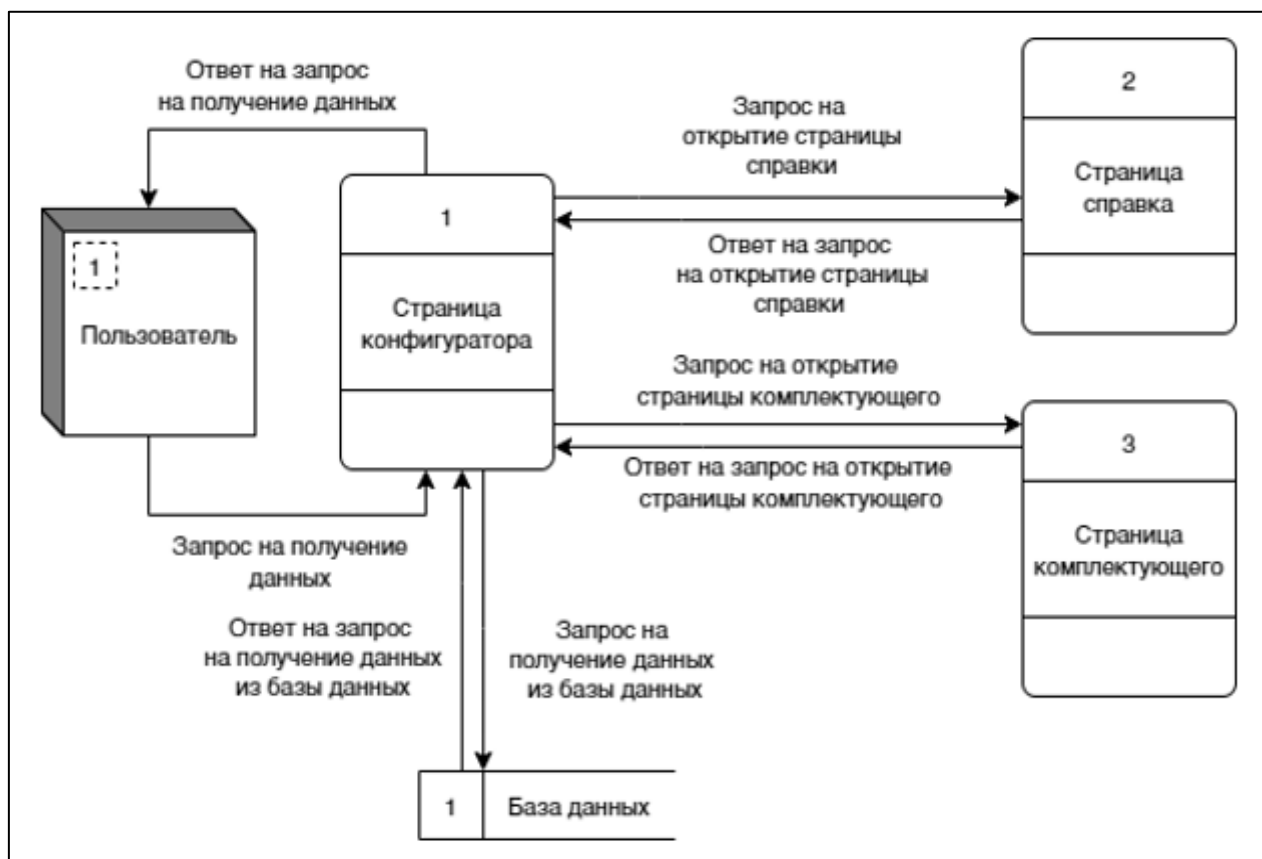


Рисунок 4 – DFD-диаграмма второго уровня

DFD-диаграмма второго уровня демонстрирует взаимодействие пользователя с основными страницами: страницей configurатора, страницей комплектующего, страницей справки.

На странице configurатора пользователь может перейти к страницам комплектующего и справки.

2.4 Проектирование UML-диаграммы стереотипов и классов

UML-унифицированный язык моделирования – это система обозначений, которую можно применять для объектно-ориентированного анализа и проектирования. Его можно использовать для визуализации, спецификации, конструирования и документирования программных систем [8].

Во время работы спроектированы UML-диаграмма стереотипов (рисунок 5). На UML-диаграмме изображены все последовательные действия, которые может совершить пользователь. При открытии программы пользователя встречает главная страница конфигуратора, с которой он может открыть страницу справки и страницу комплектующего.

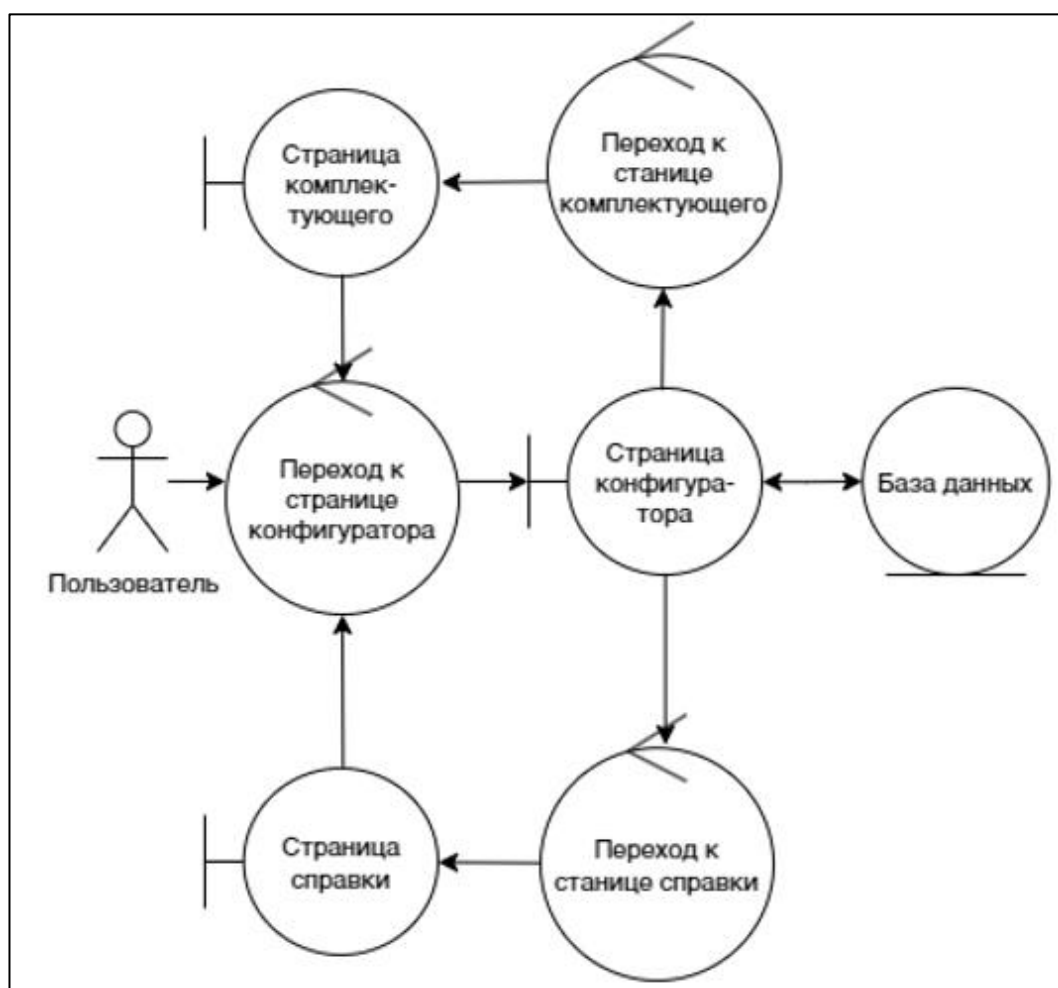


Рисунок 5 – UML-диаграмма стереотипов

2.5 Разработка классов приложения

В ходе дипломного проектирования разработано оконное приложение Windows Foundation Presentation с использованием технологии EntityFramework [3].

Получение данных реализовано через класс DAL (листинг 1), который получает данные таблицы из БД и приводит их к соответствующему классу в приложении.

Листинг 1 – Код класса DAL

```
sealed class DAL
{
    /// <summary>
    /// Поле контекста данных
    /// </summary>
    private static ConfiguratorPCEntities context;

    /// <summary>
    /// Свойство контекста данных
    /// </summary>
    public static ConfiguratorPCEntities Context
    {
        get
        {
            if (context == null)
            {
                //Создание объекта контекста данных
                context = new ConfiguratorPCEntities();
            }
            return context;
        }
    }

    private DAL()
    {
    }
}
```

В приложении используется три страницы для отображения configurатора, характеристик комплектующего и справки [14]. Навигация

между страницами осуществляется при помощи элемента интерфейса Frame в главном окне приложения и статического класса Navigator [14], представлен листингом 2.

Листинг 2 – Код класса Navigator

```
public static class Navigator
{
    //Свойство ссылки на элемент типа Frame
    public static Frame Frame { get; set; }

    //Метод возвращения на предыдущую страницу
    public static void GoBack()
    {
        if (Frame != null && Frame.CanGoBack)
        {
            Frame.GoBack();
        }
    }
}
```

Класс FeedBack разработан для отображения информационных сообщений пользователю, представлен листингом 3

Листинг 3 – Код класса FeedBack

```
public static class FeedBack
{
    /// <summary>
    /// Метод отображения сообщения об ошибке
    /// </summary>
    /// <param name="ex">Исключение</param>
    public static void ShowError(Exception ex)
    {
        //Проверка на исключение БД
        if (ex is EntityException)
        {
            FeedBack.ShowError("Ошибка подключения к базе данных. Обратитесь к системному администратору.");
            //Завершение работы приложения
            Application.Current.Shutdown();
        }
        else
        {

```



```

        ShowError(ex.Message);
    }
}

/// <summary>
/// Метод отображения сообщения об ошибке
/// </summary>
/// <param name="message">Сообщение</param>
public static void ShowError(string message)
{
    new MessageWindow("Ошибка", message, true).ShowDialog();
}

/// <summary>
/// Метод отображения сообщения
/// </summary>
/// <param name="message">Сообщение</param>
public static void ShowMessage(string message)
{
    new MessageWindow("Сообщение", message).ShowDialog();
}
}

```

Для конфигурации сборки ПК разработан класс Configurator [4].
Отрывок кода класса Configurator представлен листингом 4.

