

Минобрнауки России

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный технологический институт

(технический университет)»

|  |  |  |
| --- | --- | --- |
| Направление | 27.03.03 | Системный анализ и управление |
| Факультет |  | Информационных технологий и управления |
| Кафедра |  | Системного анализа и информационных технологий |
| ***Учебная дисциплина*** |  | ***Базы Данных*** |
| Курс 2 |  | Группа 436 |

**КУРСОВАЯ РАБОТА**

**Тема:**

**Разработка базы данных, реализующая сборку персонального компьютера.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |   (дата, подпись) | |  | | --- | | Филимонов И.И. |   (фамилия, инициалы) |
| Руководитель, доцент | |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |   (дата, подпись) | |  | | --- | | Гайков А.В. |   (фамилия, инициалы) |
| Оценка за курсовую работу | |  | | --- | | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |   (оценка) | |  | | --- | |  |   (подпись руководителя) |

Санкт-Петербург

2025



Минобрнауки России

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Санкт-Петербургский государственный технологический институт (технический университет)»

**ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Направление | 27.03.03 | | Системный анализ и управление | |
| Направленность | |  | | Информационные системы и технологии | |
| Факультет | |  | | Информационных технологий и управления | |
| Кафедра | |  | | Системного анализа и информационных технологий | |
| ***Учебная дисциплина*** | |  | | ***Базы Данных*** | |
| Курс 2 | |  | | Группа 436 | |
| Студент | |  | | Филимонов Иван Игоревич | |
|  | |  | |  | |

**Тема:** Разработка базы данных, реализующая сборку персонального компьютера.

***Исходные данные к работе:***

Литература по предметной области:

1. Фуфаев, Э. В. Разработка и эксплуатация удаленных баз данных: учеб. для сред. проф. образования / Э. В. Фуфаев, Д. Э. Фуфаев. - М.: Академия, 2018. - 251 с..
2. Зиборов В. В. Visual Basic 2010 на примерах / Виктор Зиборов. — Санкт-Петербург : БХВ-Петербург, 2010. — 329 с. ил., табл.; 24. — ISBN 978-5-9775-0402-7.
3. Структуры данных и алгоритмы: А. В. Ахо, Д. Э. Хопкрофт, Д. Д. Ульман – Москва, 2001 – 384 с.

Интернет-ресурсы по предметной области:

1. https://ru.wikipedia.org/wiki/Персональный\_компьютер
2. https://hi-tech.mail.ru/review/57070-kak-sobrat-pk-chtoby-hvatilo-nadolgo/#anchor171508422401387013

**Перечень подлежащих разработке вопросов:**

1. Анализ и описание предметной области.
2. Постановка задачи. Поиск и сбор исходных данных для заполнения базы.
3. Выделение сущностей и отношений между ними, описание их свойств.
4. Разработка инфологической и дата-логической модели.
5. Создание объектов базы данных в среде разработки.
6. Разработка интерфейса и сценариев взаимодействия.
7. Кодирование алгоритмов, построение sql-запросов, тестирование, отладка.
8. Оформление пояснительной записки и графического материала.

**Перечень графического материала:**

1. Формализованное описание предметной области.
2. Модели представления данных, схемы данных, диаграммы сущностей.
3. Описание sql-запросов к базе данных.
4. Графические схемы алгоритмов и диаграммы сценариев взаимодействия.
5. Тестовый план и пример применения.

**Виды и объем работы, выполняемой с использованием ЭВМ:**

Характеристика аппаратного обеспечения: PC на базе Intel Core i7/i5/i3, RAM не менее 2Гб,

HDD не менее 250 Гб, видеокартой (любая совместимая), наличие USB.

Характеристика программного обеспечения: MS Windows 11, MS Windows 7, MS Word 2010, MS Visio 2010

Характеристика среды программирования: SQLite (DB Browser), Visual Studio 2022.

***Требования к оформлению***

Курсовая работа должна быть оформлена в соответствии с требованиями СТО СПбГТИ 044 – 2012 КС УКДВ. Виды учебных занятий. Курсовой проект. Курсовая работа. Общие требования.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Дата выдачи задания** | |  | **14.03.2025 г.** | | |
| **Срок представления к защите** | |  | **30.05.2025 г.** | | |
| **Зав. кафедрой** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |  | д.т.н., профессор | |
| Мусаев А.А. | |
|  | (подпись) | | |  |  | |
| **Руководитель,**  **Должность** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |  | к.т.н., доцент  Гайков А.В. | |
|  | (подпись) | | |  |  | |
| **Задание принял**  **к выполнению** | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | | |  | Филимонов И.И. | |
|  | (подпись) | | |  |  | |

Оглавление

[Введение 4](#_Toc199443518)

[1 ЦЕЛИ КУРСОВОЙ РАБОТЫ 5](#_Toc199443519)

[2 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc199443520)

[2.1 Описание предметной области 6](#_Toc199443521)

[2.2 Перечень входных данных 7](#_Toc199443522)

[2.3 Перечень выходных данных 11](#_Toc199443523)

[2.4 Постановка задачи 11](#_Toc199443524)

[2.5 План работы 12](#_Toc199443525)

[3 Архитектура базы данных 13](#_Toc199443526)

[3.1 Описание объектов данных БД 13](#_Toc199443527)

[3.2 Описание запросов создания таблиц БД 19](#_Toc199443528)

[3.3 Описание приложения для взаимодействия с БД 22](#_Toc199443529)

[4 Выводы о работе 30](#_Toc199443530)

[Список литературы 31](#_Toc199443531)

[Приложение А: Листинг программы с комментариями 32](#_Toc199443532)

# Введение

База данных представляет собой организованную совокупность взаимосвязанных данных и программных средств, предназначенных для их хранения, обработки и управления. В современном мире базы данных широко применяются для автоматизации различных процессов, что значительно повышает эффективность и удобство работы с информацией.

Одной из актуальных областей для автоматизации является сборка персональных компьютеров. Управление комплектующими требует надежной и структурированной системы хранения данных. Для решения этих задач создаются специализированные базы данных, которые упрощают обработку информации и обеспечивают быстрый доступ к необходимым данным.

Цель данной курсовой работы — разработка базы данных для автоматизации процесса сборки ПК, предназначенной для хранения и обработки информации о комплектующих. В качестве системы управления базами данных используется Microsoft Access, а для создания пользовательского интерфейса применяется язык программирования Visual Basic (VB), что позволяет создать удобное и функциональное приложение для работы с базой.

В ходе работы будет разработана структура базы данных с основными таблицами и их связями, а также реализован графический интерфейс, обеспечивающий выполнение основных операций с данными. Реализация данной системы позволит закрепить практические навыки проектирования реляционных баз данных и разработки приложений на VB с использованием Access в качестве СУБД.

# 1 ЦЕЛИ КУРСОВОЙ РАБОТЫ

Целью курсовой работы является разработка базы данных для сборки персонального компьютера, обеспечивающей хранение и обработку информации о всех комплектующих компьютера. Для реализации поставленной цели используется система управления базами данных Access и язык программирования VB с применением Windows Forms для создания интерфейса взаимодействия с пользователем.

В функциональность разрабатываемой системы закладывается:

1. Хранение информации о комплектующих персонального компьютера.
2. Фильтрация комплектующих в соответствии с выбранными.
3. Расчет итоговой стоимости.
4. Сохранение сборки персонального компьютера в файл.

**Задачи курсовой работы:**

* Анализ и описание предметной области.
* Определение сущностей и связей между ними.
* Разработка инфологической и даталогической модели базы данных.
* Реализация базы данных с использованием СУБД Access.
* Создание графического интерфейса для взаимодействия с базой данных на Visual Basic (WinForms).
* Проведение тестирования системы и демонстрация её функциональности.
* Подготовка пояснительной записки и сопроводительных материалов.

# 2 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 2.1 Описание предметной области

Предметной областью базы данных является персональный компьютер. В компьютере имеются комплектующие, которые подразделяются на 8 категорий:

1. Процессор (CPU)
2. Видеокарта (Графический чип, GPU)
3. Материнская плата (Motherboard)
4. Оперативная память (RAM)
5. Блок питания (Powerplant)
6. Охлаждение (Cooling)
7. Диск (Disk)
8. Корпус (Frame)

Для каждой категории требуется таблица, хранящая данные о конкретных единицах комплектующих, которые будут предоставляться пользователю на выбор для сборки персонального компьютера.

Достаточное охлаждение

CPU

Cooling

Компьютер

Powerplant

Должен обеспечивать достаточной энергией

Поддержка максимальной скорости

Совпадение сокета

Поддержка интерфейсов

RAM

Motherboard

GPU

GPU должен помещаться по длине

Совпадение типа

Disk

Совпадение форм-фактора

Нужен для хранения операционной системы

Frame

Рисунок 1 — Концептуальная модель предметной области

## 2.2 Перечень входных данных

Входными данными являются следующие таблицы под каждую категорию:

Таблица “Cooling”, которая содержит следующие данные:

* Производитель
* Вес
* Тип
* Сокет
* Назначение
* Рассеиваемая мощность, TDP
* Длина
* Ширина
* Стоимость

Таблица “CPU”, которая содержит следующие данные:

* Производитель
* Вес
* Модель
* Сокет
* Система охлаждения в комплекте
* Общее количество ядер
* Количество производительных ядер
* Максимальное число потоков
* Объем кэша
* Базовая частота процессора
* Максимальная частота в турбо
* Тип памяти
* Частота оперативной памяти
* Максимально поддерживаемый объем памяти
* Тепловыделение (TDP)
* Интегрированное графическое ядро
* Встроенный контроллер PCI Express
* Число линий PCI Express
* Стоимость

Таблица “Disk”, которая содержит следующие данные:

* Производитель
* Вес
* Тип
* Объем накопителя
* Физический интерфейс
* Максимальная скорость последовательного чтения
* Максимальная скорость последовательной записи
* Максимальный ресурс записи
* Длина
* Стоимость

Таблица “Frame”, которая содержит следующие данные:

* Производитель
* Высота
* Ширина
* Длинна
* Вес
* Цвет
* Материал корпуса
* Форм фактор
* Максимальная длинна
* Видеокарты
* Возможность установки водяного охлаждения
* Стоимость

Таблица “GPU”, которая содержит следующие данные:

* Производитель
* Вес
* Цвет
* Модель
* Графический процессор
* Штатная частота работы видеочипа
* Турбочастота
* Поддержка трассировки лучей
* Объем видеопамяти
* Тип памяти
* Тип видеоразъемов
* Количество видеоразъемов
* Интерфейс подключения
* Разъемы дополнительного питания
* Тип охлаждения
* Подсветка элементов видеокарты
* Стоимость

Таблица “Motherboard”, которая содержит следующие данные:

* Производитель
* Вес
* Цвет
* Ширина
* Высота
* Сокет
* Тип поддерживаемой памяти
* Максимальный объем памяти
* Максимальная частота памяти
* Количество слотов памяти
* Версия PCI Express
* Количество разъемов M.2
* Порты USB
* Видеовыходы
* Форм-фактор
* Форм фактор питания
* Сетевой адаптер
* Скорость сетевого адаптера
* Стоимость

Таблица “Powerplant”, которая содержит следующие данные:

* Производитель
* Вес
* Цвет
* Мощность (номинал)
* Форм-фактор
* Основной разъем питания
* Разъемы для питания процессора
* Разъемы для питания видеокарты
* Количество разъемов 15-pin SATA
* Количество разъемов 4-pin Molex
* Система охлаждения
* Сертификация
* Стоимость

Таблица “RAM”, которая содержит следующие данные:

* Производитель
* Цвет
* Тип памяти
* Суммарный объем памяти всего комплекта
* Объем одного модуля памяти
* Количество модулей в комплекте
* Тактовая частота
* Наличие радиатора
* Стоимость

## 2.3 Перечень выходных данных

Выходными данными базы данных будет являться схематически собранный персональный компьютер, а также посчитанная стоимость сборки такого компьютера в рублях. В том числе и основные характеристики собранной конфигурации.

## 2.4 Постановка задачи

Для автоматизации предметной области необходимо требуется разработать базу данных, которая предоставляет функциональность в соответствии с целями проектирования, а также содержащая таблицы для хранения необходимых данных. Для достижения цели курсовой работы необходимо выполнить следующие этапы:

* Исследовать объекты предметной области, выделить их основные свойства, которые могут быть полезны для реализации поставленных задач;
* Проанализировать характер связи между объектами, и выделить отношения, подлежащие автоматизации, которые могут быть полезны при разработке базы данных;
* Проанализировать операции и действия, выполняемые над объектами предметной области, подлежащие автоматизации
* На основе анализа разработать модель данных предметной области;
* Определить наборы и потоки данных, требуемые для выполнения операций над объектами предметной области, и разработать на их основе систему запросов;
* Определить содержание и формат входных данных для информационной системы;
* Определить содержание и графическое представление выходных графических элементов;
* Разработать графический пользовательский интерфейс для управления объектами предметной области на языке Visual Basic с использованием технологии WinForms;
* Разработать систему управляющих команд и алгоритмов обработки для выполнения аналитических функций над объектами предметной области;
* Произвести наполнение базы данных тестовыми данными;
* Произвести тестирование функциональности базы данных на корректность данных и правильность выполнения операций;
* Перевести базу данных в штатный режим эксплуатации.

Создание такой базы данных позволит упростить процесс конструирования новых персональных компьютеров под различные задачи.

## 2.5 План работы

1. Создать таблицы, описанные в разделе 2.2
2. Сформировать запросы для построения базы данных.
3. Сформировать запросы для реализации функционала взаимодействия между пользователем и базой данных.
4. Сформировать формы для обеспечения взаимодействия пользователя с базой данных и построения конфигурации.
5. Создать меню в виде кнопочной формы, для открытия форм взаимодействия с таблицами базы данных.

Создание интерфейса взаимодействия пользователя с базой данных будет проводится на языке Visual Basic с использованием технологии WinForms.

# 3 Архитектура базы данных

## 3.1 Описание объектов данных БД

Для хранения данных, как было сказано ранее, используются следующие таблицы:

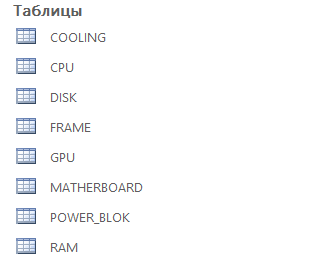


Рисунок 2 – Перечень таблиц

Таблица 1 – “Cooling” хранит информацию о охлаждении компьютера.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Вес |  |  |  |  |  |
| Тип | TEXT | 255 сим. | Любые символы | –– | –– |
| Сокет | TEXT | 255 сим. | Любые символы | –– | –– |
| Назначение | TEXT | 255 сим. | Любые символы | –– | –– |
| Рассеиваемая мощность, TDP | INTEGER | 32 байта | Числа | –– | –– |
| Длина | INTEGER | 32 байта | Числа | –– | –– |
| Ширина | INTEGER | 32 байта | Числа | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Вес |  |  |  |  |  |
| Модель | TEXT | 255 сим. | Любые символы | –– | –– |
| Сокет | TEXT | 255 сим. | Любые символы | –– | –– |
| Система охлаждения в комплекте | TEXT | 255 сим. | Любые символы | –– | –– |
| Общее количество ядер | INTEGER | 32 байта | Числа | –– | –– |
| Количество производительных ядер | INTEGER | 32 байта | Числа | –– | –– |
| Максимальное число потоков | INTEGER | 32 байта | Числа | –– | –– |
| Объем кэша | INTEGER | 32 байта | Числа | –– | –– |
| Базовая частота процессора | INTEGER | 32 байта | Числа | –– | –– |
| Максимальная частота в турбо | INTEGER | 32 байта | Числа | –– | –– |
| Тип памяти | TEXT | 255 сим. | Любые символы | –– | –– |
| Частота оперативной памяти | INTEGER | 32 байта | Числа | –– | –– |
| Максимально поддерживаемый объем памяти | INTEGER | 32 байта | Числа | –– | –– |
| Тепловыделение (TDP) | INTEGER | 32 байта | Числа | –– | –– |
| Интегрированное графическое ядро | TEXT | 255 сим. | Любые символы | –– | –– |
| Встроенный контроллер PCI Express | TEXT | 255 сим. | Любые символы | –– | –– |
| Число линий PCI Express | INTEGER | 32 байта | Числа | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

Таблица 2 – “CPU” хранит информацию о процессоре.

Таблица 3 – “Disk” хранит информацию о диске.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Вес | INTEGER | 32 байта | Числа | –– | –– |
| Тип | INTEGER | 32 байта | Числа | –– | –– |
| Объем накопителя | INTEGER | 32 байта | Числа | –– | –– |
| Физический интерфейс | INTEGER | 32 байта | Числа | –– | –– |
| Максимальная скорость последовательного чтения | INTEGER | 32 байта | Числа | –– | –– |
| Максимальная скорость последовательной записи | INTEGER | 32 байта | Числа | –– | –– |
| Максимальный ресурс записи | INTEGER | 32 байта | Числа | –– | –– |
| Длина | INTEGER | 32 байта | Числа | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

Таблица 4 – “Frame” хранит информацию о корпусе.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Высота | INTEGER | 32 байта | Числа | –– | –– |
| Ширина | INTEGER | 32 байта | Числа | –– | –– |
| Длинна | INTEGER | 32 байта | Числа | –– | –– |
| Вес | INTEGER | 32 байта | Числа | –– | –– |
| Цвет | TEXT | 255 сим. | Любые символы | –– | –– |
| Материал корпуса | TEXT | 255 сим. | Любые символы | –– | –– |
| Форм фактор | TEXT | 255 сим. | Любые символы | –– | –– |
| Максимальная длинна Видеокарты | INTEGER | 32 байта | Числа | –– | –– |
| Возможность установки водяного охлаждения | TEXT | 255 сим. | Любые символы | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

Таблица 5 – “GPU” хранит информацию о графическом чипе.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Вес | INTEGER | 32 байта | Числа | –– | –– |
| Цвет | TEXT | 255 сим. | Любые символы | –– | –– |
| Модель | TEXT | 255 сим. | Любые символы | –– | –– |
| Графический процессор | TEXT | 255 сим. | Любые символы | –– | –– |
| Штатная частота работы видеочипа | INTEGER | 32 байта | Числа | –– | –– |
| Турбочастота | INTEGER | 32 байта | Числа | –– | –– |
| Поддержка трассировки лучей | TEXT | 255 сим. | Любые символы | –– | –– |
| Объем видеопамяти | INTEGER | 32 байта | Числа | –– | –– |
| Тип памяти | INTEGER | 32 байта | Числа | –– | –– |
| Тип видеоразъемов | TEXT | 255 сим. | Любые символы | –– | –– |
| Количество видеоразъемов | INTEGER | 32 байта | Числа | –– | –– |
| Интерфейс подключения | TEXT | 255 сим. | Любые символы | –– | –– |
| Разъемы дополнительного питания | INTEGER | 32 байта | Числа | –– | –– |
| Тип охлаждения | TEXT | 255 сим. | Любые символы | –– | –– |
| Подсветка элементов видеокарты | TEXT | 255 сим. | Любые символы | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

Таблица 6 – “Motherboard” хранит информацию о материнской плате.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Вес | INTEGER | 32 байта | Числа | –– | –– |
| Цвет | TEXT | 255 сим. | Любые символы | –– | –– |
| Ширина | INTEGER | 32 байта | Числа | –– | –– |
| Высота | INTEGER | 32 байта | Числа | –– | –– |
| Сокет | TEXT | 255 сим. | Любые символы | –– | –– |
| Тип поддерживаемой памяти | TEXT | 255 сим. | Любые символы | –– | –– |
| Максимальный объем памяти | INTEGER | 32 байта | Числа | –– | –– |
| Максимальная частота памяти | INTEGER | 32 байта | Числа | –– | –– |
| Количество слотов памяти | INTEGER | 32 байта | Числа | –– | –– |
| Версия PCI Express | TEXT | 255 сим. | Любые символы | –– | –– |
| Количество разъемов M.2 | INTEGER | 32 байта | Числа | –– | –– |
| Количество портов USB | INTEGER | 32 байта | Числа | –– | –– |
| Видеовыходы | TEXT | 255 сим. | Любые символы | –– | –– |
| Форм-фактор | TEXT | 255 сим. | Любые символы | –– | –– |
| Форм фактор питания | TEXT | 255 сим. | Любые символы | –– | –– |
| Сетевой адаптер | TEXT | 255 сим. | Любые символы | –– | –– |
| Скорость сетевого адаптера | INTEGER | 32 байта | Числа | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

Таблица 7 – “Power\_block” хранит информацию о блоке питания.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Вес | INTEGER | 32 байта | Числа | –– | –– |
| Цвет | TEXT | 255 сим. | Любые символы | –– | –– |
| Мощность (номинал) | INTEGER | 32 байта | Числа | –– | –– |
| Форм-фактор | TEXT | 255 сим. | Любые символы | –– | –– |
| Основной разъем питания | TEXT | 255 сим. | Любые символы | –– | –– |
| Разъемы для питания процессора | TEXT | 255 сим. | Любые символы | –– | –– |
| Разъемы для питания видеокарты | TEXT | 255 сим. | Любые символы | –– | –– |
| Количество разъемов 15-pin SATA | INTEGER | 32 байта | Числа | –– | –– |
| Количество разъемов 4-pin Molex | INTEGER | 32 байта | Числа | –– | –– |
| Система охлаждения | TEXT | 255 сим. | Любые символы | –– | –– |
| Сертификация | TEXT | 255 сим. | Любые символы | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

Таблица 8 – “RAM” хранит информацию о оперативной памяти.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название | Тип данных | Размер | Формат | Значение по умолчанию | Примечание |
| Id | INTEGER | 32 байта | Числа | Следующее число | Ключевое, обязательное, индекс без совпадений |
| Производитель | TEXT | 255 сим. | Любые символы | –– | –– |
| Цвет | TEXT | 255 сим. | Любые символы | –– | –– |
| Тип памяти | TEXT | 255 сим. | Любые символы | –– | –– |
| Суммарный объем памяти всего комплекта | INTEGER | 32 байта | Числа | –– | –– |
| Объем одного модуля памяти | INTEGER | 32 байта | Числа | –– | –– |
| Количество модулей в комплекте | INTEGER | 32 байта | Числа | –– | –– |
| Тактовая частота | INTEGER | 32 байта | Числа | –– | –– |
| Наличие радиатора | TEXT | 255 сим. | Любые символы | –– | –– |
| Стоимость | INTEGER | 32 байта | Числа | –– | –– |

## 3.2 Описание запросов создания таблиц БД

Для создания таблиц и инициализации базы данных используются следующие запросы. Так как для взаимодействия с базой данных используется программа, написанная на языке VB, все запросы были реализованы как инструкции на языке запросов SQL.

Таблица 9 – Запросы создания таблиц БД.

|  |  |
| --- | --- |
| Создание таблицы | Инструкция SQL (Create-Statement таблицы) |
| Cooling | CREATE TABLE COOLING (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Weight INTEGER,  Type TEXT,  Soket TEXT,  For\_what TEXT,  TDP INTEGER,  Length INTEGER,  Hight INTEGER,  Cost INTEGER  ); |
| CPU | CREATE TABLE CPU (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Weight INTEGER,  Model TEXT,  Soket TEXT,  Sistem\_cooling\_in TEXT,  Cores INTEGER,  Process\_cores INTEGER,  Max\_kol\_potok INTEGER,  KESH INTEGER,  Baze\_speed INTEGER,  Max\_speed INTEGER,  Type\_RAM TEXT,  Max\_speed\_RAM INTEGER,  Max\_RAM INTEGER,  TDP INTEGER,  In\_GPU TEXT,  Control\_PCI TEXT,  Kol\_lines INTEGER,  Cost INTEGER  ); |
| Disk | CREATE TABLE DISK (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Weight INTEGER,  Type TEXT,  V\_of\_disk INTEGER,  Interface TEXT,  Max\_speed\_read INTEGER,  Max\_speed\_write INTEGER,  Max\_V\_write INTEGER,  Length INTEGER,  Cost INTEGER  ); |
| Frame | CREATE TABLE FRAME (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Hight INTEGER,  Width INTEGER,  Length INTEGER,  Weight INTEGER,  Color TEXT,  Material TEXT,  Form\_factor TEXT,  Max\_lenght\_of\_GPU INTEGER,  Water\_cooling TEXT,  Cost INTEGER  ); |
| GPU | CREATE TABLE GPU (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Weight INTEGER,  Color TEXT,  Model TEXT,  Graphic\_unit TEXT,  Frequency INTEGER,  Turbofrequency INTEGER,  DLSS TEXT,  V\_GPU INTEGER,  Type\_of\_memory TEXT,  Type\_of\_ports TEXT,  Kol\_ports INTEGER,  Interface TEXT,  Dop\_power INTEGER,  Type\_of\_cooling TEXT,  LED TEXT,  Cost INTEGER  ); |
| Motherboard | CREATE TABLE MOTHERBOARD (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Weight INTEGER,  Color TEXT,  Length INTEGER,  Width INTEGER,  Soket TEXT,  Type\_RAM TEXT,  Max\_V\_RAM INTEGER,  Max\_speed\_RAM INTEGER,  Kol\_slots\_RAM INTEGER,  PCI\_version TEXT,  Kol\_M2 INTEGER,  USB INTEGER,  Video\_port TEXT,  Form\_factor TEXT,  Esernet\_adapter TEXT,  Esernet\_speed INTEGER,  Cost INTEGER  ); |
| Power\_block | CREATE TABLE POWER\_BLOCK (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Weight INTEGER,  Color TEXT,  Сapacity INTEGER,  Form\_factor TEXT,  Power\_CPU INTEGER,  Power\_GPU INTEGER,  "15\_pin\_SATA" INTEGER,  "4\_pin\_Molex" INTEGER,  Cooling\_system TEXT,  Sertif TEXT,  Cost INTEGER  ); |
| RAM | CREATE TABLE RAM (  Id INTEGER PRIMARY KEY,  Manufacturer TEXT,  Color TEXT,  Type\_RAM TEXT,  Sum\_RAM INTEGER,  One\_RAM INTEGER,  Kol\_RAM INTEGER,  Speed\_RAM INTEGER,  Cooler\_RAM TEXT,  Cost INTEGER  ); |

## 3.3 Описание приложения для взаимодействия с БД

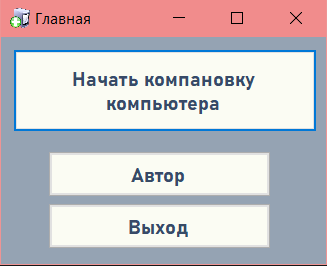
Главное меню программы представляет собой форму с кнопками, первая из них переносит пользователя в основное меню для связи с другими формами, которые реализуют различные функции взаимодействия с базой данных.

Рисунок 3 – Главное меню программы.

После пользователь попадает в основное меню, где в правой части программы показано дерево с итоговой конфигурацией. В левом углу видна интерактивная схема комплектующих, нажимая на которые пользователь будет переходить к выбору конкретного элемента. Снизу находятся кнопки, отвечающие за сохранение и подсчет стоимости.

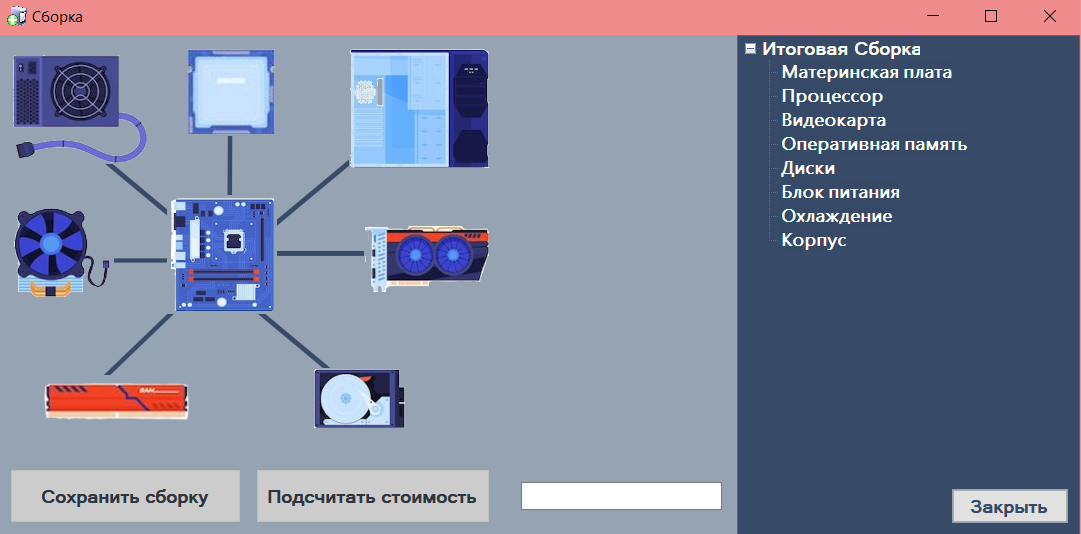


Рисунок 4 – Основное меню программы для выбора комплектующих.

В первую очередь пользователю будет доступен выбор процессора, так как это основа для дальнейшего построения компьютера.

При нажатии на картинку процессора отображается окно с выбором производителя и модели процессора. А также высвечивается другая информация об этом процессоре.

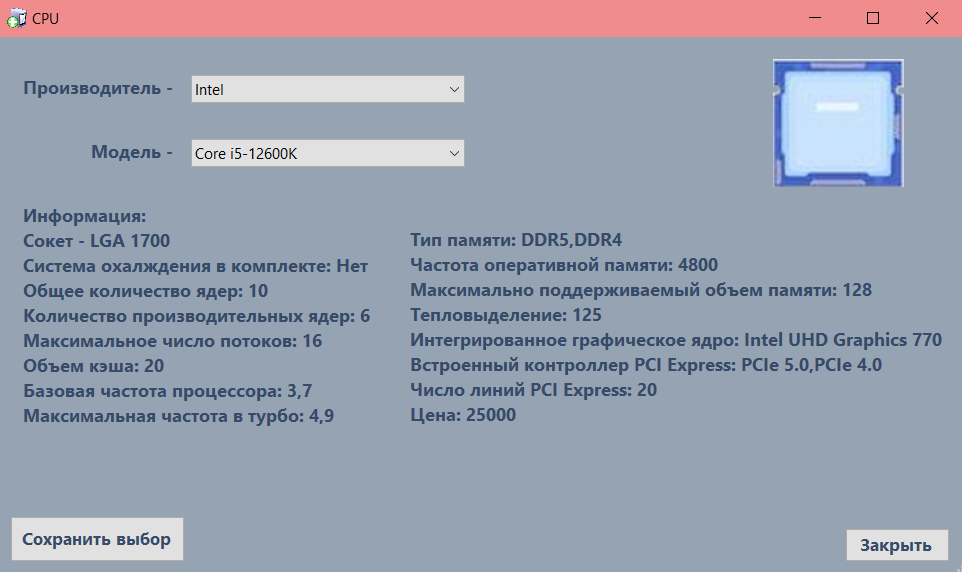


Рисунок 4 – Окно выбора процессора.