Листинг 4 – Отрывок кода класса Configurator

```

[JsonObject(MemberSerialization.OptIn)]
public class Configurator
{
    //Конструктор класса без параметров
    public Configurator() { }

    //Конструктор класса с параметром наименования
    public Configurator(string name)
    {
        this.name = name;
    }

    //Поле наименования конфигурации
    [JsonProperty]
    private string name = "Новая сборка ПК";

    //Свойство для доступа к полю наименования конфигурации
    public string Name
    {

```

```

        get => name;
        set
        {
            name = value;
            ConfiguratorPropertyChanged?
                .Invoke(this, EventArgs.Empty);
        }
    }

    //Поле количества модулей ОП
    [JsonProperty]
    private int ramQuantity = 1;

    //Свойство для доступа к полю количества модулей ОП
    public int RAMQuantity
    {
        get => ramQuantity;
        set
        {
            ramQuantity = value;
            ConfiguratorPropertyChanged?
                .Invoke(this, EventArgs.Empty);
        }
    }

    //Свойство идентификатора процессора
    [JsonProperty]
    public int ProcessorId { get; set; } = -1;

    //Поле процессора
    private Processor processor;

    //Свойство для доступа к полю процессора
    public Processor Processor
    {
        get
        {
            if (processor == null && ProcessorId != -1)
            {
                processor = DAL.Context.Processors
                    .Find(ProcessorId);
            }
            return processor;
        }
        set
        {
            processor = value;
            ProcessorId = processor == null
                ? -1 : processor.IdComponent;
            ProcessorChanged?.Invoke(this, EventArgs.Empty);
            ConfiguratorPropertyChanged?
                .Invoke(this, EventArgs.Empty);
        }
    }

```

```

    }

    //Свойство идентификатора материнской платы
    [JsonProperty]
    public int MotherboardId { get; set; } = -1;

    //Поле материнской платы
    private MotherBoard motherBoard;

    //Свойство для доступа к полю материнской платы
    public MotherBoard MotherBoard
    {
        get
        {
            if (motherBoard == null && MotherboardId != -1)
            {
                motherBoard = DAL.Context.MotherBoards
                    .Find(MotherboardId);
            }
            return motherBoard;
        }
        set
        {
            motherBoard = value;
            MotherboardId = motherBoard == null
                ? -1 : motherBoard.IdComponent;
            MotherBoardChanged?.Invoke(this, EventArgs.Empty);
            ConfiguratorPropertyChanged?
                .Invoke(this, EventArgs.Empty);
        }
    }

    //Список подходящих модулей ОП
    public List<RAM> CompatibleRAMs
    {
        get
        {
            var rams = DAL.Context.RAMs
                .AsNoTracking().ToList();
            if (Processor != null)
            {
                rams = rams
                    .Where(r => Processor.RAMTypes
                        .Any(rt => rt.Id == r.IdRAMType) &&
                        r.MemorySize <= Processor.MaxMemorySize)
                    .ToList();
            }
            if (MotherBoard != null)
            {
                rams = rams
                    .Where(r => MotherBoard.IdRAMType ==
                        r.IdRAMType && MotherBoard.IdRAMFormFactor ==
                        r.IdRAMFormFactor && MotherBoard.MaxRAMSize

```

```

        >= r.MemorySize).ToList();
    }
    return rams;
}
}

```

В приложении реализован экспорт списка комплектующих сборки ПК в форматах .pdf, .docx и .xlsx. Экспорт выполняется локально через диалоговое окно сохранения файла.

Листинг 5 – Код метода Export

```

private void Export()
{
    //Открытие диалогового окна сохранения файла
    SaveFileDialog saveFileDialog = new SaveFileDialog();
    saveFileDialog.Filter = "Документ Word
(docx)|*.docx|Документ PDF (pdf)|*.pdf|Таблица Excel
(xlsx)|*.xlsx";
    if (saveFileDialog.ShowDialog() == DialogResult.OK)
    {
        try
        {
            //Проверка выбранного формата файла экспорта
            var extension = System.IO.Path.
                GetExtension(saveFileDialog.FileName);
            switch (extension)
            {
                case ".docx":
                case ".pdf":
                    var wordApp = new Word.Application();
                    var doc = wordApp.Documents
                        .Add($"{Environment.CurrentDirectory}
                            \\Resources\\template.docx");
                    doc.Content.Find
                        .Execute(FindText: "%commonPrice%",
                            ReplaceWith: $"{currentConfigurator.CommonPrice} руб."
                            , Replace: Word.WdReplace.wdReplaceAll);

                    List<Component> skipList =
                        new List<Component>();
                    foreach (var component in
                        currentConfigurator.Components)
                    {
                        if(skipList.Any(c =>
                            c.Id == component.Id))
                        {
                            continue;
                        }
                    }
                }
            }
        }
    }
}

```

```

    }

    var row = doc.Tables[1].Rows.Add();

    row.Cells[1].Range.ParagraphFormat
        .Alignment = Word.WdParagraphAlignment
            .wdAlignParagraphLeft;

    var sameComponents = currentConfigurator
        .Components.Where(c => c.Id == component.Id).ToList();

    if (sameComponents.Count > 1)
    {
        row.Cells[1].Range.Text =
            $"{component.Name} {sameComponents.Count} шт.";
        row.Cells[2].Range.Text =
            (component.Price * sameComponents.Count).ToString();
        skipList.Add(component);
    }
    else if (component.RAM != null)
    {
        row.Cells[1].Range.Text =
            $"{component.Name} {currentConfigurator.RAMQuantity} шт.";
        row.Cells[2].Range.Text =
            (component.Price * currentConfigurator.RAMQuantity).ToString();
    }
    else
    {
        row.Cells[1].Range.Text = component.Name;
        row.Cells[2].Range.Text = component.Price
            .ToString();
    }
}

doc.Tables[1].Rows[1].Borders[Word.WdBorderType.wdBorderBottom].
LineStyle = Word.WdLineStyle.wdLineStyleDouble;
var wordExtension =
Word.WdSaveFormat.wdFormatDocumentDefault;
if (extension == ".pdf")
    wordExtension =
Word.WdSaveFormat.wdFormatPDF;
//Сохранения файла в Word
doc.SaveAs(saveFileDialog.FileName,
wordExtension);

wordApp.Quit();
break;
case ".xlsx":
    var excelApp = new Excel.Application();
    var workbook = excelApp.Workbooks.Add();
    var sheet = workbook.Worksheets[1];

    sheet.Cells[2][1] = "Конфигурация сборки
ПК";

```

```

        sheet.Cells[2][1].Font.Bold = 1;

        sheet.Cells[1][3] = "№";
        sheet.Cells[2][3] = "Наименование";
        sheet.Cells[3][3] = "Стоимость руб.";
        var titleColumnsRange =
sheet.range[sheet.Cells[1][3], sheet.Cells[3][3]];
        titleColumnsRange.Font.Bold = 1;
        titleColumnsRange.Borders.LineStyle =
Excel.XlLineStyle.xlContinuous;

        skipList = new List<Component>();
        int i = 0;
        int num = 0;
        for (i = 0; i <
currentConfigurator.Components.Count; i++)
        {
            var component =
currentConfigurator.Components[i];

            if (skipList.Any(c => c.Id ==
component.Id))
            {
                continue;
            }

            sheet.Cells[1][num + 4] = num + 1;

            var sameComponents =
currentConfigurator.Components.Where(c => c.Id ==
component.Id).ToList();

            if (sameComponents.Count > 1)
            {
                sheet.Cells[2][num + 4] =
$"{{component.Name}} {{sameComponents.Count}} шт.";
                sheet.Cells[3][num + 4] =
(component.Price * sameComponents.Count).ToString();
                skipList.Add(component);
            }
            else if (component.RAM != null)
            {
                sheet.Cells[2][num + 4] =
$"{{component.Name}} {{currentConfigurator.RAMQuantity}} шт.";
                sheet.Cells[3][num + 4] =
(component.Price * currentConfigurator.RAMQuantity).ToString();
            }
            else
            {
                sheet.Cells[2][num + 4] =
component.Name;
                sheet.Cells[3][num + 4] =
component.Price.ToString();
            }
        }
    }
}

```

```

        }

        var row = sheet.range[sheet.Cells[1][num
+ 5], sheet.Cells[3][num + 4]];
        row.Borders.LineStyle =
Excel.XlLineStyle.xlContinuous;

        num++;
    }

    sheet.Cells[2][num + 4] = "Общая
стоимость:";
    sheet.Cells[3][num + 4] =
currentConfigurator.CommonPrice.ToString();

    sheet.Columns.Autofit();
    //Сохранение файла в Excel
    workbook.SaveAs(saveFileDialog.FileName);
    excelApp.Quit();
    break;
default:
    FeedBack.ShowMessage("Не удалось сохранить
сборку ПК");
    return;
}
    FeedBack.ShowMessage($"Сборка ПК сохранена по пути:
{saveFileDialog.FileName}");
}
    catch (Exception ex)
    {
        FeedBack.ShowError(ex);
    }
}
}

```

3 Руководство пользователя

3.1 Установка БД

Для установки БД необходимо установить сервер и выполнить SQL-скрипт. Для этого необходимо:

- скачать MS SQL Server 2019 Express Editions with Tools,
- выполнить установку сервера БД,
- запустить SQL Server Management Studio,
- войти под пользователем с ролью «dbmanager»
- запустить скрипт «db.sql».

3.2 Установка приложения

Для установки клиентской части приложения необходимо в папке с файлами программы (рисунок 6) запустить программу установки приложения, согласиться с условиями использования приложения, выбрать директорию для установки приложения и дождаться конца установки.