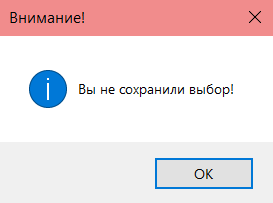
После пользователь нажимает кнопку «Сохранить выбор», иначе при закрытии пользователь предупредят о не сохранении выбора.

Рисунок 5 – Сообщение о не сохранённом выборе.

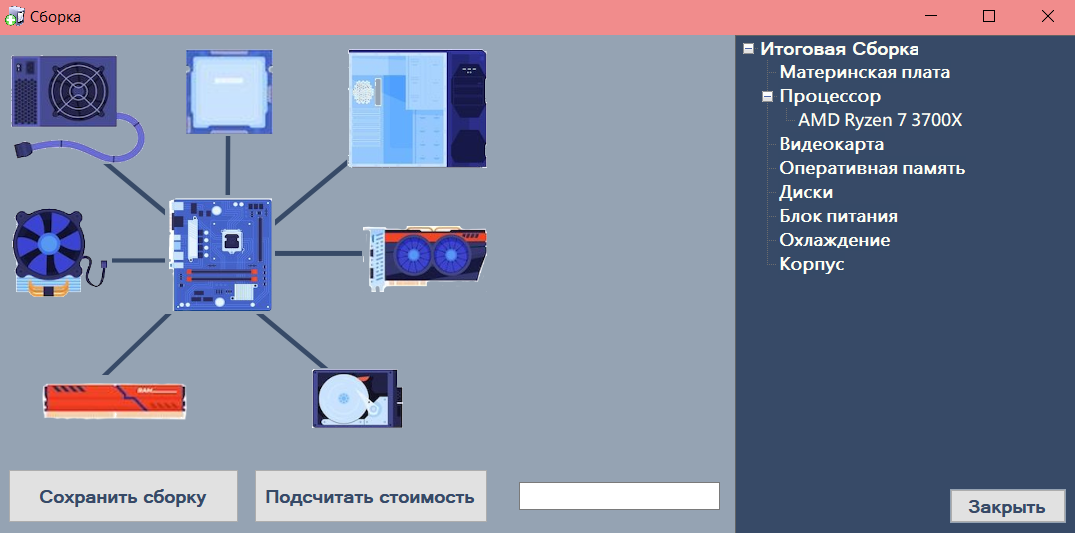
В случае успешного сохранения с право в дереве добавится еще одна ветка с короткой информаций о выбранном компоненте.

Рисунок 6 – Основное меню программы с выбранным процессором.

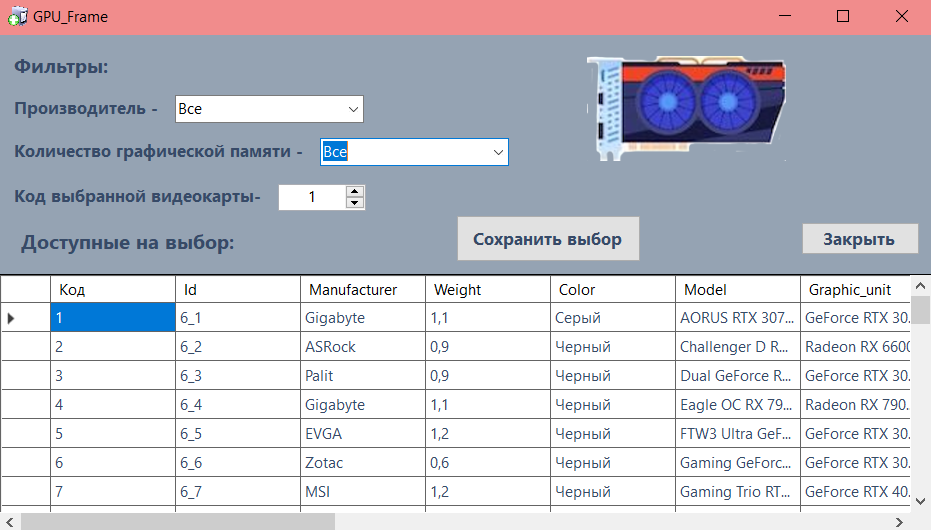
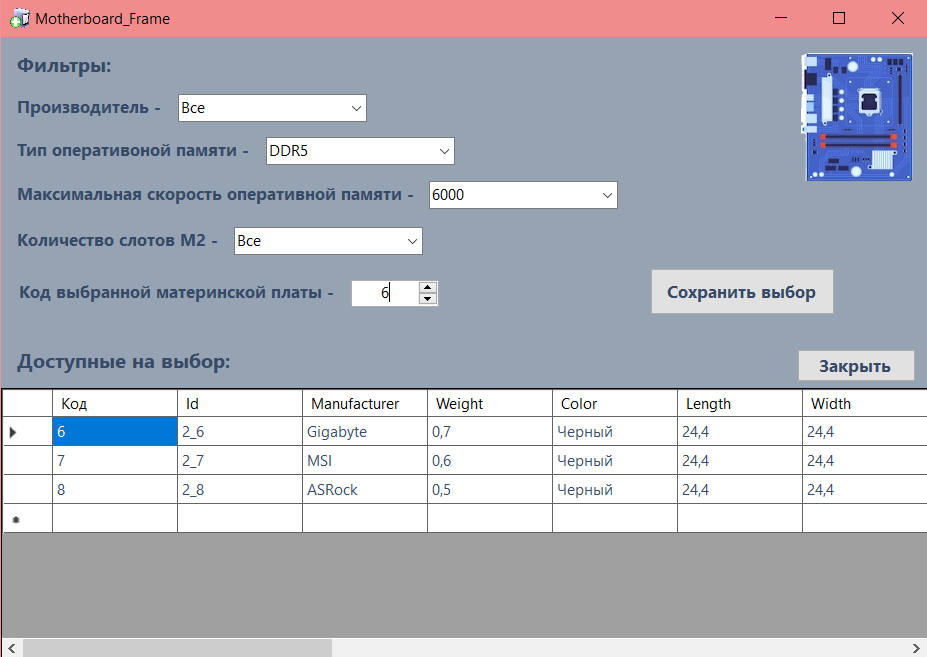
После выбора процессора становятся доступными для выбора остальные элементы. При выборе любого из них, в окне выбора комплектующего будут несколько фильтров и кнопка «Сохранить выбор». Выбор осуществляется путём введения в специальное поле уникального кода выбранного комплектующего. Программа автоматически обновляет таблицу в нижней части окна при использовании фильтров, и предоставляет выбор только из элементов, которые совместимы с остальными.

Рисунок 6 – Меню выбора видеокарты.

Рисунок 7 – Меню выбора материнской платы с активными фильтрами.

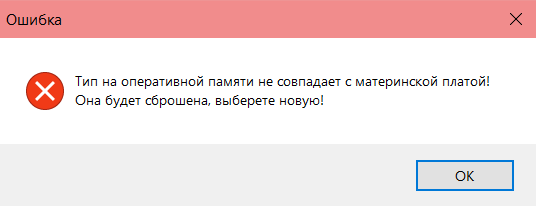
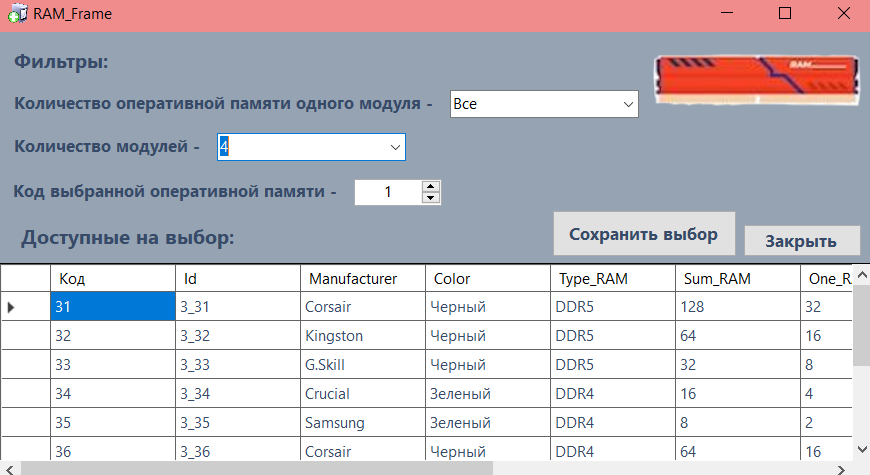
В случае принудительного выбора, не подходящего сборке комплектующего, приложение выдаст ошибку и сбросит элемент, который был выбран ранее.  
Рисунок 8 – Ошибка о несоответствии материнской платы с выбранной оперативной памятью. 

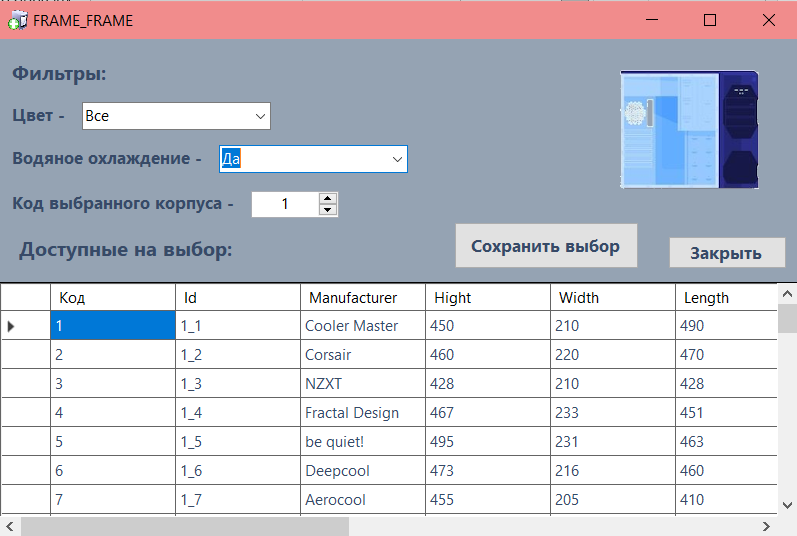
Рисунок 9 – Меню выбора оперативной памяти. 

Рисунок 10 – Меню выбора корпуса.

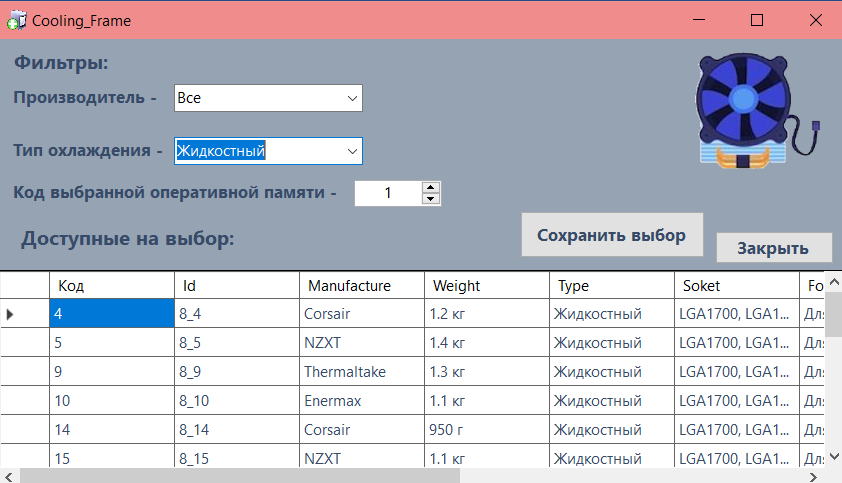
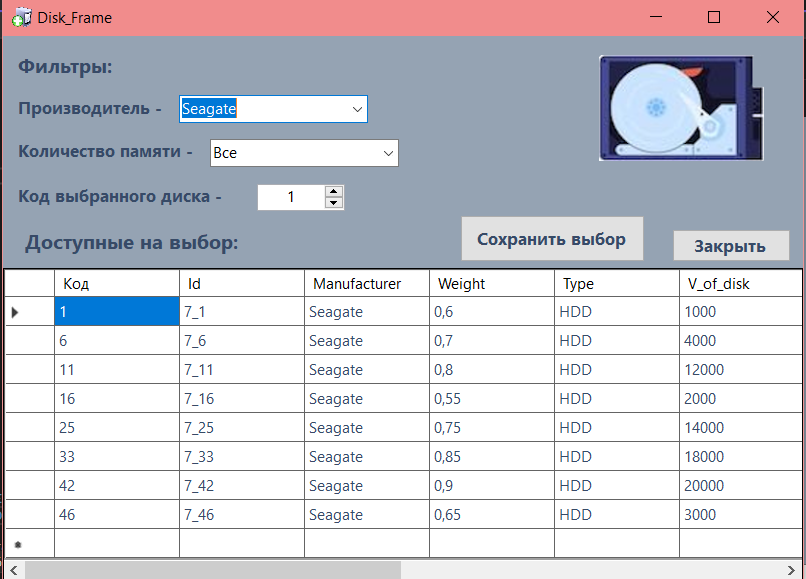
Рисунок 11 – Меню выбора диска.

Рисунок 12 – Меню выбора охлаждения.



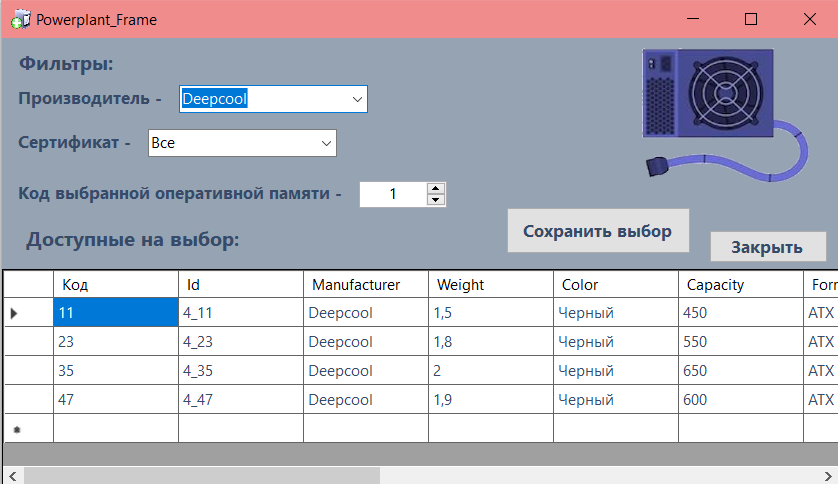
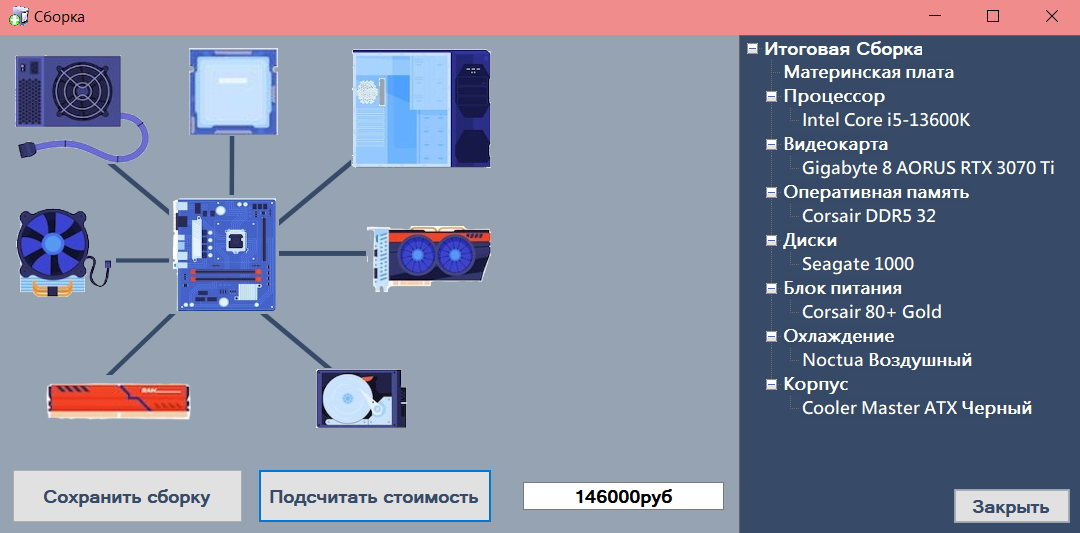
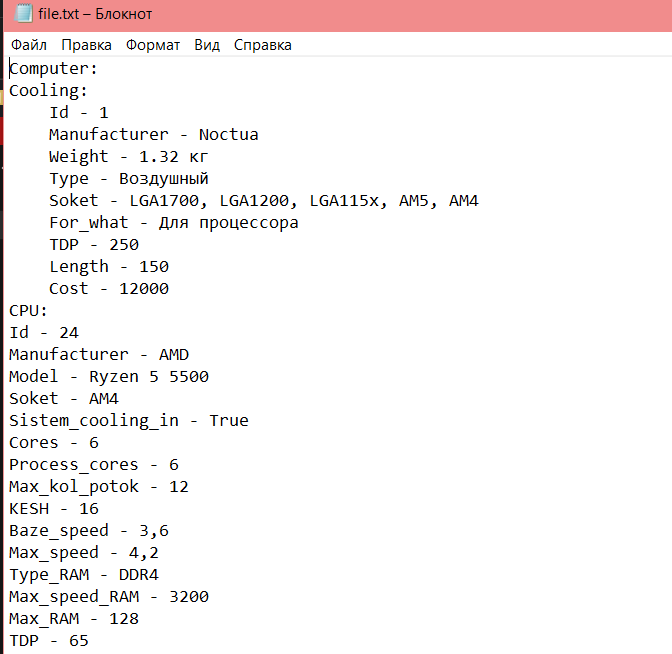
Рисунок 13 – Меню выбора блока питания.

Рисунок 14 – Основное меню с подсчитанной стоимостью.

Если в дереве с право кликнуть на не нужный элемент он пропадёт из сборки. После нажатия на кнопку «Сохранить сборку» результаты будет выведен в файл формата «.txt».

 Рисунок 15 – Пример файла вывода.

# 4 Выводы о работе

В процессе выполнения курсовой работы была изучена методология разработки баз данных и выполнены основные этапы их проектирования. Был проведён анализ предметной области: были выделены объекты, подлежащие автоматизации, был определён характер связей между ними, были установлены свойства объектов, была создана модель данных и разработано программное обеспечение для взаимодействия с базой данных на языке Visual Basic. Графический интерфейс приложения был реализован с помощью технологии WinForms. Были запрограммированы алгоритмы для обработки и поиска данных.

В ходе работы были приобретены важные практические навыки в создании и проектировании баз данных, было освоено создание запросов на языке SQL, а также была получена практика разработки программного обеспечения и использования библиотек на языках высокого уровня для связи приложения с базой данных.

# Список литературы

1. Фуфаев, Э. В. Разработка и эксплуатация удаленных баз данных: учеб. для сред. проф. образования / Э. В. Фуфаев, Д. Э. Фуфаев. - М.: Академия, 2018. - 251 с..
2. Зиборов В. В. Visual Basic 2010 на примерах / Виктор Зиборов. — Санкт-Петербург : БХВ-Петербург, 2010. — 329 с. ил., табл.; 24. — ISBN 978-5-9775-0402-7.
3. Структуры данных и алгоритмы: А. В. Ахо, Д. Э. Хопкрофт, Д. Д. Ульман – Москва, 2001 – 384 с.
4. Хомоненко, А. Д. Базы данных: учеб. для вузов / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев; под ред. А. Д. Хомоненко.- 6-е изд. - СПб. : КОРОНА-Век, 2017. - 736 с
5. Карпова, Т.С. Базы данных. Модели, разработка, реализация / Т.С. Карпова. - М.: СПб: Питер, 2001. - 304 c.
6. Постолит, А.В. Visual Studio .NET: разработка приложений баз данных / А.В. Постолит, - М.: СПб: БХВ, 2003. - 544 c.

# Приложение А: Листинг программы с комментариями

Public Class Main\_Form

Private Sub Butt\_start\_Click(sender As Object, e As EventArgs) Handles Butt\_start.Click

Sborka.Show()

Me.Hide()

End Sub

Private Sub butt\_avtor\_Click(sender As Object, e As EventArgs) Handles butt\_avtor.Click

MessageBox.Show("Автор: Филимонов Иван Гр.436

Дата создания 18.05.25", "Информация", MessageBoxButtons.OK, MessageBoxIcon.Information)

End Sub

Private Sub butt\_exit\_Click(sender As Object, e As EventArgs) Handles butt\_exit.Click

Me.Close()

End Sub

End Class

Imports System.Diagnostics.Metrics

Imports System.IO

Imports System.Windows.Forms.VisualStyles.VisualStyleElement.Window

Public Class Sborka

Private \_computer As Computer

Public Property computer As Computer ' Public свойство для доступа к объекту

Get

Return \_computer

End Get

Set(value As Computer)

\_computer = value

End Set

End Property

Private Sub butt\_motherboard\_Click(sender As Object, e As EventArgs) Handles butt\_motherboard.Click

Dim result As DialogResult = Motherboard\_Frame.ShowDialog()

If result = DialogResult.OK Then

Dim Motherboard\_OUT As Motherboard = Motherboard\_Frame.ResultObject

computer.Motherboard = Motherboard\_OUT

Motherboard\_Frame.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(0)

If treeNode\_Son.Nodes.Count = 0 Then

treeNode\_Son.Nodes.Add(computer.Motherboard.Manufacturer + " " + computer.Motherboard.Soket + " " + computer.Motherboard.Form\_factor)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(computer.Motherboard.Manufacturer + " " + computer.Motherboard.Soket + " " + computer.Motherboard.Form\_factor)

End If

treeNode\_Son.ExpandAll()

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

Motherboard\_Frame.Dispose()

result = Nothing

End If

End Sub

Private Sub Butt\_Frame\_Click(sender As Object, e As EventArgs) Handles Butt\_Frame.Click

Dim result As DialogResult = FRAME\_FRAME.ShowDialog()

If result = DialogResult.OK Then

Dim Frame\_OUT As Frame = FRAME\_FRAME.ResultObject

computer.Frame = Frame\_OUT

FRAME\_FRAME.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(7)

If treeNode\_Son.Nodes.Count = 0 Then

treeNode\_Son.Nodes.Add(computer.Frame.Manufacturer + " " + computer.Frame.Form\_factor + " " + computer.Frame.Color)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(computer.Frame.Manufacturer + " " + computer.Frame.Form\_factor + " " + computer.Frame.Color)

End If

treeNode\_Son.ExpandAll()

If computer.Motherboard IsNot Nothing Then

If computer.Motherboard.Form\_factor <> computer.Frame.Form\_factor Then

Dim treeNode\_Son\_X As TreeNode = treeNode.Nodes(7)

treeNode\_Son\_X.Nodes.Clear()

computer.Frame = Nothing

MessageBox.Show("Форм фактор на материнской плате не совпадает с корпусом! Он будет сброшен, выберете новый!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error)

End If

End If

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

CPU\_Frame.Dispose()

result = Nothing

End If

End Sub

Private Sub butt\_core\_Click(sender As Object, e As EventArgs) Handles butt\_core.Click

Dim result As DialogResult = CPU\_Frame.ShowDialog()

If result = DialogResult.OK Then

butt\_disk.Enabled = True

butt\_GPU.Enabled = True

butt\_motherboard.Enabled = True

butt\_save.Enabled = True

butt\_summ.Enabled = True

Butt\_RAM.Enabled = True

Butt\_powerplant.Enabled = True

butt\_disk.Enabled = True

Butt\_cooling.Enabled = True

Butt\_Frame.Enabled = True

Dim CPU\_OUT As CPU = CPU\_Frame.ResultObject

computer.CPU = CPU\_OUT

CPU\_Frame.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(1)

If treeNode\_Son.Nodes.Count = 0 Then

treeNode\_Son.Nodes.Add(computer.CPU.Manufacturer + " " + computer.CPU.Model)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(computer.CPU.Manufacturer + " " + computer.CPU.Model)

End If

treeNode\_Son.ExpandAll()

If computer.Motherboard IsNot Nothing Then

If computer.CPU.Soket <> computer.Motherboard.Soket Then

Dim treeNode\_Son\_X As TreeNode = treeNode.Nodes(0)

treeNode\_Son\_X.Nodes.Clear()

computer.Motherboard = Nothing

MessageBox.Show("Сокет на процессоре не совпадает с материнской платой! Она будет сброшена, выберете новую!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error)

End If

End If

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

CPU\_Frame.Dispose()

result = Nothing

End If

End Sub

Private Sub Butt\_powerplant\_Click(sender As Object, e As EventArgs) Handles Butt\_powerplant.Click

Dim result As DialogResult = Powerplant\_Frame.ShowDialog()

If result = DialogResult.OK Then

Dim Powerplant\_OUT As Powerplant = Powerplant\_Frame.ResultObject

computer.Powerplant = Powerplant\_OUT

Powerplant\_Frame.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(5)

If treeNode\_Son.Nodes.Count = 0 Then

treeNode\_Son.Nodes.Add(computer.Powerplant.Manufacturer + " " + computer.Powerplant.Sertif)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(computer.Powerplant.Manufacturer + " " + computer.Powerplant.Sertif)

End If

treeNode\_Son.ExpandAll()

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

Powerplant\_Frame.Dispose()

result = Nothing

End If

End Sub

Private Sub Butt\_cooling\_Click(sender As Object, e As EventArgs) Handles Butt\_cooling.Click

Dim result As DialogResult = Cooling\_Frame.ShowDialog()

If result = DialogResult.OK Then

Dim Cooling\_OUT As Cooling = Cooling\_Frame.ResultObject

computer.Cooling = Cooling\_OUT

Cooling\_Frame.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(6)

If treeNode\_Son.Nodes.Count = 0 Then

treeNode\_Son.Nodes.Add(computer.Cooling.Manufacturer + " " + computer.Cooling.Type)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(computer.Cooling.Manufacturer + " " + computer.Cooling.Type)

End If

treeNode\_Son.ExpandAll()

If computer.CPU Is Nothing Then

If Array.Find(computer.Cooling.Soket.Remove(" ").Trim.Split(","), Function(s) s = computer.CPU.Soket) IsNot Nothing Then

Dim treeNode\_Son\_X As TreeNode = treeNode.Nodes(0)

treeNode\_Son\_X.Nodes.Clear()

computer.Cooling = Nothing

MessageBox.Show("Сокет на процессоре не совпадает с охлаждением! Она будет сброшена, выберете новую!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error)

End If

End If

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

Cooling\_Frame.Dispose()

result = Nothing

End If

End Sub

Private Sub Butt\_RAM\_Click(sender As Object, e As EventArgs) Handles Butt\_RAM.Click

Dim result As DialogResult = RAM\_Frame.ShowDialog()

If result = DialogResult.OK Then

Dim RAM\_OUT As RAM = RAM\_Frame.ResultObject

computer.Ram = RAM\_OUT

RAM\_Frame.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(3)

If treeNode\_Son.Nodes.Count = 0 Then

treeNode\_Son.Nodes.Add(computer.Ram.Manufacturer + " " + computer.Ram.Type\_RAM + " " + computer.Ram.Sum\_RAM.ToString)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(computer.Ram.Manufacturer + " " + computer.Ram.Type\_RAM + " " + computer.Ram.Sum\_RAM.ToString)

End If

treeNode\_Son.ExpandAll()

If computer.Motherboard IsNot Nothing Then

Dim treeNode\_Son\_X As TreeNode = treeNode.Nodes(0)

treeNode\_Son\_X.Nodes.Clear()

computer.Ram = Nothing

MessageBox.Show("Тип на оперативной памяти не совпадает с материнской платой! Она будет сброшена, выберете новую!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error)

End If

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

RAM\_Frame.Dispose()

result = Nothing

End If

End Sub

Private Sub butt\_disk\_Click(sender As Object, e As EventArgs) Handles butt\_disk.Click

Dim result As DialogResult = Disk\_Frame.ShowDialog()

If result = DialogResult.OK Then

Dim disk\_OUT As Disk = Disk\_Frame.ResultObject

computer.Disk = disk\_OUT

Disk\_Frame.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(4)

If treeNode\_Son.Nodes.Count < 1 Then

treeNode\_Son.Nodes.Add(disk\_OUT.Manufacturer + " " + disk\_OUT.V\_of\_disk.ToString)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(disk\_OUT.Manufacturer + " " + disk\_OUT.V\_of\_disk.ToString)

End If

treeNode\_Son.ExpandAll()

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

Disk\_Frame.Dispose()

result = Nothing

End If

End Sub

Private Sub butt\_GPU\_Click(sender As Object, e As EventArgs) Handles butt\_GPU.Click

Dim result As DialogResult = GPU\_Frame.ShowDialog()

If result = DialogResult.OK Then

Dim GPU\_OUT As GPU = GPU\_Frame.ResultObject

computer.GPU = GPU\_OUT

GPU\_Frame.Dispose()

result = Nothing

Dim treeNode As TreeNode = Tree.Nodes(0)

Dim treeNode\_Son As TreeNode = treeNode.Nodes(2)

If treeNode\_Son.Nodes.Count = 0 Then

treeNode\_Son.Nodes.Add(computer.GPU.Manufacturer + " " + computer.GPU.V\_GPU.ToString + " " + computer.GPU.Model)

Else

treeNode\_Son.Nodes.Clear()

treeNode\_Son.Nodes.Add(computer.GPU.Manufacturer + " " + computer.GPU.V\_GPU.ToString + " " + computer.GPU.Model)

End If

treeNode\_Son.ExpandAll()

ElseIf result = DialogResult.Cancel Then

MessageBox.Show("Вы не сохранили выбор!", "Внимание!", MessageBoxButtons.OK, MessageBoxIcon.Information)

GPU\_Frame.Dispose()

result = Nothing

End If

End Sub

Dim cost As Integer

Private Sub butt\_summ\_Click(sender As Object, e As EventArgs) Handles butt\_summ.Click

If computer.Cooling IsNot Nothing Then

cost += computer.Cooling.Cost

End If

If computer.CPU IsNot Nothing Then

cost += computer.CPU.Cost

End If

If computer.Disk IsNot Nothing Then

cost += computer.Disk.Cost

End If

If computer.Frame IsNot Nothing Then

cost += computer.Frame.Cost

End If

If computer.GPU IsNot Nothing Then

cost += computer.GPU.Cost

End If

If computer.Motherboard IsNot Nothing Then

cost += computer.Motherboard.Cost

End If

If computer.Powerplant IsNot Nothing Then

cost += computer.Powerplant.Cost

End If

If computer.Ram IsNot Nothing Then

cost += computer.Ram.Cost

End If

txt\_summ.Text = cost.ToString + "руб"