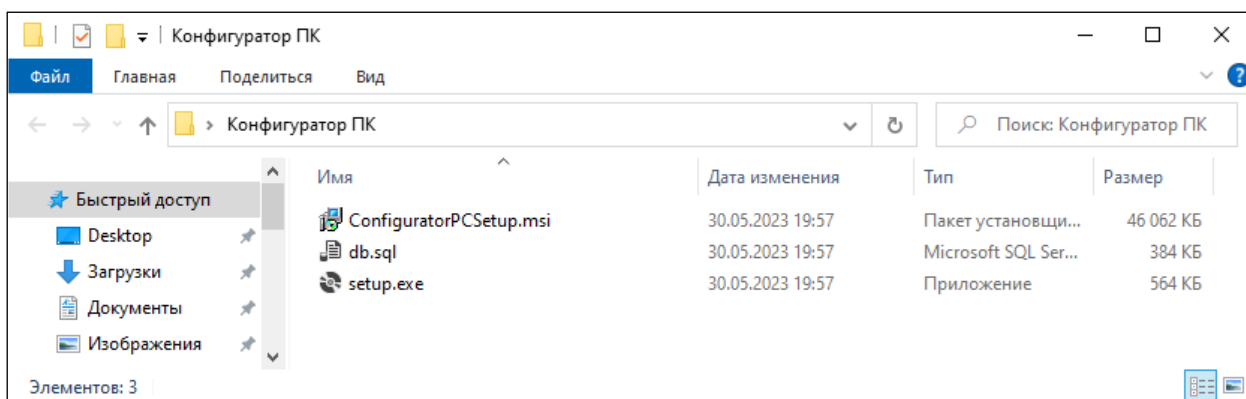


Рисунок 6 – Вид папки с инсталляционным пакетом «Setup»

3.3 Инструкция по работе

Конфигурация компонентов ПК производится автоматически при выборе одного из комплектующих и при выборе следующего будут отображены только совместимые комплектующие. Для начала работы с конфигуратором нажмите кнопку «+ Добавить» у любой ячейки комплектующего (рисунок 7).



Рисунок 7 – Вид ячейки комплектующего

При нажатии на кнопку «+ Добавить» в ячейке комплектующего раскроется список со всеми совместимыми комплектующими. Чтобы добавить комплектующее нажать кнопку «Добавить» у выбранного комплектующего (рисунок 8). После этого комплектующее займет ячейку и будет отображаться в ней, а кнопка «+ Добавить» поменяется на кнопку «- Убрать». Для того чтобы убрать комплектующее из ячейки нажать на кнопку «- Убрать» и тогда ячейка освободится.

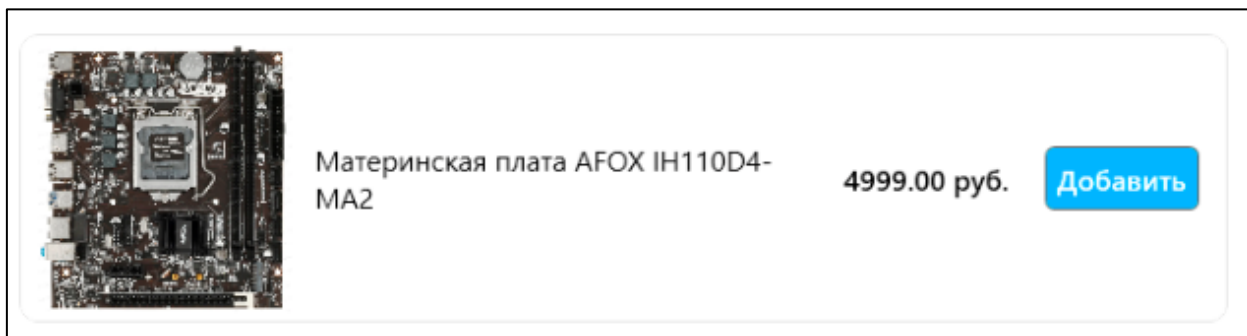


Рисунок 8 – Вид материнской платы в списке

Выбрав материнскую плату, процессор и оперативную память, появится возможность выбрать количество модулей оперативной памяти (рисунок 9), которое ограничено характеристиками выбранных комплектующих. Количество и тип хранилищ данных ограничивается характеристиками выбранных комплектующих и, наоборот, подбираемые комплектующие будут конфигурироваться под выбранные хранилища данных.

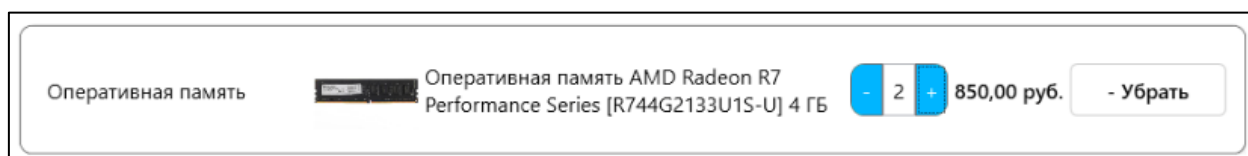


Рисунок 9 – Вид ячейки оперативной памяти

Для того чтобы просмотреть подробные характеристики комплектующего нужно нажать по его наименованию, после чего откроется страница со всеми его характеристиками и изображениями (рисунок 10). Также если ячейка соответствующего комплектующего свободна будет доступна кнопка «Добавить».

Для упрощения поиска комплектующих доступны функции фильтрации и сортировки комплектующих. Для поиска по наименованию воспользуйтесь поисковой строкой, а для использования сортировки или различных фильтров используйте боковое меню с соответствующими функциями (рисунок 11).

Для экспорта сборки ПК в форматах .docx, .pdf и .xlsx нажать на кнопку экспорт. После чего указать имя, путь и расширение файла. Ваша сборка ПК будет экспортирована в указанный файл.

✕ Характеристики Материнская плата AFOX IH110D4-MA2




Общая информация:
 LGA 1151, Intel H110, 2xDDR4-2666 МГц, Mini-DTX

Тип:
 Материнская плата

Производитель:
 AFOX

Стоимость:
 4999,00 руб.

[Добавить](#)

Подробная информация

Форм-фактор и размеры

Форм-фактор Mini-DTX

Рисунок 10 – Вид страницы характеристик материнской платы

Процессор - Свернуть

Сортировка
 Сначала недорогие

Цена
 899,00 × 19699,00 ×

Производитель
 Выбрать все

Сокет
 Выбрать все

Интегрированное графическое ядро
 Выбрать все

Тип памяти

Поиск по наименованию...



	Процессор AMD A6-9500E OEM	899.00 руб.	Добавить
	Процессор Intel Pentium G4400 OEM	2799.00 руб.	Добавить

Рисунок 11 – Вид списка процессоров

Сборка ПК сохраняется автоматически при изменении. Также доступна функция создания, изменения наименования и удаления сборок ПК (рисунок 12).

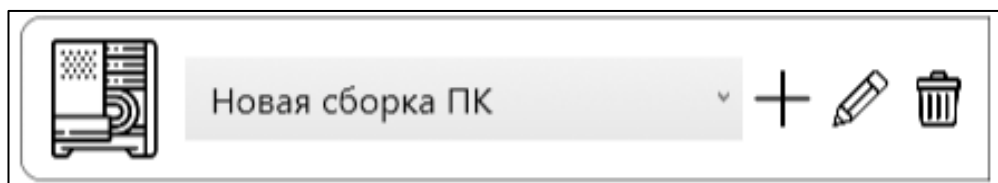


Рисунок 12 – Вид меню управления сборками

4 Тестирование приложения

4.1 Тестирование приложения методом черного ящика

Разработанное приложение протестировано методом черного ящика для выявления ошибок и сбоев [16]. В процессе тестирования проверен весь функционал приложения. Результаты тестирования представлены в таблице 1.

На этапе тестирования выявлена проблема с экспортом списка хранилищ данных. Проблема заключалась в том, что каждое хранилище данных, занимает свою ячейку и при экспорте одинаковые хранилища данных отображались как отдельные компоненты.

После выявления проблемы во время тестирования приняты меры по ее устранению. В итоге был разработан метод обнаружения одинаковых хранилищ данных и записывания их в список с указанием количества без дублирования [20].

Разработанное приложение полностью соответствует всем требованиям, перечисленным в первом разделе, функционирует и готово к эксплуатации [19].

Таблица 1 – Набор тестов и результаты тестирования

Входные данные	Ожидаемый результат	Результат программы
Нажатие на кнопку «Справка»	Открытие окна «Справка»	Совпадает с ожидаемым результатом
Нажатие на кнопку «+ Добавить» у ячейки процессора	Раскрытие списка процессоров	Совпадает с ожидаемым результатом
Нажатие на кнопку «- Свернуть» у ячейки процессора	Сворачивание списка процессоров	Совпадает с ожидаемым результатом

Продолжение таблицы 1

Входные данные	Ожидаемый результат	Результат программы
Выбор пункта сортировки «Сначала дорогие» в списке процессоров	Отображение процессоров по убыванию стоимости	Совпадает с ожидаемым результатом
Ввод текста «Intel» в поисковую строку в списке процессоров	Отображение процессоров, содержащих в наименовании текст «Intel»	Совпадает с ожидаемым результатом
Ввод в поля фильтрации цены 1000 и 10000	Отображение процессоров со стоимостью от 1000 и до 10000 рублей	Совпадает с ожидаемым результатом
Нажатие на кнопку «Добавить» у процессора с сокетом AM4 и нажатие на кнопку «+ Добавить» у ячейки материнской платы	Сворачивание списка процессоров, добавление информации о процессоре в ячейку процессора и раскрытие списка материнских плат с сокетом AM4	Совпадает с ожидаемым результатом
Нажатие на наименование процессора в списке процессоров	Открытие окна характеристик процессора	Совпадает с ожидаемым результатом
Нажатие на наименование процессора в ячейка процессора	Открытие окна характеристик процессора	Совпадает с ожидаемым результатом
Запуск приложения, когда приложение уже запущено	Вывод сообщения: «Приложение уже запущено», без дальнейшего запуска	Совпадает с ожидаемым результатом
Запуск приложения с отсутствующим подключением к БД	Вывод сообщения об ошибке: «Ошибка подключения к базе данных. Обратитесь к системному администратору» и закрытие приложения	Совпадает с ожидаемым результатом

Продолжение таблицы 1

Входные данные	Ожидаемый результат	Результат программы
Разрыв подключения к БД при работе в приложения	Вывод сообщения об ошибке: «Ошибка подключения к базе данных. Обратитесь к системному администратору» и закрытие приложения	Совпадает с ожидаемым результатом
Изменение текущей конфигурации ПК	Автоматическое сохранение конфигурации	Совпадает с ожидаемым результатом
Нажатие на кнопку «Добавить» на верхней панели	Открытие диалогового окна для ввода наименования новой конфигурации ПК	Совпадает с ожидаемым результатом
Нажатие на кнопку «Переименовать» на верхней панели	Открытие диалогового окна для ввода наименования текущей конфигурации ПК	Совпадает с ожидаемым результатом
Нажатие на кнопку «Удалить» на верхней панели	Открытие диалогового окна с сообщением «Вы уверены что хотите удалить сборку?»	Совпадает с ожидаемым результатом
Нажатие на кнопку «Да» в диалоговом окне удаления единственной сборки ПК	Удаление выбранной сборки ПК и создание новой сборки ПК с наименованием «Новая сборка ПК»	Совпадает с ожидаемым результатом
Нажатие на кнопку «Да» в диалоговом окне удаления сборки ПК	Удаление выбранной сборки ПК	Совпадает с ожидаемым результатом
Нажатие на кнопку «Экспорт» и выбор расширения .pdf	Экспорт списка комплектующих текущей сборки ПК в формате .pdf	Совпадает с ожидаемым результатом
Нажатие на кнопку «Экспорт» и выбор расширения .docx	Экспорт списка комплектующих текущей сборки ПК в формате .docx	Совпадает с ожидаемым результатом