End Sub

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

Dim filePath As String = "file.txt"

Dim textToSave As String = $"Computer:"

if computer.Cooling IsNot Nothing Then

textToSave = textToSave + $"

Cooling:

Id - {computer.Cooling.Id}

Manufacturer - {computer.Cooling.Manufacturer}

Weight - {computer.Cooling.Weight}

Type - {computer.Cooling.Type}

Soket - {computer.Cooling.Soket}

For\_what - {computer.Cooling.For\_what}

TDP - {computer.Cooling.TDP}

Length - {computer.Cooling.Length}

Cost - {computer.Cooling.Cost}"

End If

If computer.CPU IsNot Nothing Then

textToSave = textToSave + $"

CPU:

Id - {computer.CPU.Id}

Manufacturer - {computer.CPU.Manufacturer}

Model - {computer.CPU.Model}

Soket - {computer.CPU.Soket}

Sistem\_cooling\_in - {computer.CPU.Sistem\_cooling\_in}

Cores - {computer.CPU.Cores}

Process\_cores - {computer.CPU.Process\_cores}

Max\_kol\_potok - {computer.CPU.Max\_kol\_potok}

KESH - {computer.CPU.KESH}

Baze\_speed - {computer.CPU.Baze\_speed}

Max\_speed - {computer.CPU.Max\_speed}

Type\_RAM - {computer.CPU.Type\_RAM}

Max\_speed\_RAM - {computer.CPU.Max\_speed\_RAM}

Max\_RAM - {computer.CPU.Max\_RAM}

TDP - {computer.CPU.TDP}

In\_GPU - {computer.CPU.In\_GPU}

Control\_PCI - {computer.CPU.Control\_PCI}

Kol\_lines - {computer.CPU.Kol\_lines}

Cost - {computer.CPU.Cost}"

End If

If computer.Disk IsNot Nothing Then

textToSave = textToSave + $"

Disk:

Id - {computer.Disk.Id}

Manufacturer - {computer.Disk.Manufacturer}

Weight - {computer.Disk.Weight}

Type - {computer.Disk.Type}

V\_of\_disk - {computer.Disk.V\_of\_disk}

Interface - {computer.Disk.Interface\_q}

Max\_speed\_read - {computer.Disk.Max\_speed\_read}

Max\_speed\_write - {computer.Disk.Max\_speed\_write}

Max\_V\_write - {computer.Disk.Max\_V\_write}

Length - {computer.Disk.Length}

Cost - {computer.Disk.Cost}"

End If

If computer.Frame IsNot Nothing Then

textToSave = textToSave + $"

Frame:

Id - {computer.Frame.Id}

Manufacturer - {computer.Frame.Manufacturer}

Hight - {computer.Frame.Hight}

Width - {computer.Frame.Width}

Length - {computer.Frame.Length}

Weight - {computer.Frame.Weight}

Color - {computer.Frame.Color}

Material - {computer.Frame.Material}

Form\_factor - {computer.Frame.Form\_factor}

Max\_lenght\_of\_GPU - {computer.Frame.Max\_lenght\_of\_GPU}

Water\_cooling - {computer.Frame.Water\_cooling}

Cost - {computer.Frame.Cost}"

End If

If computer.GPU IsNot Nothing Then

textToSave = textToSave + $"

GPU:

Id - {computer.GPU.Id}

Manufacturer - {computer.GPU.Manufacturer}

Weight - {computer.GPU.Weight}

Color - {computer.GPU.Color}

Model - {computer.GPU.Model}

Graphic\_unit - {computer.GPU.Graphic\_unit}

Frequency - {computer.GPU.Frequency}

Turbofrequency - {computer.GPU.Turbofrequency}

DLSS - {computer.GPU.DLSS}

V\_GPU - {computer.GPU.V\_GPU}

Type\_of\_memory - {computer.GPU.Type\_of\_memory}

Type\_of\_ports - {computer.GPU.Type\_of\_ports}

Kol\_ports - {computer.GPU.Kol\_ports}

Interface - {computer.GPU.Interface\_q}

Dop\_power - {computer.GPU.Dop\_power}

Type\_of\_cooling - {computer.GPU.Type\_of\_cooling}

LED - {computer.GPU.LED}

Cost - {computer.GPU.Cost}"

End If

If computer.Motherboard IsNot Nothing Then

textToSave = textToSave + $"

Motherboard:

Id - {computer.Motherboard.Id}

Manufacturer - {computer.Motherboard.Manufacturer}

Weight - {computer.Motherboard.Weight}

Color - {computer.Motherboard.Color}

Length - {computer.Motherboard.Length}

Width - {computer.Motherboard.Width}

Soket - {computer.Motherboard.Soket}

Type\_RAM - {computer.Motherboard.Type\_RAM}

Max\_V\_RAM - {computer.Motherboard.Max\_V\_RAM}

Max\_speed\_RAM - {computer.Motherboard.Max\_speed\_RAM}

Kol\_slots\_RAM - {computer.Motherboard.Kol\_slots\_RAM}

PCI\_version - {computer.Motherboard.PCI\_version}

Kol\_M2 - {computer.Motherboard.Kol\_M2}

USB - {computer.Motherboard.USB}

Video\_port - {computer.Motherboard.Video\_port}

Form\_factor - {computer.Motherboard.Form\_factor}

Esernet\_adapter - {computer.Motherboard.Esernet\_adapter}

Esernet\_speed - {computer.Motherboard.Esernet\_speed}

Cost - {computer.Motherboard.Cost}"

End If

If computer.Powerplant IsNot Nothing Then

textToSave = textToSave + $"Powerplant:

Id - {computer.Powerplant.Id}

Manufacturer - {computer.Powerplant.Manufacturer}

Weight - {computer.Powerplant.Weight}

Color - {computer.Powerplant.Color}

Сapacity - {computer.Powerplant.Сapacity}

Form\_factor - {computer.Powerplant.Form\_factor}

Power\_CPU - {computer.Powerplant.Power\_CPU}

Power\_GPU - {computer.Powerplant.Power\_GPU}

15\_pin\_SATA - {computer.Powerplant.l\_15\_pin\_SATA}

4\_pin\_Molex - {computer.Powerplant.l\_4\_pin\_Molex}

Cooling\_system - {computer.Powerplant.Cooling\_system}

Sertif - {computer.Powerplant.Sertif}

Cost - {computer.Powerplant.Cost}"

End If

If computer.Ram IsNot Nothing Then

textToSave = textToSave + $"

Ram:

Id - {computer.Ram.Id}

Manufacturer - {computer.Ram.Manufacturer}

Color - {computer.Ram.Color}

Type\_RAM - {computer.Ram.Type\_RAM}

Sum\_RAM - {computer.Ram.Sum\_RAM}

One\_RAM - {computer.Ram.One\_RAM}

Kol\_RAM - {computer.Ram.Kol\_RAM}

Speed\_RAM - {computer.Ram.Speed\_RAM}

Cooler\_RAM - {computer.Ram.Cooler\_RAM}

Cost - {computer.Ram.Cost}"

End If

textToSave = textToSave + "

Итог: " + cost.ToString + "руб."

FileOpen(1, filePath, OpenMode.Output)

Print(1, textToSave) ' Запись строки

MessageBox.Show($"Сборка сохранена.

Путь:{filePath}", "Успех!", MessageBoxButtons.OK, MessageBoxIcon.Information)

FileClose(1) ' Закрытие файла

End Sub

Private Sub Sborka\_FormClosing(sender As Object, e As FormClosingEventArgs) Handles MyBase.FormClosing

Application.Exit()

End Sub

Private Sub Sborka\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

butt\_disk.Enabled = False

butt\_GPU.Enabled = False

butt\_motherboard.Enabled = False

butt\_save.Enabled = False

butt\_summ.Enabled = False

Butt\_RAM.Enabled = False

Butt\_powerplant.Enabled = False

Butt\_Frame.Enabled = False

butt\_disk.Enabled = False

Butt\_cooling.Enabled = False

Me.computer = New Computer

Tree.ExpandAll()

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

Me.Close()

End Sub

Private Sub Tree\_AfterSelect(sender As Object, e As TreeViewEventArgs) Handles Tree.AfterSelect

If Tree.SelectedNode.Tag Is Nothing Then

Dim Parent As TreeNode = Tree.SelectedNode.Parent

Select Case Tree.SelectedNode.Parent.Tag

Case 1

computer.Motherboard = Nothing

Parent.Nodes.Clear()

Case 2

computer.CPU = Nothing

Parent.Nodes.Clear()

Case 3

computer.GPU = Nothing

Parent.Nodes.Clear()

Case 4

computer.Ram = Nothing

Parent.Nodes.Clear()

Case 5

computer.Disk = Nothing

Parent.Nodes.Clear()

Case 6

computer.Powerplant = Nothing

Parent.Nodes.Clear()

Case 7

computer.Cooling = Nothing

Parent.Nodes.Clear()

Case 8

computer.Frame = Nothing

Parent.Nodes.Clear()

End Select

End If

End Sub

End Class

Imports System.Security.Cryptography

Public Class Computer

Private \_Cooling As Cooling

Private \_CPU As CPU

Private \_Disk As Disk

Private \_Frame As Frame

Private \_GPU As GPU

Private \_Motherboard As Motherboard

Private \_Powerplant As Powerplant

Private \_Ram As RAM

Public Sub New()

End Sub

Public Sub New(Cooling As Cooling, CPU As CPU, Disk As Disk, Frame As Frame, GPU As GPU, Motherboard As Motherboard, Powerplant As Powerplant, Ram As RAM)

\_Cooling = Cooling

\_CPU = CPU

\_Disk = Disk

\_Frame = Frame

\_GPU = GPU

\_Motherboard = Motherboard

\_Powerplant = Powerplant

\_Ram = Ram

End Sub

Public Property GPU() As GPU

Get

Return \_GPU

End Get

Set(value As GPU)

\_GPU = value

End Set

End Property

Public Property Motherboard() As Motherboard

Get

Return \_Motherboard

End Get

Set(value As Motherboard)

\_Motherboard = value

End Set

End Property

Public Property Powerplant() As Powerplant

Get

Return \_Powerplant

End Get

Set(value As Powerplant)

\_Powerplant = value

End Set

End Property

Public Property Ram() As RAM

Get

Return \_Ram

End Get

Set(value As RAM)

\_Ram = value

End Set

End Property

Public Property Disk() As Disk

Get

Return \_Disk

End Get

Set(value As Disk)

\_Disk = value

End Set

End Property

Public Property Frame() As Frame

Get

Return \_Frame

End Get

Set(value As Frame)

\_Frame = value

End Set

End Property

Public Property Cooling() As Cooling

Get

Return \_Cooling

End Get

Set(value As Cooling)

\_Cooling = value

End Set

End Property

Public Property CPU() As CPU

Get

Return \_CPU

End Get

Set(value As CPU)

\_CPU = value

End Set

End Property

End Class

Public Class RAM

' Объявление переменных класса

Private \_Id As Integer

Private \_Manufacturer As String

Private \_Color As String

Private \_Type\_RAM As String ' Например, DDR4, DDR5

Private \_Sum\_RAM As Integer ' Общий объем оперативной памяти в ГБ

Private \_One\_RAM As Integer ' Объем одного модуля памяти в ГБ

Private \_Kol\_RAM As Integer ' Количество модулей памяти

Private \_Speed\_RAM As Integer ' Частота памяти в МГц

Private \_Cooler\_RAM As Boolean ' Наличие радиатора/охлаждения

Private \_Cost As Decimal

' Конструктор класса

Public Sub New(Id As Integer, Manufacturer As String, Color As String, Type\_RAM As String, Sum\_RAM As Integer, One\_RAM As Integer, Kol\_RAM As Integer, Speed\_RAM As Integer, Cooler\_RAM As Boolean, Cost As Decimal)

\_Id = Id

\_Manufacturer = Manufacturer

\_Color = Color

\_Type\_RAM = Type\_RAM

\_Sum\_RAM = Sum\_RAM

\_One\_RAM = One\_RAM

\_Kol\_RAM = Kol\_RAM

\_Speed\_RAM = Speed\_RAM

\_Cooler\_RAM = Cooler\_RAM

\_Cost = Cost

End Sub

' Property для каждой переменной класса

Public Property Id As Integer

Get

Return \_Id

End Get

Set(value As Integer)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Color As String

Get

Return \_Color

End Get

Set(value As String)

\_Color = value

End Set

End Property

Public Property Type\_RAM As String

Get

Return \_Type\_RAM

End Get

Set(value As String)

\_Type\_RAM = value

End Set

End Property

Public Property Sum\_RAM As Integer

Get

Return \_Sum\_RAM

End Get

Set(value As Integer)

\_Sum\_RAM = value

End Set

End Property

Public Property One\_RAM As Integer

Get

Return \_One\_RAM

End Get

Set(value As Integer)

\_One\_RAM = value

End Set

End Property

Public Property Kol\_RAM As Integer

Get

Return \_Kol\_RAM

End Get

Set(value As Integer)

\_Kol\_RAM = value

End Set

End Property

Public Property Speed\_RAM As Integer

Get

Return \_Speed\_RAM

End Get

Set(value As Integer)

\_Speed\_RAM = value

End Set

End Property

Public Property Cooler\_RAM As Boolean

Get

Return \_Cooler\_RAM

End Get

Set(value As Boolean)

\_Cooler\_RAM = value

End Set

End Property

Public Property Cost As Decimal

Get

Return \_Cost

End Get

Set(value As Decimal)

\_Cost = value

End Set

End Property

End Class

Public Class Powerplant

' Объявление переменных класса

Private \_Id As Integer

Private \_Manufacturer As String

Private \_Weight As Double ' Вес в кг

Private \_Color As String

Private \_Сapacity As Integer ' Мощность в Ваттах

Private \_Form\_factor As String ' Форм-фактор (ATX, SFX и т.д.)

Private \_Power\_CPU As String ' Тип и количество разъемов питания CPU (например, 4+4 pin)

Private \_Power\_GPU As String ' Тип и количество разъемов питания GPU (например, 6+2 pin x 2)

Private \_15\_pin\_SATA As String ' Количество 15-pin SATA разъемов

Private \_4\_pin\_Molex As String ' Количество 4-pin Molex разъемов

Private \_Cooling\_system As String ' Тип системы охлаждения (активная, пассивная, гибридная)

Private \_Sertif As String ' Сертификация 80 PLUS (например, 80 PLUS Bronze, 80 PLUS Gold)

Private \_Cost As Decimal

' Конструктор класса

Public Sub New(Id As Integer, Manufacturer As String, Weight As Double, Color As String, Сapacity As Integer, Form\_factor As String, Power\_CPU As String, Power\_GPU As String, l\_15\_pin\_SATA As String, l\_4\_pin\_Molex As String, Cooling\_system As String, Sertif As String, Cost As Decimal)

\_Id = Id

\_Manufacturer = Manufacturer

\_Weight = Weight

\_Color = Color

\_Сapacity = Сapacity

\_Form\_factor = Form\_factor

\_Power\_CPU = Power\_CPU

\_Power\_GPU = Power\_GPU

\_15\_pin\_SATA = l\_15\_pin\_SATA

\_4\_pin\_Molex = l\_4\_pin\_Molex

\_Cooling\_system = Cooling\_system

\_Sertif = Sertif

\_Cost = Cost

End Sub

' Property для каждой переменной класса

Public Property Id As Integer

Get

Return \_Id

End Get

Set(value As Integer)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Weight As Double

Get

Return \_Weight

End Get

Set(value As Double)

\_Weight = value

End Set

End Property

Public Property Color As String

Get

Return \_Color

End Get

Set(value As String)

\_Color = value

End Set

End Property

Public Property Сapacity As Integer

Get

Return \_Сapacity

End Get

Set(value As Integer)

\_Сapacity = value

End Set

End Property

Public Property Form\_factor As String

Get

Return \_Form\_factor

End Get

Set(value As String)

\_Form\_factor = value

End Set

End Property

Public Property Power\_CPU As String

Get

Return \_Power\_CPU

End Get

Set(value As String)

\_Power\_CPU = value

End Set

End Property

Public Property Power\_GPU As String

Get

Return \_Power\_GPU

End Get

Set(value As String)

\_Power\_GPU = value

End Set

End Property

Public Property l\_15\_pin\_SATA As String

Get

Return \_15\_pin\_SATA

End Get

Set(value As String)

\_15\_pin\_SATA = value

End Set

End Property

Public Property l\_4\_pin\_Molex As String

Get

Return \_4\_pin\_Molex

End Get

Set(value As String)

\_4\_pin\_Molex = value

End Set

End Property

Public Property Cooling\_system As String

Get

Return \_Cooling\_system

End Get

Set(value As String)

\_Cooling\_system = value

End Set

End Property

Public Property Sertif As String

Get

Return \_Sertif

End Get

Set(value As String)

\_Sertif = value

End Set

End Property

Public Property Cost As Decimal

Get

Return \_Cost

End Get

Set(value As Decimal)

\_Cost = value

End Set

End Property

End Class

Public Class Motherboard

' Объявление переменных класса

Private \_Id As Integer

Private \_Manufacturer As String

Private \_Weight As Double ' в кг

Private \_Color As String

Private \_Length As Double ' в см

Private \_Width As Double

Private \_Soket As String ' Например, LGA 1700, AM5

Private \_Type\_RAM As String ' Например, DDR4, DDR5

Private \_Max\_V\_RAM As Integer ' Максимальный объем ОЗУ в ГБ

Private \_Max\_speed\_RAM As Integer ' Максимальная частота ОЗУ в МГц

Private \_Kol\_slots\_RAM As Integer ' Количество слотов ОЗУ

Private \_PCI\_version As String ' Версия PCI Express (например, PCIe 4.0, PCIe 5.0)

Private \_Kol\_M2 As Integer ' Количество слотов M.2

Private \_USB As Integer 'количество портов USB

Private \_Video\_port As String ' Тип и количество видеопортов (например, HDMI x 1, DisplayPort x 1)

Private \_Form\_factor As String ' Форм-фактор (например, ATX, Micro-ATX, Mini-ITX)

Private \_Esernet\_adapter As String ' Модель Ethernet-адаптера (например, Intel I225-V)

Private \_Esernet\_speed As String ' Скорость Ethernet-адаптера в Мбит/с (например, 2500)

Private \_Cost As Decimal

' Конструктор класса

Public Sub New(Id As Integer, Manufacturer As String, Weight As Double, Color As String, Length As Double, Width As Double, Soket As String, Type\_RAM As String, Max\_V\_RAM As Integer, Max\_speed\_RAM As Integer, Kol\_slots\_RAM As Integer, PCI\_version As String, Kol\_M2 As Integer, USB As Integer, Video\_port As String, Form\_factor As String, Esernet\_adapter As String, Esernet\_speed As String, Cost As Decimal)

\_Id = Id

\_Manufacturer = Manufacturer

\_Weight = Weight

\_Color = Color

\_Length = Length

\_Width = Width

\_Soket = Soket

\_Type\_RAM = Type\_RAM

\_Max\_V\_RAM = Max\_V\_RAM

\_Max\_speed\_RAM = Max\_speed\_RAM

\_Kol\_slots\_RAM = Kol\_slots\_RAM

\_PCI\_version = PCI\_version

\_Kol\_M2 = Kol\_M2

\_USB = USB

\_Video\_port = Video\_port

\_Form\_factor = Form\_factor

\_Esernet\_adapter = Esernet\_adapter

\_Esernet\_speed = Esernet\_speed

\_Cost = Cost

End Sub

' Property для каждой переменной класса

Public Property Id As Integer

Get

Return \_Id

End Get

Set(value As Integer)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Weight As Double

Get

Return \_Weight

End Get

Set(value As Double)

\_Weight = value

End Set

End Property

Public Property Color As String

Get

Return \_Color

End Get

Set(value As String)

\_Color = value

End Set

End Property

Public Property Length As Double

Get

Return \_Length

End Get

Set(value As Double)

\_Length = value

End Set

End Property

Public Property Width As Double

Get

Return \_Width

End Get

Set(value As Double)

\_Width = value

End Set

End Property

Public Property Soket As String

Get

Return \_Soket

End Get

Set(value As String)

\_Soket = value

End Set

End Property

Public Property Type\_RAM As String

Get

Return \_Type\_RAM

End Get

Set(value As String)

\_Type\_RAM = value

End Set

End Property

Public Property Max\_V\_RAM As Integer

Get

Return \_Max\_V\_RAM

End Get

Set(value As Integer)

\_Max\_V\_RAM = value

End Set

End Property

Public Property Max\_speed\_RAM As Integer

Get

Return \_Max\_speed\_RAM

End Get

Set(value As Integer)

\_Max\_speed\_RAM = value

End Set

End Property

Public Property Kol\_slots\_RAM As Integer

Get

Return \_Kol\_slots\_RAM

End Get

Set(value As Integer)

\_Kol\_slots\_RAM = value

End Set

End Property

Public Property PCI\_version As String

Get

Return \_PCI\_version

End Get

Set(value As String)

\_PCI\_version = value

End Set

End Property

Public Property Kol\_M2 As Integer

Get

Return \_Kol\_M2

End Get

Set(value As Integer)

\_Kol\_M2 = value

End Set

End Property

Public Property USB As Integer

Get

Return \_USB

End Get

Set(value As Integer)

\_USB = value

End Set

End Property

Public Property Video\_port As String

Get

Return \_Video\_port

End Get

Set(value As String)

\_Video\_port = value

End Set

End Property

Public Property Form\_factor As String

Get

Return \_Form\_factor

End Get

Set(value As String)

\_Form\_factor = value

End Set

End Property

Public Property Esernet\_adapter As String

Get

Return \_Esernet\_adapter

End Get

Set(value As String)

\_Esernet\_adapter = value

End Set

End Property

Public Property Esernet\_speed As String

Get

Return \_Esernet\_speed

End Get

Set(value As String)

\_Esernet\_speed = value

End Set

End Property

Public Property Cost As Decimal

Get

Return \_Cost

End Get

Set(value As Decimal)

\_Cost = value

End Set

End Property

End Class

Public Class GPU

' Объявление переменных класса

Private \_Id As Integer

Private \_Manufacturer As String

Private \_Weight As Double ' в кг

Private \_Color As String

Private \_Model As String

Private \_Graphic\_unit As String ' Чипсет (например, NVIDIA GeForce RTX 3080)

Private \_Frequency As Integer ' Базовая частота в МГц

Private \_Turbofrequency As Integer ' Турбо частота в МГц

Private \_DLSS As Boolean ' Поддержка DLSS

Private \_V\_GPU As Integer ' Объем видеопамяти в ГБ

Private \_Type\_of\_memory As String ' Тип памяти (например, GDDR6X)

Private \_Type\_of\_ports As String ' Тип портов (например, HDMI, DisplayPort)

Private \_Kol\_ports As Integer ' Количество портов

Private \_Interface As String ' Интерфейс (например, PCIe 4.0)

Private \_Dop\_power As String ' Дополнительное питание (например, 8+8 pin)

Private \_Type\_of\_cooling As String ' Тип охлаждения (воздушное, водяное)

Private \_LED As Boolean ' Наличие LED-подсветки

Private \_Cost As Decimal

' Конструктор класса

Public Sub New(Id As Integer, Manufacturer As String, Weight As Double, Color As String, Model As String, Graphic\_unit As String, Frequency As Integer, Turbofrequency As Integer, DLSS As Boolean, V\_GPU As Integer, Type\_of\_memory As String, Type\_of\_ports As String, Kol\_ports As Integer, Interface\_q As String, Dop\_power As String, Type\_of\_cooling As String, LED As Boolean, Cost As Decimal)

\_Id = Id

\_Manufacturer = Manufacturer

\_Weight = Weight

\_Color = Color

\_Model = Model

\_Graphic\_unit = Graphic\_unit

\_Frequency = Frequency

\_Turbofrequency = Turbofrequency

\_DLSS = DLSS

\_V\_GPU = V\_GPU

\_Type\_of\_memory = Type\_of\_memory

\_Type\_of\_ports = Type\_of\_ports

\_Kol\_ports = Kol\_ports

\_Interface = Interface\_q

\_Dop\_power = Dop\_power

\_Type\_of\_cooling = Type\_of\_cooling

\_LED = LED

\_Cost = Cost

End Sub

' Property для каждой переменной класса

Public Property Id As Integer

Get

Return \_Id

End Get

Set(value As Integer)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Weight As Double

Get

Return \_Weight

End Get

Set(value As Double)

\_Weight = value

End Set

End Property

Public Property Color As String

Get

Return \_Color

End Get

Set(value As String)

\_Color = value

End Set

End Property

Public Property Model As String

Get

Return \_Model

End Get

Set(value As String)

\_Model = value

End Set

End Property

Public Property Graphic\_unit As String

Get

Return \_Graphic\_unit

End Get

Set(value As String)

\_Graphic\_unit = value

End Set

End Property

Public Property Frequency As Integer

Get

Return \_Frequency

End Get

Set(value As Integer)

\_Frequency = value

End Set

End Property

Public Property Turbofrequency As Integer

Get

Return \_Turbofrequency

End Get

Set(value As Integer)

\_Turbofrequency = value

End Set

End Property

Public Property DLSS As Boolean

Get

Return \_DLSS

End Get

Set(value As Boolean)

\_DLSS = value

End Set

End Property

Public Property V\_GPU As Integer

Get

Return \_V\_GPU

End Get

Set(value As Integer)

\_V\_GPU = value

End Set

End Property

Public Property Type\_of\_memory As String

Get

Return \_Type\_of\_memory

End Get

Set(value As String)

\_Type\_of\_memory = value

End Set

End Property

Public Property Type\_of\_ports As String

Get

Return \_Type\_of\_ports

End Get

Set(value As String)

\_Type\_of\_ports = value

End Set

End Property

Public Property Kol\_ports As Integer

Get

Return \_Kol\_ports

End Get

Set(value As Integer)

\_Kol\_ports = value

End Set

End Property

Public Property Interface\_q As String

Get

Return \_Interface

End Get

Set(value As String)

\_Interface = value

End Set

End Property

Public Property Dop\_power As String

Get

Return \_Dop\_power

End Get

Set(value As String)

\_Dop\_power = value

End Set

End Property

Public Property Type\_of\_cooling As String

Get

Return \_Type\_of\_cooling

End Get

Set(value As String)

\_Type\_of\_cooling = value

End Set

End Property

Public Property LED As Boolean

Get

Return \_LED

End Get

Set(value As Boolean)

\_LED = value

End Set

End Property

Public Property Cost As Decimal

Get

Return \_Cost

End Get

Set(value As Decimal)

\_Cost = value

End Set

End Property

End Class

Public Class Frame

' Объявление переменных класса

Private \_Id As Integer

Private \_Manufacturer As String

Private \_Hight As Double ' в мм

Private \_Width As Double ' в мм

Private \_Length As Double ' в мм

Private \_Weight As Double ' в кг

Private \_Color As String

Private \_Material As String

Private \_Form\_factor As String ' ATX, Micro-ATX, Mini-ITX и т.д.

Private \_Max\_lenght\_of\_GPU As Integer ' в мм

Private \_Water\_cooling As Boolean

Private \_Cost As Decimal

' Конструктор класса

Public Sub New(Id As Integer, Manufacturer As String, Hight As Double, Width As Double, Length As Double, Weight As Double, Color As String, Material As String, Form\_factor As String, Max\_lenght\_of\_GPU As Integer, Water\_cooling As Boolean, Cost As Decimal)

\_Id = Id

\_Manufacturer = Manufacturer

\_Hight = Hight

\_Width = Width

\_Length = Length

\_Weight = Weight

\_Color = Color

\_Material = Material

\_Form\_factor = Form\_factor

\_Max\_lenght\_of\_GPU = Max\_lenght\_of\_GPU

\_Water\_cooling = Water\_cooling

\_Cost = Cost

End Sub

' Property для каждой переменной класса

Public Property Id As Integer

Get

Return \_Id

End Get

Set(value As Integer)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Hight As Double

Get

Return \_Hight

End Get

Set(value As Double)

\_Hight = value

End Set

End Property

Public Property Width As Double

Get

Return \_Width

End Get

Set(value As Double)

\_Width = value

End Set

End Property

Public Property Length As Double

Get

Return \_Length

End Get

Set(value As Double)

\_Length = value

End Set

End Property

Public Property Weight As Double

Get

Return \_Weight

End Get

Set(value As Double)

\_Weight = value

End Set

End Property

Public Property Color As String

Get

Return \_Color

End Get

Set(value As String)

\_Color = value

End Set

End Property

Public Property Material As String

Get

Return \_Material

End Get

Set(value As String)

\_Material = value

End Set

End Property

Public Property Form\_factor As String

Get

Return \_Form\_factor

End Get

Set(value As String)

\_Form\_factor = value

End Set

End Property

Public Property Max\_lenght\_of\_GPU As Integer

Get

Return \_Max\_lenght\_of\_GPU

End Get

Set(value As Integer)

\_Max\_lenght\_of\_GPU = value

End Set

End Property

Public Property Water\_cooling As Boolean

Get

Return \_Water\_cooling

End Get

Set(value As Boolean)

\_Water\_cooling = value

End Set

End Property

Public Property Cost As Decimal

Get

Return \_Cost

End Get

Set(value As Decimal)

\_Cost = value

End Set

End Property

End Class

Public Class Disk

' Объявление переменных класса

Private \_Id As Integer

Private \_Manufacturer As String

Private \_Weight As Double

Private \_Type As String

Private \_V\_of\_disk As Integer ' Объем диска в ГБ или ТБ

Private \_Interface As String

Private \_Max\_speed\_read As Integer ' в МБ/с

Private \_Max\_speed\_write As Integer ' в МБ/с

Private \_Max\_V\_write As Integer ' ресурс записи в TBW (Терабайт записи)

Private \_Length As Double ' в мм

Private \_Cost As Decimal

' Конструктор класса

Public Sub New(Id As Integer, Manufacturer As String, Weight As Double, Type As String, V\_of\_disk As Integer, Interface\_q As String, Max\_speed\_read As Integer, Max\_speed\_write As Integer, Max\_V\_write As Integer, Length As Double, Cost As Decimal)

\_Id = Id

\_Manufacturer = Manufacturer

\_Weight = Weight

\_Type = Type

\_V\_of\_disk = V\_of\_disk

\_Interface = Interface\_q

\_Max\_speed\_read = Max\_speed\_read

\_Max\_speed\_write = Max\_speed\_write

\_Max\_V\_write = Max\_V\_write

\_Length = Length

\_Cost = Cost

End Sub

' Property для каждой переменной класса

Public Property Id As Integer

Get

Return \_Id

End Get

Set(value As Integer)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Weight As Double