Продолжение таблицы 1

Входные данные	Ожидаемый результат	Результат программы
Нажатие на кнопку «Экспорт» и выбор расширения .xlsx	Экспорт списка комплектующих текущей сборки ПК в формате .xlsx	Совпадает с ожидаемым результатом
Выбор хранилища данных	Заполнение ячейки выбранным хранилищем данных и добавление еще одной ячейки хранилища данных	Совпадает с ожидаемым результатом
Нажатие на кнопку «Экспорт» и выбор расширения .xlsx	Экспорт списка комплектующих текущей сборки ПК в формате .xlsx	Совпадает с ожидаемым результатом
Ввод наименования уже существующей сборки ПК в диалоговом окне создания сборки ПК	Вывод сообщения: «Наименование занято другой конфигурацией»	Совпадает с ожидаемым результатом
Нажатие на кнопку «О программе»	Открытие окна «О программе»	Совпадает с ожидаемым результатом
Выбор материнской платы со встроенным процессором и радиатором	Отсутствие компонентов в списках процессоров и охлаждения процессоров	Совпадает с ожидаемым результатом
Выбор другой конфигурации ПК в списке конфигураций	Заполнение ячеек комплектующих из выбранной конфигурации ПК	Совпадает с ожидаемым результатом

По итогам тестирования разработанное приложение функционирует корректно, явных ошибок и сбоев в его работе не выявлено, следовательно, его можно рекомендовать для внедрения и эксплуатации.

5 Определение затрат на разработку приложения

Для расчета экономического обоснования приложения «Конфигуратор сборки ПК» необходимо сделать расчет трудозатрат и денежных средств, затраченных на проект.

Затраты на разработку приложения $Z_{\text{спп}}$, руб., определяются по формуле

$$Z_{\text{спп}} = Z_{\text{спп}}^{\text{МВ}} + Z_{\text{общ}}, \quad (1)$$

где $Z_{\text{спп}}^{\text{МВ}}$ – затраты на оплату машинного времени, руб.;

$Z_{\text{общ}}$ – общие затраты, руб.

Трудоёмкость разработки приложения t , чел.ч, определяется по формуле

$$t = t_o + t_u + t_a + t_b + t_{\text{п}} + t_{\text{от}} + t_{\text{д}}, \quad (2)$$

где t_o – затраты труда на подготовку описания задачи, чел.ч;

t_u – затраты труда на исследование алгоритма решения задачи, чел.ч;

t_a – затраты труда на разработку алгоритма, чел.ч;

t_b – затраты труда на разработку диаграмм алгоритма, чел.ч;

$t_{\text{п}}$ – затраты труда на программирование по готовой диаграмме, чел.ч;

$t_{\text{от}}$ – затраты труда на отладку программы ЭВМ, чел.ч;

$t_{\text{д}}$ – затраты труда на подготовку документации, чел.ч.

Составление затрат вычисляется при помощи условного числа операторов. Условное число операторов Q , ед, в программе определяется по формуле

$$Q=q \cdot c \cdot (1+p), \quad (3)$$

где q – число операторов (исходных команд), ед;

c – коэффициент, учитывающий новизну и сложность программы;

p – коэффициент коррекции программы в ходе разработки, зависит от точности и корректности поставленной задачи (0.05-0.10).

В разработанной программе число операторов составляет около 720 ($q=720$).

Коэффициент, учитывающий новизну и сложность программы, определяется исходя из таблицы Б.1 (приложение Б) на пересечении групп сложности и степени новизны.

Приложение по степени новизны относится к одной из четырёх групп:

- 1) группа А – разработка принципиально новых задач,
- 2) группа Б – разработка оригинальных программ,
- 3) группа В – разработка программ с использованием типовых решений,
- 4) группа Г – разовая типовая задача.

По степени сложности приложение относится к одной из трёх групп:

- 1) алгоритмы оптимизации и моделирования систем,
- 2) задачи учёта, отчётности и статистики,
- 3) стандартные алгоритмы.

Созданное приложение по степени новизны относится к разработке программ с использованием типовых решений (группа В), а по степени сложности алгоритма – к стандартным алгоритмам (группа 3).

По таблице Б.1 (приложение Б) коэффициент $c = 1$ и коэффициент $B = 1,5$.

С учётом того, что задача была поставлена достаточно чётко, коэффициент p принимается равным 0,06 ($p=0,06$).

Условное число операторов, согласно формуле (3), составляет

$$Q=720 \cdot 1 \cdot (1,00+0,06)=763,20 \text{ ед}$$

Затраты труда на подготовку описания задачи t_0 , чел.ч, точно определить невозможно, т.к. это связано с творческим характером работы. С учетом этого можно принять данное значение равным 50 чел.ч ($t_0=50$).

Затраты труда на изучение описания задачи с учётом уточнения описания и квалификации программиста t_u , чел.ч, определяются по формуле

$$t_u=(Q \cdot B)/(75 \cdot 85 \cdot K), \quad (4)$$

где B – коэффициент увеличения затрат труда вследствие недостаточного описания задачи, уточнений и некоторой недоработки;

K – коэффициент квалификации работника.

По таблице Б.2 (приложение Б) для работающих до двух лет $K = 0,80$.

$$t_u=763,20 \cdot 1,50/(80 \cdot 0,80)=17,89 \text{ чел.ч}$$

Далее необходимо вычислить затраты труда на различных стадиях разработки приложения.

Затраты труда на разработку алгоритма решения задачи t_a , чел.ч, определяются по формуле

$$t_a=Q/(50 \dots 75 \cdot K), \quad (5)$$

$$t_a=763,20/(65 \cdot 0,80)=14,68 \text{ чел.ч}$$

Затраты труда на разработку диаграмм решения задачи t_6 , чел.ч, определяются по формуле

$$t_6 = Q / (50 \dots 75 \cdot K), \quad (6)$$

$$t_6 = 763,20 / (65 \cdot 0,80) = 14,68 \text{ чел.ч}$$

Затраты труда на составление программы по готовой диаграмме t_{Π} , чел.ч, определяются по формуле

$$t_{\Pi} = Q / (50 \dots 75 \cdot K), \quad (7)$$

$$t_{\Pi} = 763,20 / (65 \cdot 0,80) = 14,68 \text{ чел.ч}$$

Затраты труда на отладку программы на ЭВМ при комплексной отладке $t_{от}$, чел.ч, определяются по формуле

$$t_{от} = 1,50 \cdot t_{от}^A, \quad (8)$$

где $t_{от}^A$ – затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи, чел.ч.

Затраты труда на отладку программы на ЭВМ при автономной отладке одной задачи $t_{от}^A$, чел.ч, определяются по формуле

$$t_{от}^A = Q / (50 \dots 75 \cdot K), \quad (9)$$

$$t_{от}^A = 763,20 / (65 \cdot 0,80) = 14,68 \text{ чел.ч}$$

Далее требуется рассчитать затраты труда на отладку программы на ЭВМ при комплексной отладке по формуле (8)

$$t_{от} = 1,50 \cdot 14,68 = 22,02 \text{ чел.ч}$$

Затраты труда на подготовку документации по задаче t_d , чел.ч, определяются по формуле

$$t_d = t_{др} + t_{до}, \quad (10)$$

где $t_{др}$ – затраты труда на подготовку материалов рукописи, чел.ч;

$t_{до}$ – затраты на редактирование, печать и оформление документации, чел.ч.

Затраты труда на подготовку материалов рукописи $t_{др}$, чел.ч, определяются по формуле

$$t_{др} = Q / (150 \dots 200 \cdot K), \quad (11)$$

$$t_{др} = 763,20 / (175 \cdot 0,80) = 5,45 \text{ чел.ч}$$

Затраты на редактирование, печать и оформление документации $t_{до}$, чел.ч, определяются по формуле

$$t_{до} = 0,75 \cdot t_{др}, \quad (12)$$

$$t_{до} = 0,75 \cdot 5,45 = 4,09 \text{ чел.ч}$$

Далее требуется рассчитать затраты труда на подготовку документации по задаче по формуле (10)

$$t_d = 5,45 + 4,09 = 9,54 \text{ чел.ч}$$

Трудоёмкость разработки приложения, согласно формуле (2), составляет

$$t=50+17,89+14,68+14,68+14,68+22,02+9,54=143,49 \text{ чел.ч}$$

При шестидневной рабочей неделе и недельной норме 40 часов это составляет 21 рабочий день.

Затраты на оплату машинного времени при отладке программы $З_{спп}^{MB}$, руб., определяются по формуле

$$З_{спп}^{MB} = C_{\text{час}} \cdot t_{\text{ЭВМ}}, \quad (13)$$

где $C_{\text{час}}$ – цена машино–часа арендного времени, руб/ч;

$t_{\text{ЭВМ}}$ – фактическое время отладки программы на ЭВМ, чел.ч.

Фактическое время отладки $t_{\text{ЭВМ}}$, чел.ч, определяется по формуле:

$$t_{\text{ЭВМ}} = t_{\text{п}} + t_{\text{д}} + t_{\text{от}}, \quad (14)$$

$$t_{\text{ЭВМ}} = 14,68 + 9,54 + 22,02 = 46,24 \text{ чел.ч}$$

Цена машино-часа $C_{\text{час}}$, руб/ч, определяется по формуле

$$C_{\text{час}} = З_{\text{ЭВМ}} / T_{\text{ЭВМ}}, \quad (15)$$

где $T_{\text{ЭВМ}}$ – действительный месячный фонд времени ЭВМ, ч.

Действительный месячный фонд времени ЭВМ $T_{\text{ЭВМ}}$, ч, определяется по формуле

$$T_{\text{ЭВМ}} = 6,70 \cdot (K_{\text{д}} - K_{\text{пв}}) - t_{\text{пр}} \cdot 4, \quad (16)$$

где $K_{\text{д}}$ – общее количество дней в месяце;

$K_{\text{пв}}$ – количество праздничных и выходных дней в месяце;

$t_{\text{пр}}$ – время простоя в профилактических работах, ч.

Общее количество дней $K_d = 29$, число праздничных и выходных дней $K_{\text{пв}} = 7$ [18].

Время простоя в профилактических работах определяется как еженедельная профилактика по 4 часа.

Далее требуется рассчитать действительный месячный фонд времени ЭВМ рассчитывается по формуле (16)

$$T_{\text{ЭВМ}} = 6,70 \cdot (29 - 7) - 4 \cdot 4 = 131,40 \text{ ч}$$

Затраты на эксплуатацию ЭВМ $Z_{\text{общ}}$, руб., определяются по формуле

$$Z_{\text{ЭВМ}} = Z_{\text{ам}} + Z_{\text{эл}}, \quad (17)$$

где $Z_{\text{ам}}$ – издержки на амортизацию, руб.;

$Z_{\text{эл}}$ – издержки на электроэнергию, потребляемую ЭВМ, руб..

Компьютер, на котором выполнена разработка программы, приобретён по рыночной цене 95706,00 руб. С учётом того, что рыночная цена компьютера менее 100 тыс. руб., компьютер не является амортизируемым имуществом в соответствии со ст.256-257 НК РФ ч. 2 [1], следовательно $Z_{\text{ам}} = 0$ руб.

Стоимость электроэнергии, потребляемой за месяц, $Z_{\text{эл}}$, руб., определяется по формуле

$$Z_{\text{эл}} = P_{\text{ЭВМ}} \cdot T_{\text{ЭВМ}} \cdot C_{\text{эл}}, \quad (18)$$

где $P_{\text{ЭВМ}}$ – суммарная мощность ЭВМ, кВт;

$C_{\text{эл}}$ – стоимость 1 кВт·ч электроэнергии, руб.

Согласно техническому паспорту ЭВМ, потребление электроэнергии составляет 0,66 кВт. Стоимость электроэнергии в г. Архангельске, где проходила разработка, составляет 10,24 руб/кВт·ч [2].

С учетом этого стоимость электроэнергии, потребляемой за месяц, по формуле (18)

$$З_{эл} = 0,66 \cdot 131,40 \cdot 10,24 = 888,05 \text{ руб.}$$

Затраты на эксплуатацию ЭВМ, согласно формуле (17), составляют

$$З_{ЭВМ} = 0 + 888,05 = 888,05 \text{ руб.}$$

Цена машино-часа, согласно формуле (15), составляет

$$C_{час} = 888,05 / 131,40 = 6,76 \text{ руб/ч}$$

Затраты на оплату машинного времени при отладке программы, согласно формуле (13), составляют

$$З_{спп}^{МВ} = 6,76 \cdot 46,24 = 312,51 \text{ руб.}$$

Общие затраты $З_{общ}$, руб., определяются по формуле

$$З_{общ} = З_{зп} + З_{страх} + З_{ам} + З_{эл} + З_{пр} \quad (19)$$

где $З_{зп}$ – издержки на заработную плату, руб.;

$З_{отч}$ – издержки на отчисления в страховые взносы, руб.;

$З_{пр}$ – издержки на прочие и накладные расходы, руб.

Заработная плата работников $З_{зп}$, руб., определяется по формуле

$$З_{\text{зп}} = \text{оклад} \cdot K_{\text{сев}} \cdot K_{\text{р-н}}, \quad (20)$$

где $K_{\text{сев}}$ – коэффициент, учитывающий северную надбавку для работающих в местах, приравненных к Крайнему Северу;

$K_{\text{р-н}}$ – коэффициент, учитывающий районную надбавку на территориях, приравненных к Крайнему Северу.

Районный коэффициент в г. Архангельске составляет 20% от основной заработной платы, а выплаты за выслугу лет, проработанных на территории, приравненной к территории Крайнего Севера – 50% от основной заработной платы.

Примем оклад программиста равным 17001,32 руб.

Заработная плата работников по формуле (20) составляет

$$З_{\text{зп}} = 17001,32 \cdot (1 + 0,20 + 0,50) = 28902,24 \text{ руб.}$$

Страховые взносы с оплаты труда $З_{\text{страх}}$, руб., определяются по формуле

$$З_{\text{страх}} = (\Phi_{\text{от}} \cdot H_{\text{страх}}) / 100, \quad (21)$$

где $\Phi_{\text{от}}$ – фонд оплаты труда, руб.;

$H_{\text{страх}}$ – размер страховых взносов с оплаты труда, %.

В силу того, что число работников соответствует одному, то $\Phi_{\text{от}}$ можно принять равным $З_{\text{зп}}$ ($\Phi_{\text{от}} = 28902,24$), а $H_{\text{страх}}$ составляют 30% от суммы заработной платы без учета взносов на травматизм в соответствии со ст. 425 НК РФ ч. 2 [1].

Страховые взносы с оплаты труда по формуле (21) составляют

$$З_{\text{страх}} = 28902,24 \cdot 30 / 100 = 8670,67 \text{ руб.}$$

Прочие затраты $Z_{пр}$, руб., принимаются в размере 10% в общей сумме затрат и определяются по формуле

$$Z_{пр} = (Z_{зп} + Z_{страх} + Z_{ам} + Z_{эл}) \cdot 10/90 \quad (22)$$

$$Z_{пр} = (28902,24 + 8670,67 + 0 + 888,05) \cdot 10/90 = 4272,55 \text{ руб.}$$

Общие затраты, согласно формуле (19), составляют

$$Z_{общ} = 28902,24 + 8670,67 + 0 + 888,05 + 4272,55 = 42733,51 \text{ руб.}$$

На основании данных о затратах построена диаграмма затрат (рисунок 13).

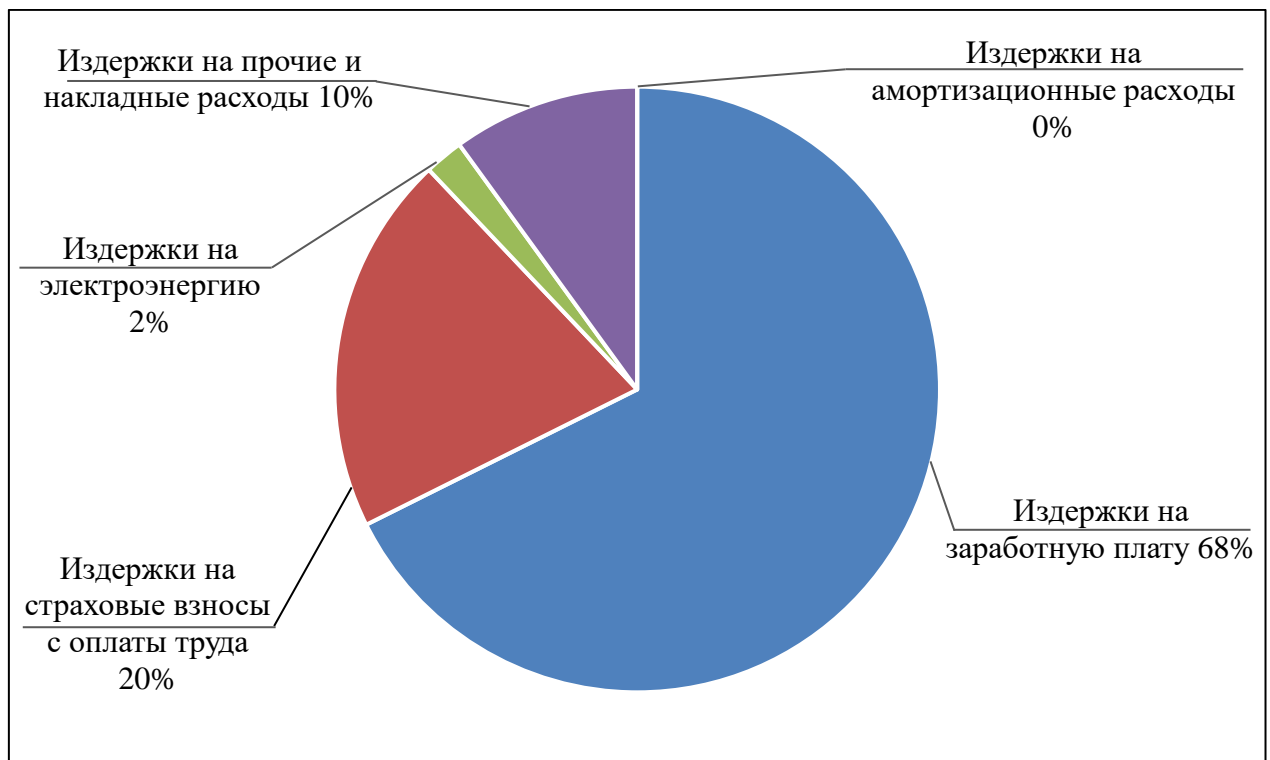


Рисунок 13 – Структура затрат. Диаграмма круговая

На прочие расходы приходится 10% от общих затрат, расходы на электроэнергию занимают 2% от общих затрат, расходы на страховые взносы с оплаты труда – 20%. Амортизационные расходы отсутствуют. Наибольшую долю затрат составляют издержки на заработную плату – 68%, значит процесс разработки приложения является трудоемким.

Затраты на разработку приложения, согласно формуле (1), составляют

$$Z_{\text{сп}} = 312,51 + 42733,51 = 43046,02 \text{ руб.}$$

Во время разработки приложения существуют различные риски, которые могут оказать негативное влияние на разработку. Оценка рисков помогает определить, какие факторы могут повлиять на успешность приложения и как их можно минимизировать или устранить. Кроме того, оценка рисков может помочь избежать непредвиденных ситуаций, которые могут привести к финансовым потерям или даже к неудаче приложения. Возможные риски и способы их минимизации приведены в таблице 2.

Таблица 2 – Возможные риски и способы их минимизации

Риск	Способ минимизации риска
Отключение электроэнергии во время работы за ЭВМ	Использование источника бесперебойного питания
Временная недееспособность разработчика в результате травмы или болезни	Соблюдение правил техники безопасности на рабочем месте и прохождение периодических профилактических медицинских осмотров
Неточность в техническом задании на разработку	Уточнение технического задания на всех этапах разработки, согласование результатов разработки на каждом этапе
Несоответствие системных требований желаниям заказчика	Согласование системных требований, применение технологий, позволяющих понизить системные требования
Изменение сроков разработки	Планирование графика работ, контроль за исполнением графика

Наиболее значимые риски определены и пути их минимизации установлены. Действия по минимизации рисков не приведут к повышению затрат.