Get

Return \_Weight

End Get

Set(value As Double)

\_Weight = value

End Set

End Property

Public Property Type As String

Get

Return \_Type

End Get

Set(value As String)

\_Type = value

End Set

End Property

Public Property V\_of\_disk As Integer

Get

Return \_V\_of\_disk

End Get

Set(value As Integer)

\_V\_of\_disk = value

End Set

End Property

Public Property Interface\_q As String

Get

Return \_Interface

End Get

Set(value As String)

\_Interface = value

End Set

End Property

Public Property Max\_speed\_read As Integer

Get

Return \_Max\_speed\_read

End Get

Set(value As Integer)

\_Max\_speed\_read = value

End Set

End Property

Public Property Max\_speed\_write As Integer

Get

Return \_Max\_speed\_write

End Get

Set(value As Integer)

\_Max\_speed\_write = value

End Set

End Property

Public Property Max\_V\_write As Integer

Get

Return \_Max\_V\_write

End Get

Set(value As Integer)

\_Max\_V\_write = value

End Set

End Property

Public Property Length As Double

Get

Return \_Length

End Get

Set(value As Double)

\_Length = value

End Set

End Property

Public Property Cost As Decimal

Get

Return \_Cost

End Get

Set(value As Decimal)

\_Cost = value

End Set

End Property

End Class

Public Class CPU

' Объявление переменных класса

Private \_Id As Integer

Private \_Manufacturer As String

Private \_Model As String

Private \_Soket As String

Private \_Sistem\_cooling\_in As Boolean

Private \_Cores As Integer

Private \_Process\_cores As Integer

Private \_Max\_kol\_potok As Integer

Private \_KESH As Double

Private \_Baze\_speed As Double

Private \_Max\_speed As Double

Private \_Type\_RAM As String

Private \_Max\_speed\_RAM As Integer

Private \_Max\_RAM As Integer

Private \_TDP As Integer

Private \_In\_GPU As String

Private \_Control\_PCI As String

Private \_Kol\_lines As Integer

Private \_Cost As Decimal

' Конструктор класса, принимающий все переменные

Public Sub New(Id As Integer, Manufacturer As String, Model As String, Soket As String, Sistem\_cooling\_in As Boolean, Cores As Integer, Process\_cores As Integer, Max\_kol\_potok As Integer, KESH As Double, Baze\_speed As Double, Max\_speed As Double, Type\_RAM As String, Max\_speed\_RAM As Integer, Max\_RAM As Integer, TDP As Integer, In\_GPU As String, Control\_PCI As String, Kol\_lines As Integer, Cost As Decimal)

\_Id = Id

\_Manufacturer = Manufacturer

\_Model = Model

\_Soket = Soket

\_Sistem\_cooling\_in = Sistem\_cooling\_in

\_Cores = Cores

\_Process\_cores = Process\_cores

\_Max\_kol\_potok = Max\_kol\_potok

\_KESH = KESH

\_Baze\_speed = Baze\_speed

\_Max\_speed = Max\_speed

\_Type\_RAM = Type\_RAM

\_Max\_speed\_RAM = Max\_speed\_RAM

\_Max\_RAM = Max\_RAM

\_TDP = TDP

\_In\_GPU = In\_GPU

\_Control\_PCI = Control\_PCI

\_Kol\_lines = Kol\_lines

\_Cost = Cost

End Sub

Public Sub New()

End Sub

' Property для каждой переменной класса

Public Property Id As Integer

Get

Return \_Id

End Get

Set(value As Integer)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Model As String

Get

Return \_Model

End Get

Set(value As String)

\_Model = value

End Set

End Property

Public Property Soket As String

Get

Return \_Soket

End Get

Set(value As String)

\_Soket = value

End Set

End Property

Public Property Sistem\_cooling\_in As Boolean

Get

Return \_Sistem\_cooling\_in

End Get

Set(value As Boolean)

\_Sistem\_cooling\_in = value

End Set

End Property

Public Property Cores As Integer

Get

Return \_Cores

End Get

Set(value As Integer)

\_Cores = value

End Set

End Property

Public Property Process\_cores As Integer

Get

Return \_Process\_cores

End Get

Set(value As Integer)

\_Process\_cores = value

End Set

End Property

Public Property Max\_kol\_potok As Integer

Get

Return \_Max\_kol\_potok

End Get

Set(value As Integer)

\_Max\_kol\_potok = value

End Set

End Property

Public Property KESH As Double

Get

Return \_KESH

End Get

Set(value As Double)

\_KESH = value

End Set

End Property

Public Property Baze\_speed As Double

Get

Return \_Baze\_speed

End Get

Set(value As Double)

\_Baze\_speed = value

End Set

End Property

Public Property Max\_speed As Double

Get

Return \_Max\_speed

End Get

Set(value As Double)

\_Max\_speed = value

End Set

End Property

Public Property Type\_RAM As String

Get

Return \_Type\_RAM

End Get

Set(value As String)

\_Type\_RAM = value

End Set

End Property

Public Property Max\_speed\_RAM As Integer

Get

Return \_Max\_speed\_RAM

End Get

Set(value As Integer)

\_Max\_speed\_RAM = value

End Set

End Property

Public Property Max\_RAM As Integer

Get

Return \_Max\_RAM

End Get

Set(value As Integer)

\_Max\_RAM = value

End Set

End Property

Public Property TDP As Integer

Get

Return \_TDP

End Get

Set(value As Integer)

\_TDP = value

End Set

End Property

Public Property In\_GPU As String

Get

Return \_In\_GPU

End Get

Set(value As String)

\_In\_GPU = value

End Set

End Property

Public Property Control\_PCI As String

Get

Return \_Control\_PCI

End Get

Set(value As String)

\_Control\_PCI = value

End Set

End Property

Public Property Kol\_lines As Integer

Get

Return \_Kol\_lines

End Get

Set(value As Integer)

\_Kol\_lines = value

End Set

End Property

Public Property Cost As Decimal

Get

Return \_Cost

End Get

Set(value As Decimal)

\_Cost = value

End Set

End Property

End Class

Imports System.Runtime.Intrinsics.Arm

Imports System.Windows.Forms.VisualStyles.VisualStyleElement.Rebar

Public Class Cooling

Private \_Id As String

Private \_Manufacturer As String

Private \_Weight As String

Private \_Type As String

Private \_Soket As String

Private \_For\_what As String

Private \_TDP As Integer

Private \_Length As Single

Private \_Width As Single

Private \_Cost As Single

Public Sub New(Id As String, Manufacturer As String, Weight As String, Type As String,

Soket As String, For\_what As String, TDP As Integer,

Length As Single, Width As Single, Cost As Single)

\_Id = Id

\_Manufacturer = Manufacturer

\_Weight = Weight

\_Type = Type

\_Soket = Soket

\_For\_what = For\_what

\_TDP = TDP

\_Length = Length

\_Width = Width

\_Cost = Cost

End Sub

Public Property Id As String

Get

Return \_Id

End Get

Set(value As String)

\_Id = value

End Set

End Property

Public Property Manufacturer As String

Get

Return \_Manufacturer

End Get

Set(value As String)

\_Manufacturer = value

End Set

End Property

Public Property Weight As String

Get

Return \_Weight

End Get

Set(value As String)

\_Weight = value

End Set

End Property

Public Property Type As String

Get

Return \_Type

End Get

Set(value As String)

\_Type = value

End Set

End Property

Public Property Soket As String

Get

Return \_Soket

End Get

Set(value As String)

\_Soket = value

End Set

End Property

Public Property For\_what As String

Get

Return \_For\_what

End Get

Set(value As String)

\_For\_what = value

End Set

End Property

Public Property TDP As Integer

Get

Return \_TDP

End Get

Set(value As Integer)

\_TDP = value

End Set

End Property

Public Property Length As Single

Get

Return \_Length

End Get

Set(value As Single)

\_Length = value

End Set

End Property

Public Property Width As Single

Get

Return \_Width

End Get

Set(value As Single)

\_Width = value

End Set

End Property

Public Property Cost As Single

Get

Return \_Cost

End Get

Set(value As Single)

\_Cost = value

End Set

End Property

End Class

Public Class Cooling\_Frame

Private \_resultObject As Cooling

Dim count1, count2, all(0 To 1) As Integer

Dim manu(), type\_cooling() As String

Dim intriger As Integer = 0

Public Property ResultObject() As Cooling ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As Cooling)

\_resultObject = value

End Set

End Property

Public Sub Obnov()

Dim whereClauses As New List(Of String)()

Dim command\_7 As String

If intriger < 2 Then

intriger += 1

Return

Else

CO.Open()

' Формируем условия WHERE

If Cmb\_manuf.Text() <> "Все" Then

whereClauses.Add($"Manufacture = '{Cmb\_manuf.SelectedItem}'")

End If

If Cmb\_type.Text() <> "Все" Then

whereClauses.Add($"Type = '{Cmb\_type.SelectedItem}'")

End If

If Sborka.computer.CPU IsNot Nothing Then

Dim s As String = Sborka.computer.CPU.Soket.ToString().Replace(" ", "")

whereClauses.Add($"Soket LIKE '%{s}%'")

whereClauses.Add($"TDP >= {Sborka.computer.CPU.TDP}")

End If

' Собираем итоговый SQL

command\_7 = "SELECT \* FROM COOLING"

If whereClauses.Count > 0 Then

command\_7 = command\_7 & " WHERE " & String.Join(" AND ", whereClauses)

End If

Dim COMMAND7 As New System.Data.OleDb.OleDbCommand(command\_7, CO)

Dim DATA7 As System.Data.OleDb.OleDbDataReader = COMMAND7.ExecuteReader(CommandBehavior.CloseConnection)

Dim dataTable As New DataTable()

dataTable.Load(DATA7)

DataGV\_1.DataSource = dataTable

CO.Close

End If

End Sub

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Private Sub Cmb\_manuf\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_manuf.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_type\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_type.SelectedIndexChanged

Obnov()

End Sub

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

CO.open()

Dim Manufacturer As String, Weight As String, Type As String,

Soket As String, For\_what As String, TDP As Integer,

Length As Single, Width As Single, Cost As Single

Dim choose\_Cooling As Cooling

Dim id As Integer = Nud\_id.Value

Dim command\_8 = $"SELECT \* FROM COOLING WHERE Id = '8\_{id}'"

Dim COMMAND8 As New System.Data.OleDb.OleDbCommand(command\_8, CO)

Dim DATA8 As System.Data.OleDb.OleDbDataReader = COMMAND8.ExecuteReader(CommandBehavior.CloseConnection)

Try

DATA8.Read()

Manufacturer = DATA8.GetValue(2)

Weight = DATA8.GetValue(3)

Type = DATA8.GetValue(4)

Soket = DATA8.GetValue(5)

For\_what = DATA8.GetValue(6)

Length = Single.Parse(DATA8.GetValue(7).Replace(" мм", "").Trim())

Width = Single.Parse(DATA8.GetValue(8).Replace(" мм", "").Trim())

Cost = DATA8.GetValue(9)

TDP = DATA8.GetValue(10)

choose\_Cooling = New Cooling(id, Manufacturer, Weight, Type,

Soket, For\_what, TDP, Length, Width, Cost)

Me.ResultObject = choose\_Cooling

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex.ToString)

Finally

CO.close()

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

Private Sub Cooling\_Frame\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_manuf.Items.Clear()

Cmb\_type.Items.Clear()

Dim command\_1 As String = "SELECT COUNT(\*)

FROM (SELECT DISTINCT Manufacture FROM COOLING)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT Type FROM COOLING);"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To 1

DATA1.Read()

all(i) = DATA1.GetValue(0)

Next

count1 = all(0)

count2 = all(1)

ReDim manu(0 To count1 - 1), type\_cooling(0 To count2 - 1)

Dim command\_2 As String = "SELECT DISTINCT Manufacture FROM COOLING"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_manuf.Items.AddRange(manu)

Cmb\_manuf.Items.Add("Все")

Cmb\_manuf.SelectedItem = "Все"

Dim command\_3 As String = "SELECT DISTINCT Type FROM COOLING"

Dim COMMAND3 As New System.Data.OleDb.OleDbCommand(command\_3, CO)

Dim DATA3 As System.Data.OleDb.OleDbDataReader = COMMAND3.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count2 - 1

DATA3.Read()

type\_cooling(i) = DATA3.GetValue(0)

Next

Cmb\_type.Items.AddRange(type\_cooling)

Cmb\_type.Items.Add("Все")

Cmb\_type.SelectedItem = "Все"

CO.Close

End Sub

End Class

Imports System.Reflection

Public Class CPU\_Frame

Dim Id, Soket, Sistem\_cooling\_in, Cores, Process\_cores, Max\_kol\_potok, KESH, Baze\_speed, Max\_speed, Type\_RAM, Max\_speed\_RAM, Max\_RAM, TDP, In\_GPU, Control\_PCI, Kol\_lines, Cost As String

Private \_resultObject As CPU

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Public Property ResultObject() As CPU ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As CPU)

\_resultObject = value

End Set

End Property

Private Sub butt\_save\_core\_Click(sender As Object, e As EventArgs) Handles butt\_save\_core.Click

Dim choose\_CPU As CPU

Try

Dim Id\_2 As Integer = Int32.Parse(Id.Substring(2))

Dim core\_2 As Integer = Int32.Parse(Cores)

Dim Process\_cores\_2 As Integer = Int32.Parse(Process\_cores)

Dim Max\_kol\_potok\_2 As Integer = Int32.Parse(Max\_kol\_potok)

Dim Max\_speed\_RAM\_2 As Integer = Int32.Parse(Max\_speed\_RAM)

Dim Max\_RAM\_2 As Integer = Int32.Parse(Max\_RAM)

Dim TDP\_2 As Integer = Int32.Parse(TDP)

Dim Cost\_2 As Decimal = Decimal.Parse(Cost)

Dim KESH\_2 As Double = Double.Parse(KESH)

Dim Baze\_speed\_2 As Double = Double.Parse(Baze\_speed)

Dim Max\_speed\_2 As Double = Double.Parse(Max\_speed)

Dim result As Boolean = If(Sistem\_cooling\_in = "Нет", False, True)

choose\_CPU = New CPU(Id\_2, Cmb\_Manufacturer.Text, Cmb\_model.Text, Soket,

result, core\_2, Process\_cores\_2, Max\_kol\_potok\_2,

KESH\_2, Baze\_speed\_2, Max\_speed\_2, Type\_RAM, Max\_speed\_RAM\_2,

Max\_RAM\_2, TDP\_2, In\_GPU, Control\_PCI, Kol\_lines, Cost\_2)

\_resultObject = New CPU()

ResultObject = choose\_CPU

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex, "Ошибка! ")

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

Private Sub CPU\_Frame\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_model.Items.Clear()

Cmb\_Manufacturer.Items.Clear()

Dim count1 As Integer

Dim manu() As String

Dim command\_1 As String = "SELECT COUNT(\*) FROM (SELECT DISTINCT Manufacturer FROM CPU)"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

DATA1.Read()

count1 = DATA1.GetValue(0)

ReDim manu(0 To count1 - 1)

Dim command\_2 As String = "SELECT DISTINCT Manufacturer FROM CPU"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_Manufacturer.Items.AddRange(manu)

CO.Close

End Sub

Private Sub Cmb\_Manufacturer\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_Manufacturer.SelectedIndexChanged

CO.Open

Cmb\_model.Items.Clear()

Dim count1 As Integer

Dim ai As String = Cmb\_Manufacturer.Items.Item(Cmb\_Manufacturer.SelectedIndex)