В конечном итоге трудоёмкость разработки приложения составила 143,49 чел.ч. В результате выполненных расчётов затраты на разработку приложения составляют 43046,02 руб.

Существует множество аналогов в сети Интернет. Заказчиком предъявлено требование работы приложения в локальной сети, с отсутствующим подключением к сети Интернет. Аналоги, соответствующие требованию, отсутствуют. Разработка приложения обоснована, так как отсутствуют аналоги, соответствующие требованиям заказчика.

6 Охрана труда и техника безопасности при работе с ПК

6.1 Общие требования безопасности

Освещенность на поверхности стола в зоне размещения рабочего документа должна быть 300-500 лк. Освещение не должно составлять бликов на поверхности экрана и превышать 300 лк. В помещении необходимо наличие как искусственных источников освещения, так и естественных.

Рабочее место для работы с ПК должно быть оборудовано следующим:

- клавиатурой, располагающейся на поверхности стола на расстоянии 100-300 мм от края, обращенного к пользователю,
- рабочим столом, имеющим ширину от 800 мм до 1400 мм, глубину от 800 мм до 1000 мм, имеющий пространство для ног с высотой не менее 600 мм, высотой не менее 500 мм, глубиной на уровне колен не менее 450 мм и на уровне вытянутых ног не менее 650 мм,
- рабочим стулом, регулируемым по высоте и углам наклона сиденья и спинки,
- расстояние от глаз до экрана должно быть 600-700 мм, а также угол наклона экрана монитора должен быть 10-15 градусов по отношению к вертикали.

На рабочем месте и в помещении необходимо поддерживать порядок и чистоту, а также проводить систематическое проветривание. В случае аварии нужно прекратить работу до устранения аварийных причин.

6.2 Требования безопасности перед началом работы

Перед началом работы с ПК необходимо выполнить следующее:

- подготовить рабочее место,
- отрегулировать освещение на рабочем месте,

- убедиться в отсутствии бликов на экране,
- проверить провода питания и отсутствие оголенных участков проводов,
- проверить правильность установки стола, стула, подставки для ног, угла наклона экрана, положения клавиатуры, положения «мыши», при необходимости произвести регулировку рабочего стола и кресла, а также расположение элементов компьютера в соответствии с требованиями эргономики и для исключения неудобных поз и длительных напряжений тела.

6.3 Требования безопасности во время работы

Во время работы с ПК запрещается:

- прикасаться к задней панели системного блока при наличии питания,
- переключать разъёмы интерфейсных кабелей периферийных устройств при включенном питании,
- допускать попадание влаги на поверхность системного блока, монитора, рабочую поверхность клавиатуры, принтеров и других устройств,
- производить самостоятельное вскрытие и ремонт оборудования,
- работать на компьютере при снятых кожухах,
- отключать оборудование от электросети и вынимать электровилку, держась за шнур.

6.4 Требования охраны труда в аварийных ситуациях

В случаях обрыва проводов питания, неисправности заземления и других повреждений, появления гари, нужно немедленно отключить питание и сообщить об аварийной ситуации руководителю.

При возникновении пожара, задымлении:

- открыть запасные выходы из здания, обесточить электропитание, закрыть окна и прикрыть двери,
- немедленно сообщить по телефону «112» в пожарную охрану, оповестить работающих, поставить в известность руководителя подразделения, сообщить о возгорании на пост охраны,
- приступить к тушению пожара первичными средствами пожаротушения, если это не сопряжено с риском для жизни,
- организовать встречу пожарной команды,
- покинуть здание и находиться в зоне эвакуации.
- При несчастном случае требуется:
 - немедленно организовать первую помощь пострадавшему и при необходимости доставить его в медицинскую организацию,
 - принять неотложные меры по предотвращению развития аварийной или иной чрезвычайной ситуации и воздействия травмирующих факторов на других лиц,
 - сохранить до начала расследования несчастного случая обстановку, какой она была на момент происшествия, если это не угрожает жизни и здоровью других лиц и не ведёт к катастрофе, аварии или возникновению иных чрезвычайных обстоятельств, а в случае невозможности ее сохранения – зафиксировать сложившуюся обстановку (составить схемы, провести другие мероприятия).

Нельзя приступать к работе до устранения неисправностей.

6.5 Требования охраны труда по окончании работы

По окончании работы с ПК требуется отключить ПК от электросети, отключив тумблеры, а также вытащить вилку из розетки, протереть внешнюю поверхность ПК и прибрать рабочее место.

ЗАКЛЮЧЕНИЕ

В результате проделанной работы по написанию дипломного проекта достигнута поставленная цель в виде разработанного приложение «Конфигуратор сборки ПК».

В ходе проделанной работы разработаны:

- база данных,
- приложение.

Также выполнены следующие поставленные задачи:

- проведен анализ предметной области,
- выбраны эффективные алгоритмы,
- спроектированы модели, необходимые для разработки,
- реализован экспорт данных в форматах .xlsx, .docx и .pdf,
- проведена отладка и тестирование ИС,
- проанализированы полученные в ходе тестирования и отладки результаты работы приложения,
- составлено руководство оператор БД,
- составлено руководство пользователя по установке и эксплуатации приложения.

В процессе разработки задействованы различные информационные ресурсы для расширения теоретических знаний о СУБД Microsoft SQL Server 2019, среде разработки Visual Studio 2022, языке программирования C#, фреймворках Entity Framework и Windows Presentation Foundation.

Приложение является актуальным для тех, кто хочет изучить процесс сборки ПК и собрать собственную конфигурацию ПК.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Налоговый кодекс Российской Федерации. Части I и II. – Москва : ИНФРА-М, 2009. – 608 с. (Библиотека кодексов; Вып. 3 (155)). – URL: <https://znanium.com/catalog/product/189838> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.

2. Об установлении цен (тарифов) на электрическую энергию для населения и приравненных к нему категорий потребителей по Архангельской области. – Текст : электронный // Официальное опубликование правовых актов : [сайт]. – 2023. – URL: <http://publication.pravo.gov.ru/Document/View/2901202211300057> (дата обращения: 19.05.2023).

3. Абрамян, А. В. Разработка пользовательского интерфейса на основе технологии Windows Presentation Foundation : учебник / А. В. Абрамян, М. Э. Абрамян. – Ростов-На-Дону ; Таганрог : Издательство Южного федерального университета, 2017 – URL: <https://znanium.com/catalog/document?id=339526> (дата обращения 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст: электронный.

4. Белугина, С. В. Архитектура компьютерных систем. Курс лекций / С. В. Белугина. – Санкт-Петербург : Лань, 2020. – 160 с. – URL: <https://e.lanbook.com/book/148235> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей.

5. Глушаков, С. В. Компьютер своими руками : учеб. пособие / С. В. Глушаков, А. Н. Шевченко – Москва : Издательский центр «Астрель», 2008.

6. Голицына, О. Л. Базы данных : учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – 4-е изд., перераб. и доп. – Москва : ФОРУМ : ИНФРА-М, 2020. – 400 с. – URL: <https://znanium.com/catalog/document?id=362825> (дата обращения 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

7. Голицына, О. Л. Основы проектирования баз данных : учебное пособие / О. Л. Голицына, Т. Л. Партыка, И. И. Попов. – 2-е изд., перераб. и доп. Москва : ФОРУМ : ИНФРА-М, 2021. – 416 с. – URL: <https://znanium.com/catalog/document?id=364900> (дата обращения 19.05.2023).

– Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

8. Гома, Х. UML. Проектирование систем реального времени, параллельных и распределенных приложений : практическое руководство / Х. Гома. – Москва : ДМК Пресс, 2016. – 700 с. – URL: <https://znanium.com/catalog/product/2012565> (дата обращения: 05.06.2023). –

Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

9. Дадян, Э. Г. Данные: хранение и обработка : учебник / Э. Г. Дадян. – Москва : ИНФРА-М, 2020. – 205 с. – URL: <https://znanium.com/catalog/document?id=346013> (дата обращения 19.05.2023).

– Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

10. Журавлев, А. Е. Организация и архитектура ЭВМ. Вычислительные системы : учебное пособие для спо / А. Е. Журавлев. – 2-е изд., стер. – Санкт-Петербург : Лань, 2021. – 144 с. – URL: <https://e.lanbook.com/book/179036> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей.

11. Кара-Ушанов, В. Ю. SQL – язык реляционных баз данных : учебное пособие / В. Ю. Кара-Ушанов. – Екатеринбург : Изд-во Уральского ун-та, 2016. – URL: <https://znanium.com/catalog/product/1936331> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

12. Компаниец, В. С. Проектирование и юзабилити-исследование пользовательских интерфейсов : учебное пособие / В. С. Компаниец, А. Е. Лызь – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2020. – URL: <https://znanium.com/catalog/product/1894461> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

13. Кравацкий, Ю. Выбор, сборка, апгрейд качественного компьютера : учебное пособие / Ю. Кравацкий, М. Рамендик. – Москва : СОЛОН-Пресс, 2009. – URL: <https://e.lanbook.com/book/13680> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
14. Мартишин, С. А. Базы данных. Практическое применение СУБД SQL и NoSQL-типа для применения проектирования информационных систем : учебное пособие / С. А. Мартишин, В. Л. Симонов, М. В. Храпченко. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2018. – 368 с. – URL: <http://znanium.com/bookread2.php?book=926871> (дата обращения 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
15. Назаров, С. В. Архитектура и проектирование программных систем : монография / С.В. Назаров. – 2-е изд., перераб. и доп. – Москва : ИНФРА-М, 2023. – 374 с. – URL: <https://znanium.com/catalog/product/1895672> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
16. Плаксин, М. А. Тестирование и отладка программ для профессионалов будущих и настоящих : учебное пособие / М. А. Плаксин. – Москва : Лаборатория знаний, 2020. – 170 с. – URL: <https://znanium.com/catalog/product/1987457> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
17. Подбельский, В. В. Язык С#. Базовый курс : учебное пособие / В. В. Подбельский. – Москва : Финансы и статистика, 2022 – URL: <https://znanium.com/catalog/product/1913989> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.
18. Производственный календарь на 2023 год – Текст : электронный // КонсультантПлюс : [сайт]. – 2023. – URL: <https://www.consultant.ru/law/ref/calendar/proizvodstvennye/2023/> (дата обращения: 19.05.2023).

19. Проскуряков, А. В. Качество и тестирование программного обеспечения. Метрология программного обеспечения : учебное пособие / А. В. Проскуряков – Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2022. – 197 с. – URL: <https://znanium.com/catalog/product/2057599> (дата обращения: 19.05.2023). – Режим доступа: для зарегистрир. пользователей. – Текст : электронный.

20. Федорова, Г.Н. Разработка модулей программного обеспечения для компьютерных систем (4–е изд., перераб.) : учебник / Г.Н. Федорова – Москва: Академия. 2020.

Приложение А

(справочное)

Руководство оператора БД

А.1 Редактирование записей таблиц и добавление компонента

Работа с базой данных проводится в Microsoft Sql Server Management Studio. Для редактирования записей таблиц нужно нажать правой кнопкой мыши по таблице и выбрать пункт «Редактировать первые 200 строк». После чего откроется таблица, в которой можно добавлять, изменять и удалять записи.

Добавление любого компонента, как материнская плата или процессор начинается с заполнения записи в таблице Component.

Описание полей в таблице Component:

- Id – идентификатор, автоинкрементное число,
- IdManufacturer – идентификатор производителя из таблицы Manufacturer,
- Name – наименование,
- Price – стоимость (руб.).
- Описание полей в таблицы Manufacturer:
 - Id – идентификатор, автоинкрементное число,
 - Name – наименование.

В базе данных присутствует множество таблицы на подобии Manufacturer, которые имеют одно поле наименование, поэтому их описание полей будет опускаться.

A.2 Добавление процессора

Заполнить запись в таблице Processor.

Описание полей в таблице Processor:

- IdComponent – идентификатор компонента из таблицы Component,
- IdSocket – идентификатор сокета из таблицы Socket,
- IdGraphicsProcessingUnit – идентификатор встроенного графического ядра из таблицы GraphicsProcessingUnit, необязательно,
- IdCore – идентификатор ядра из таблицы Core,
- MaxMemorySize – максимальный поддерживаемый объем оперативной памяти (ГБ),
- HasCooler – наличие кулера, 1 – присутствует, 0 – отсутствует,
- CoreQuantity – количество ядер,
- MaxThreadQuantity – максимальное количество потоков,
- ProductiveCoreQuantity – количество производительных ядер,
- EnergyEfficientCoreQuantity – количество энергоэффективных ядер, необязательно,
- CacheL2Size – объем L2 кэша (МБ),
- CacheL3Size – объем L3 кэша (МБ), необязательно,
- TechProcess – техпроцесс (нм),
- BaseFrequency – базовая частота (ГГц),
- MaxFrequency – максимальная частота в турбо режиме (ГГц), необязательно,
- BaseFrequencyEnergyEfficientCore – базовая частота энергоэффективных ядер (ГГц), необязательно,
- MaxFrequencyEnergyEfficientCore – частота в турбо режиме энергоэффективных ядер (ГГц), необязательно,
- FreeMultiplier – свободный множитель, 1 – присутствует, 0 – отсутствует,

- MaxRAMFrequency – максимальная частота оперативной памяти (МГц),
- StreamRAMQuantity – количество каналов оперативной памяти,
- HasECC – поддержка режима ECC, 1 – присутствует, 0 – отсутствует,
- TDP – тепловыделение (Вт),
- MaxTemperature – максимальная температура процессора (°C),
- IdPCIEController – идентификатор контроллера PCI Express из таблицы PCIEController,
- PCIEQuantity – число линий PCI Express.

Описание полей в таблице GraphicsProcessingUnit:

- Id – идентификатор, автоинкрементное число,
- Name – наименование,
- MaxFrequency – максимальная частота (МГц),
- ExecutiveUnitQuantity – количество исполнительных блоков,
- ShadingUnitsQuantity – количество потоковых процессоров.

Заполнить поддерживаемые процессором типы оперативной памяти. Добавить в таблицу таблицу ProcessorCompatibleMemoryType столько записей сколько процессор поддерживает типов оперативной памяти. Указать IdProcessor (идентификатор компонента процессора) и IdRAMType (идентификатор типа оперативной памяти из таблицы RAMType).

А.3 Добавление материнской платы

Заполнить запись в таблице MotherBoard.

Описание полей в таблице MotherBoard:

- IdComponent – идентификатор компонента из таблицы Component,
- IdMotherBoardFormFactor – идентификатор форм-фактора материнской платы из таблицы MotherBoardFormFactor,
- IdSocket – идентификатор сокета из таблицы Socket,

- IdRAMType – идентификатор типа оперативной памяти из таблицы RAMType,
- IdRAMFormFactor – идентификатор форм-фактора оперативной памяти из таблицы RAMFormFactor,
- MaxRAMSize – максимальный поддерживаемый объем оперативной памяти (ГБ),
- RAMQuantity – количество разъемов для модулей оперативной памяти,
- PCIEx16Quantity – количество разъемов PCI-E x16,
- SATAQuantity – количество разъемов SATA,
- M2Quantity – количество разъемов M.2,
- Height – высота (мм),
- Width – ширина (мм),
- IdChipset – идентификатор чипсета из таблицы Chipset,
- MaxRAMFrequency – максимальная частота оперативной памяти (МГц),
- IdPCIControllerVersion – идентификатор версии контроллера PCI из таблицы PCIController, необязательно,
- RJ45Quantity – количество разъемов RJ-45,
- AnalogAudioOutputQuantity – количество аналоговых аудио выходов,
- CoolerPowerSupply – разъем подключения питания кулера, необязательно,
- M2KeyE – наличие разъема M.2 с ключом E, 1 – присутствует, 0 – отсутствует,
- InterfaceLPT – наличие параллельного порта LPT, 1 – присутствует, 0 – отсутствует,
- SoundSchema – звуковая схема, необязательно,
- IdSoundAdapterChipset – идентификатор чипсета звукового адаптера из таблицы SoundAdapterChipset, необязательно,

- IdNetworkAdapterChipset – идентификатор чипсета сетевого адаптера из таблицы NetworkAdapterChipset, необязательно,
- NetworkAdapterSpeed – скорость сетевого адаптера (Гбит/с), необязательно,
- HasWiFi – наличие встроенного адаптера WiFi, 1 – присутствует, 0 – отсутствует,
- HasBluetooth – наличие встроенного адаптера Bluetooth, 1 – присутствует, 0 – отсутствует,
- PowerPhaseQuantity – количество фаз питания,
- IdMotherBoardPowerPlug – идентификатор основного разъема питания из таблицы MotherBoardPowerPlug,
- IdProcessorPowerPlug – идентификатор разъема питания процессора из таблицы ProcessorPowerPlug, необязательно,
- StreamRAMQuantity – количество каналов памяти,
- EmbeddedProcessor – модель встроенного процессора, необязательно.

Заполнить поддерживаемые материнской платой ядра процессоров в таблице MotherBoardCompatibleCore. Добавить в таблицу столько записей сколько ядер поддерживает материнская плата. Указать IdMotherBoard (идентификатор компонента материнской платы) и IdCore (идентификатор ядра из таблицы Core).

Заполнить разъемы задней панели материнской платы в таблице MotherBoardConnector. Указать IdMotherBoard (идентификатор компонента материнской платы), IdConnector (идентификатор разъема из таблицы Connector) и Quantity (количество разъемов).

Заполнить информацию о разъемах M.2 в таблице MotherBoardM2Key. Указать IdMotherBoard (идентификатор компонента материнской платы), IdFormFactor (идентификатор форм-фактора ключа M.2 из таблицы M2FormFactor) и IdKey (идентификатор типа ключа M.2 из таблицы M2Key).

Заполнить видеовыходы материнской платы в таблице MotherBoardVideoOutput. Указать IdMotherBoard (идентификатор компонента

материнской платы) и IdVideoOutput (идентификатор видеовыхода из таблицы VideoOutput).

A.4 Добавление корпуса

Заполнить запись в таблице Case.

Описание полей в таблице Case:

- IdComponent – идентификатор компонента из таблицы Component,
- IdCaseSize – идентификатор типоразмера из таблицы CaseSize,
- IdPowerSupplyFormFactor – идентификатор форм-фактора блока питания из таблицы PowerSupplyFormFactor,
- ExpansionSlotsQuantity – количество слотов расширения,
- MaxVideoCardLength – максимальная длина видеокарты (мм),
- MaxCoolerHeight – максимальная высота кулера (мм),
- LiquidCoolerCompatible – возможность установки жидкостного охлаждения, 1 – присутствует, 0 – отсутствует,
- Storage35Quantity – количество слотов для накопителей форм-фактора 3.5”,
- Storage25Quantity – количество слотов для накопителей форм-фактора 2.5”,
- MotherBoardOrientation – ориентация материнской платы,
- Length – длина (мм),
- Width – ширина (мм),
- Height – высота (мм),
- IdMainColor – идентификатор цвета из таблицы Color,
- HasWindow – наличие бокового окна, 1 – присутствует, 0 – отсутствует,
- IdLightingType – идентификатор типа подсветки из таблицы LightingType, необязательно,
- PowerSupplyOrientation – размещение блока питания,

- HasCardReader – наличие кард-ридера, 1 – присутствует, 0 – отсутствует.

Заполнить поддерживаемые корпусом форм-факторы материнских плат в таблице CaseCompatibleMotherBoardFormFactor. Добавить в таблицу столько записей сколько форм-факторов материнских плат поддерживает корпус. Указать IdCase (идентификатор компонента корпуса) и IdMotherBoardFormFactor (идентификатор форм-фактора материнской платы из таблицы MotherBoardFormFactor).

Заполнить монтажные размеры радиатора жидкостной системы охлаждения в таблице CaseRadiatorSize, если корпус поддерживает жидкостную систему охлаждения. Указать IdCase (идентификатор компонента корпуса) и IdRadiatorSize (идентификатор монтажного размера радиатора жидкостной системы охлаждения из таблицы RadiatorSize).

Заполнить разъемы задней панели корпуса в таблице CaseConnector. Указать IdCase (идентификатор компонента корпуса), IdConnector (идентификатор разъема из таблицы Connector) и Quantity (количество разъемов).

Заполнить информацию о материалах корпуса в таблице CaseMaterial и материалах передней панели корпуса CaseFrontPanelMaterial. Указать IdCase (идентификатор компонента корпуса) и IdMaterial (идентификатор материала из таблицы Material).

A.5 Добавление видеокарты

Заполнить запись в таблице VideoCard.

Описание полей в таблице VideoCard:

- IdComponent – идентификатор компонента из таблицы Component,
- Length– длина (мм),
- IdGraphicProcessor – идентификатор графического процессора из таблицы GraphicProcessor,

- IdMicroarchitecture – идентификатор микроархитектуры из таблицы Microarchitecture,
- TechProcess– техпроцесс (нм),
- VideoMemorySize– объем видеопамяти (ГБ),
- IdVideoMemoryType – идентификатор типа видеопамяти из таблицы VideoMemoryType,
- MemoryBusBitRate – разрядность шины памяти,
- MaxMemoryBandwidth – максимальная пропускная способность памяти (Гбит/с),
- EffectiveMemoryFrequency – эффективная частота памяти (МГц),
- VideoChipFrequency – штатная частота видеочипа (МГц),
- ALUQuantity – количество универсальных процессоров (ALU),
- TextureBlockQuantity – число текстурных блоков,
- RasterizationBlockQuantity – число блоков растеризации,
- RayTracingSupport – поддержка трассировки лучей,
- MaxMonitorQuantity – максимальное число подключенных мониторов,
- IdPCIEController – идентификатор интерфейса подключения из таблицы PCIEController,
- PowerSupply – рекомендуемый блок питания (Вт),
- CoolerType – тип охлаждения,
- FanType – тип вентилятора, необязательно,
- FanQuantity – количество вентиляторов, необязательно,
- ExpansionSlotSize – количество занимаемых слотов расширения,
- Thickness – толщина (мм),
- Mass – масса (г), необязательно,
- IdVideoCardPowerPlug – идентификатор разъема дополнительного питания из таблицы VideoCardPowerPlug.

Заполнить видеовыходы видеокарты в таблице VideoCardVideoOutput. Добавить в таблицу столько записей сколько видеовыходов имеет

видеокарта. Указать IdVideoCard (идентификатор компонента видеокарты), IdVideoOutput (идентификатор видеовыхода из таблицы VideoOutput) и Quantity (количество).

A.6 Добавление охлаждения процессора

Чтобы добавить кулер или систему жидкостного охлаждения сначала требуется заполнить запись в таблице ProcessorCooler.

Описание полей в таблице ProcessorCooler:

- IdComponent – идентификатор компонента из таблицы Component,
- IdRadiatorMaterial – идентификатор материала радиатора из таблицы Material,
- FanQuantity – количество вентиляторов,
- FanSize – размер вентиляторов,
- FanConnector – разъем подключения вентиляторов,
- MaxRotationSpeed – максимальная скорость вращения (об/мин),
- MinRotationSpeed – минимальная скорость вращения (об/мин),
- AdjustmentRotationSpeed – регулировка скорости вращения, необязательно,
- MaxNoiseLevel – максимальный уровень шума (дБ), необязательно,
- MaxAirflow – максимальный воздушный поток (CFM), необязательно,
- MaxStaticPressure – максимальное статическое давление (Па), необязательно,
- BearingType – тип подшипника вентилятора,
- TDP – рассеиваемая мощность (Вт).

Заполнить поддерживаемые сокеты охлаждения процессора в таблице CoolerCompatibleSocket. Добавить в таблицу столько записей сколько сокетов поддерживает охлаждение процессора. Указать IdProcessorCooler

(идентификатор компонента охлаждения процессора) и IdSocket (идентификатор сокета из таблицы Socket).

A.6.1 Добавление кулера

Заполнить запись в таблице Cooler.

- Описание полей в таблице Cooler:
- IdProcessorCooler – идентификатор компонента из таблицы ProcessorCooler,
- Height – высота (мм),
- ConstructionType – тип конструкции,
- IdBaseMaterial – идентификатор материала основания из таблицы Material,
- TermPipeQuantity – количество тепловых трубок, обязательно,
- TermPipeDiameter – диаметр тепловых трубок, обязательно,
- NickelCoating – никелированное покрытие, обязательно,
- Width – ширина (мм),
- Length – длина (мм).

A.6.2 Добавление системы жидкостного охлаждения

Заполнить запись в таблице LiquidCooler.

Описание полей в таблице LiquidCooler:

- IdProcessorCooler – идентификатор компонента из таблицы ProcessorCooler,
- Serviced – обслуживаемая, 1 – да, 0 – нет,
- IdWaterblockMaterial – идентификатор материала водоблока из таблицы Material,
- WaterblockSize – размер водоблока,

- IdRadiatorSize – идентификатор размера радиатора из таблицы RadiatorSize,

- RadiatorLength – длина радиатора (мм),
- RadiatorWidth – ширина радиатора (мм),
- RadiatorThickness – толщина радиатора (мм),
- PumpRotationSpeed – скорость вращения помпы (об/мин),
- PumpConnector – разъем подключения помпы,
- PipeLength – длина трубок (мм),
- TransparentPipe – прозрачные трубки, 1 – да, 0 – нет.

A.7 Добавление оперативной памяти

Заполнить запись в таблице RAM.

Описание полей в таблице RAM:

- IdComponent – идентификатор компонента из таблицы Component,
- IdRAMFormFactor – идентификатор форм-фактора оперативной памяти из таблицы RAMFormFactor,
- IdRAMType – идентификатор типа оперативной памяти из таблицы RAMType,
- MemorySize – объем модуля оперативной памяти (ГБ),
- Frequency – тактовая частота (МГц),
- HasRegistr – регистровая память, 1 – присутствует, 0 – отсутствует,
- HasECC – поддержка технологии ECC, 1 – присутствует, 0 – отсутствует,
- CASLatency – время рабочего цикла,
- RASToCAASDelay – время полного доступа к данным, необязательно,
- RowPrechargeDelay – задержка в тактах для перехода от одной строки в таблице к другой, необязательно,
- ActivateToPreChargeDelay – задержка между командой активации доступа и командой закрытия строки, необязательно,

- HasRadiator – наличие радиатора, 1 – присутствует, 0 – отсутствует,
- Voltage – напряжение питания (В).

A.8 Добавление хранилища данных

Чтобы добавить жесткий диск или твердотельный накопитель сначала требуется заполнить запись в таблице DataStorage.

Описание полей в таблице DataStorage:

- IdComponent – идентификатор компонента из таблицы Component,
- MemorySize – объем (ГБ),
- Width – ширина (мм),
- Length – длина (мм),
- Thickness – толщина (мм).

A.8.1 Добавление жесткого диска

Заполнить запись в таблице HDD.

Описание полей в таблице HDD:

- IdDataStorage – идентификатор компонента из таблицы DataStorage,
- FormFactor – форм-фактор диска,
- CacheSize – объем кэш-памяти (МБ),
- RotationSpeed – скорость вращения шпинделя (об/мин),
- WriteTech – технология записи,
- ActiveNoiseLevel – уровень шума во время работы (дБ),
- PassiveNoiseLevel – уровень шума в простое (дБ),
- PassiveEnergyUse – энергопотребление в режиме ожидания (Вт),
- MaxEnergyUse – максимально энергопотребление (Вт),
- MaxTemp – максимальная рабочая температура (°C).

А.8.2 Добавление твердотельного накопителя

Заполнить запись в таблице SSD.

Описание полей в таблице SSD:

- IdDataStorage – идентификатор компонента из таблицы DataStorage,
- BitQuantityOnCell – количество бит на ячейку,
- MemoryStructure – структура памяти,
- WriteSpeed – максимальная скорость последовательной записи (Мбайт/с),
- ReadSpeed – максимальная скорость последовательного чтения (Мбайт/с),
- TotalBytesWritten – максимальный ресурс записи (ТБ),
- DWPD – доля от общего объема в процентах, на запись которой в день рассчитан накопитель в течение срока службы.

А.8.3 Добавление М.2 накопителя

Для добавления М.2 накопителя нужно заполнить таблицы DataStorage и SSD, а после заполнить M2SSD. Указать IdSSD (идентификатор компонента из таблицы SSD) и IdFormFactor (идентификатор форм-фактора М.2 накопителя из таблицы M2FormFactor).

Также следует указать ключи М.2 накопителя в таблице M2SSDKey. Указать IdM2SSD (идентификатор компонента из таблицы M2SSD) и IdKey (идентификатор ключа из таблицы M2Key).

А.9 Добавление блока питания

Заполнить запись в таблице PowerSupply.

Описание полей в таблице PowerSupply:

- IdDataStorage – идентификатор компонента из таблицы DataStorage,

- IdPowerSupplyFormFactor – идентификатор форм-фактора блока питания из таблицы PowerSupplyFormFactor,
- Power – мощность (Вт),
- SATAConnectorQuantity – количество разъемов SATA,
- IdColor – идентификатор цвета из таблицы Color,
- CoolerSystem – тип системы охлаждения,
- Length – длина (мм),
- Width – ширина (мм),
- Height – высота (мм).

Заполнить разъемы питания материнской платы блока питания в таблице PowerSupplyMotherBoardConnector. Указать IdPowerSupply (идентификатор компонента блока питания), IdMotherBoardPowerConnector (идентификатор разъема питания материнской платы из таблицы MotherBoardPowerConnector) и Quantity (количество разъемов).

Заполнить разъемы питания процессора блока питания в таблице PowerSupplyProcessorPowerConnector. Указать IdPowerSupply (идентификатор компонента блока питания), IdProcessorPowerConnector (идентификатор разъема питания процессора из таблицы ProcessorPowerConnector) и Quantity (количество разъемов).

Заполнить разъемы видеокарты блока питания в таблице PowerSupplyVideoPowerConnector. Указать IdPowerSupply (идентификатор компонента блока питания), IdVideoPowerConnector (идентификатор разъема питания видеокарты из таблицы VideoPowerConnector) и Quantity (количество разъемов).

Приложение Б

(справочное)

Таблицы коэффициентов

В таблице Б.1 представлены значения коэффициента с, а в таблице Б.2, коэффициент квалификации разработчика.

Таблица Б.1 – Значение коэффициента с

Язык программирования	Группа сложности	Степень новизны				Коэффициент В
		А	Б	В	Г	
Высокого уровня	1	1,38	1,26	1,15	1,20	1,20
	2	1,30	1,19	1,08	0,65	1,35
	3	1,20	1,10	1,00	0,60	1,50
Низкого уровня	1	1,58	1,45	1,32	0,79	1,20
	2	1,49	1,37	1,24	0,74	1,35
	3	1,38	1,26	1,15	0,69	1,50

Таблица Б.2 – Коэффициент квалификации разработчика

Опыт работы	Коэффициент квалификации
До двух лет	0,80
2-3 года	1,00
3-5 лет	1,10 – 1,20
5-7 лет	1,30 – 1,40
Более 7 лет	1,50 – 1,60