Dim manu() As String

Dim command\_1 As String = $"SELECT COUNT(\*) FROM (SELECT DISTINCT Model FROM CPU WHERE Manufacturer = '{ai}')"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

DATA1.Read()

count1 = DATA1.GetValue(0)

ReDim manu(0 To count1 - 1)

Dim command\_2 As String = $"SELECT DISTINCT Model FROM CPU WHERE Manufacturer = '{ai}'"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_model.Items.AddRange(manu)

CO.Close

End Sub

Private Sub Cmb\_model\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_model.SelectedIndexChanged

CO.Open

Dim ai As String = Cmb\_model.Items.Item(Cmb\_model.SelectedIndex)

Dim command\_1 As String = $"SELECT Soket,Sistem\_cooling\_in,Cores,Process\_cores,Max\_kol\_potok,KESH,Baze\_speed,Max\_speed,Type\_RAM,Max\_speed\_RAM,Max\_RAM,TDP,In\_GPU,Control\_PCI,Kol\_lines,Cost,Id FROM CPU WHERE Model = '{ai}'"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

DATA1.Read()

Soket = DATA1.GetValue(0)

Sistem\_cooling\_in = DATA1.GetValue(1)

Cores = DATA1.GetValue(2)

Process\_cores = DATA1.GetValue(3)

Max\_kol\_potok = DATA1.GetValue(4)

KESH = DATA1.GetValue(5)

Baze\_speed = DATA1.GetValue(6)

Max\_speed = DATA1.GetValue(7)

Type\_RAM = DATA1.GetValue(8)

Max\_speed\_RAM = DATA1.GetValue(9)

Max\_RAM = DATA1.GetValue(10)

TDP = DATA1.GetValue(11)

In\_GPU = DATA1.GetValue(12)

Control\_PCI = DATA1.GetValue(13)

Kol\_lines = DATA1.GetValue(14)

Cost = DATA1.GetValue(15)

Id = DATA1.GetValue(16)

lbl\_info.Text = "Информация:" + vbCrLf + "Сокет - " + Soket + vbCrLf + "Система охалждения в комплекте: " + Sistem\_cooling\_in + vbCrLf + "Общее количество ядер: " + Cores + vbCrLf + "Количество производительных ядер: " + Process\_cores + vbCrLf + "Максимальное число потоков: " + Max\_kol\_potok + vbCrLf + "Объем кэша: " + KESH + vbCrLf + "Базовая частота процессора: " + Baze\_speed + vbCrLf + "Максимальная частота в турбо: " + Max\_speed

lbl\_info2.Text = "Тип памяти: " + Type\_RAM + vbCrLf + "Частота оперативной памяти: " + Max\_speed\_RAM + vbCrLf + "Максимально поддерживаемый объем памяти: " + Max\_RAM + vbCrLf + "Тепловыделение: " + TDP + vbCrLf + "Интегрированное графическое ядро: " + In\_GPU + vbCrLf + "Встроенный контроллер PCI Express: " + Control\_PCI + vbCrLf + "Число линий PCI Express: " + Kol\_lines + vbCrLf + "Цена: " + Cost

CO.Close

End Sub

End Class

Public Class Disk\_Frame

Private \_resultObject As Disk

Dim count1, count2, all(0 To 1) As Integer

Dim manu(), V\_graf() As String

Dim intriger As Integer = 0

Public Property ResultObject() As Disk ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As Disk)

\_resultObject = value

End Set

End Property

Public Sub Obnov()

Dim whereClauses As New List(Of String)()

Dim command\_7 As String

If intriger < 2 Then

intriger += 1

Return

Else

CO.Open()

' Формируем условия WHERE

If Cmb\_Manuf.Text() <> "Все" Then

whereClauses.Add($"Manufacturer = '{Cmb\_Manuf.SelectedItem}'")

End If

If Cmb\_V\_disk.Text() <> "Все" Then

whereClauses.Add($"V\_of\_disk = {Cmb\_V\_disk.SelectedItem}")

End If

' Собираем итоговый SQL

command\_7 = "SELECT \* FROM DISK"

If whereClauses.Count > 0 Then

command\_7 = command\_7 & " WHERE " & String.Join(" AND ", whereClauses)

End If

Dim COMMAND7 As New System.Data.OleDb.OleDbCommand(command\_7, CO)

Dim DATA7 As System.Data.OleDb.OleDbDataReader = COMMAND7.ExecuteReader(CommandBehavior.CloseConnection)

Dim dataTable As New DataTable()

dataTable.Load(DATA7)

DataGV.DataSource = dataTable

CO.Close

End If

End Sub

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Private Sub Cmb\_V\_disk\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_V\_disk.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_Manuf\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_Manuf.SelectedIndexChanged

Obnov()

End Sub

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

CO.open()

Dim Manufacturer As String, Weight As Double, Type As String, V\_of\_disk As Integer, Interface\_q As String, Max\_speed\_read As Integer, Max\_speed\_write As Integer, Max\_V\_write As Integer, Length As Double, Cost As Decimal

Dim choose\_DISK As Disk

Dim id As Integer = Nud\_id.Value

Dim command\_8 = $"SELECT \* FROM DISK WHERE Id = '7\_{id}'"

Dim COMMAND8 As New System.Data.OleDb.OleDbCommand(command\_8, CO)

Dim DATA8 As System.Data.OleDb.OleDbDataReader = COMMAND8.ExecuteReader(CommandBehavior.CloseConnection)

Try

DATA8.Read()

Manufacturer = DATA8.GetValue(2)

Weight = DATA8.GetValue(3)

Type = DATA8.GetValue(4)

V\_of\_disk = DATA8.GetValue(5)

Interface\_q = DATA8.GetValue(6)

Max\_speed\_read = DATA8.GetValue(7)

Max\_speed\_write = DATA8.GetValue(8)

Max\_V\_write = Integer.Parse(DATA8.GetValue(9).ToString().Replace(" TBW", "").Trim())

Length = DATA8.GetValue(10)

Cost = DATA8.GetValue(11)

choose\_DISK = New Disk(id, Manufacturer, Weight, Type, V\_of\_disk, Interface\_q, Max\_speed\_read, Max\_speed\_write, Max\_V\_write, Length, Cost)

Me.ResultObject = choose\_DISK

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex.ToString)

Finally

CO.close()

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

Private Sub Disk\_Frame\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_Manuf.Items.Clear()

Cmb\_V\_disk.Items.Clear()

Dim command\_1 As String = "SELECT COUNT(\*)

FROM (SELECT DISTINCT Manufacturer FROM DISK)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT V\_of\_disk FROM DISK);"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To 1

DATA1.Read()

all(i) = DATA1.GetValue(0)

Next

count1 = all(0)

count2 = all(1)

ReDim manu(0 To count1 - 1), V\_graf(0 To count2 - 1)

Dim command\_2 As String = "SELECT DISTINCT Manufacturer FROM DISK"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_Manuf.Items.AddRange(manu)

Cmb\_Manuf.Items.Add("Все")

Cmb\_Manuf.SelectedItem = "Все"

Dim command\_3 As String = "SELECT DISTINCT V\_of\_disk FROM DISK"

Dim COMMAND3 As New System.Data.OleDb.OleDbCommand(command\_3, CO)

Dim DATA3 As System.Data.OleDb.OleDbDataReader = COMMAND3.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count2 - 1

DATA3.Read()

V\_graf(i) = DATA3.GetValue(0)

Next

Cmb\_V\_disk.Items.AddRange(V\_graf)

Cmb\_V\_disk.Items.Add("Все")

Cmb\_V\_disk.SelectedItem = "Все"

CO.Close

End Sub

End Class

Imports System.Diagnostics.Metrics

Imports System.Windows.Forms.VisualStyles.VisualStyleElement

Imports FxResources.System

Public Class FRAME\_FRAME

Private \_resultObject As Frame

Dim count1, count2, all(0 To 1) As Integer

Dim Color\_in(), Water() As String

Dim intriger As Integer = 0

Public Property ResultObject() As Frame ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As Frame)

\_resultObject = value

End Set

End Property

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Public Sub Obnov()

Dim whereClauses As New List(Of String)()

Dim command\_7 As String

If intriger < 2 Then

intriger += 1

Return

Else

CO.Open()

' Формируем условия WHERE

If Cmb\_Color.Text() <> "Все" Then

whereClauses.Add($"Color = '{Cmb\_Color.SelectedItem}'")

End If

If Cmb\_water.Text() <> "Все" Then

whereClauses.Add($"Water\_cooling = '{Cmb\_water.SelectedItem}'")

End If

If Sborka.computer.Motherboard IsNot Nothing Then

Dim form As String = Sborka.computer.Motherboard.Form\_factor

whereClauses.Add($"Form\_factor = '{Form}'")

End If

' Собираем итоговый SQL

command\_7 = "SELECT \* FROM FRAME"

If whereClauses.Count > 0 Then

command\_7 = command\_7 & " WHERE " & String.Join(" AND ", whereClauses)

End If

Dim COMMAND7 As New System.Data.OleDb.OleDbCommand(command\_7, CO)

Dim DATA7 As System.Data.OleDb.OleDbDataReader = COMMAND7.ExecuteReader(CommandBehavior.CloseConnection)

Dim dataTable As New DataTable()

dataTable.Load(DATA7)

DataGV.DataSource = dataTable

CO.Close

End If

End Sub

Private Sub Cmb\_water\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_water.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_Color\_in\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_Color.SelectedIndexChanged

Obnov()

End Sub

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

CO.open()

Dim Manufacturer As String, Hight As Double, Width As Double, Length As Double, Weight As Double, Color\_in As String, Material As String, Form\_factor As String, Max\_lenght\_of\_GPU As Integer, Water\_cooling As Boolean, Cost As Decimal

Dim choose\_Frame As Frame

Dim id As Integer = Nud\_id.Value

Dim command\_8 = $"SELECT \* FROM FRAME WHERE Id = '1\_{id}'"

Dim COMMAND8 As New System.Data.OleDb.OleDbCommand(command\_8, CO)

Dim DATA8 As System.Data.OleDb.OleDbDataReader = COMMAND8.ExecuteReader(CommandBehavior.CloseConnection)

Try

DATA8.Read()

Manufacturer = DATA8.GetValue(2)

Hight = DATA8.GetValue(3)

Width = DATA8.GetValue(4)

Length = DATA8.GetValue(5)

Weight = DATA8.GetValue(6)

Color\_in = DATA8.GetValue(7)

Material = DATA8.GetValue(8)

Form\_factor = DATA8.GetValue(9)

Max\_lenght\_of\_GPU = DATA8.GetValue(10)

Water\_cooling = If(DATA8.GetValue(11) = "Да", True, False)

Cost = DATA8.GetValue(12)

choose\_Frame = New Frame(id, Manufacturer, Hight, Width, Length, Weight, Color\_in,

Material, Form\_factor, Max\_lenght\_of\_GPU, Water\_cooling, Cost)

Me.ResultObject = choose\_Frame

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex.ToString)

Finally

CO.close()

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

Private Sub FRAME\_FRAME\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_Color.Items.Clear()

Cmb\_water.Items.Clear()

Dim command\_1 As String = "SELECT COUNT(\*)

FROM (SELECT DISTINCT Color FROM FRAME)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT Water\_cooling FROM FRAME);"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To 1

DATA1.Read()

all(i) = DATA1.GetValue(0)

Next

count1 = all(0)

count2 = all(1)

ReDim Color\_in(0 To count1 - 1), Water(0 To count2 - 1)

Dim command\_2 As String = "SELECT DISTINCT Color FROM FRAME"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

Color\_in(i) = DATA2.GetValue(0)

Next

Cmb\_Color.Items.AddRange(Color\_in)

Cmb\_Color.Items.Add("Все")

Cmb\_Color.SelectedItem = "Все"

Dim command\_3 As String = "SELECT DISTINCT Water\_cooling FROM FRAME"

Dim COMMAND3 As New System.Data.OleDb.OleDbCommand(command\_3, CO)

Dim DATA3 As System.Data.OleDb.OleDbDataReader = COMMAND3.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count2 - 1

DATA3.Read()

Water(i) = DATA3.GetValue(0)

Next

Cmb\_water.Items.AddRange(Water)

Cmb\_water.Items.Add("Все")

Cmb\_water.SelectedItem = "Все"

CO.Close

End Sub

End Class

Public Class GPU\_Frame

Private \_resultObject As GPU

Dim count1, count2, all(0 To 1) As Integer

Dim manu(), V\_graf() As String

Dim intriger As Integer = 0

Public Property ResultObject() As GPU ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As GPU)

\_resultObject = value

End Set

End Property

Public Sub Obnov()

Dim whereClauses As New List(Of String)()

Dim interfaceClauses As New List(Of String)()

Dim command\_7 As String

If intriger < 2 Then

intriger += 1

Return

Else

CO.Open()

' Формируем условия WHERE

If Cmb\_Manuf.Text() <> "Все" Then

whereClauses.Add($"Manufacturer = '{Cmb\_Manuf.SelectedItem}'")

End If

If Cmb\_V\_graf.Text() <> "Все" Then

whereClauses.Add($"V\_GPU = {Cmb\_V\_graf.SelectedItem}")

End If

If Sborka.computer.CPU IsNot Nothing Then

Dim Interface\_i() As String = Sborka.computer.CPU.Control\_PCI.Split(",", StringSplitOptions.None)

For Each i As String In Interface\_i

interfaceClauses.Add($"Interface = '{i}'")

Next

End If

If interfaceClauses.Count > 1 Then

Dim interfaceCondition As String = "(" & String.Join(" OR ", interfaceClauses) & ")"

whereClauses.Add(interfaceCondition)

End If

' Собираем итоговый SQL

command\_7 = "SELECT \* FROM GPU"

If whereClauses.Count > 0 Then

command\_7 = command\_7 & " WHERE " & String.Join(" AND ", whereClauses)

End If

Dim COMMAND7 As New System.Data.OleDb.OleDbCommand(command\_7, CO)

Dim DATA7 As System.Data.OleDb.OleDbDataReader = COMMAND7.ExecuteReader(CommandBehavior.CloseConnection)

Dim dataTable As New DataTable()

dataTable.Load(DATA7)

DataGV.DataSource = dataTable

CO.Close

End If

End Sub

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Private Sub GPU\_Frame\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_Manuf.Items.Clear()

Cmb\_V\_graf.Items.Clear()

Dim command\_1 As String = "SELECT COUNT(\*)

FROM (SELECT DISTINCT Manufacturer FROM GPU)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT V\_GPU FROM GPU);"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To 1

DATA1.Read()

all(i) = DATA1.GetValue(0)

Next

count1 = all(0)

count2 = all(1)

ReDim manu(0 To count1 - 1), V\_graf(0 To count2 - 1)

Dim command\_2 As String = "SELECT DISTINCT Manufacturer FROM GPU"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_Manuf.Items.AddRange(manu)

Cmb\_Manuf.Items.Add("Все")

Cmb\_Manuf.SelectedItem = "Все"

Dim command\_3 As String = "SELECT DISTINCT V\_GPU FROM GPU"

Dim COMMAND3 As New System.Data.OleDb.OleDbCommand(command\_3, CO)

Dim DATA3 As System.Data.OleDb.OleDbDataReader = COMMAND3.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count2 - 1

DATA3.Read()

V\_graf(i) = DATA3.GetValue(0)

Next

Cmb\_V\_graf.Items.AddRange(V\_graf)

Cmb\_V\_graf.Items.Add("Все")

Cmb\_V\_graf.SelectedItem = "Все"

CO.Close

End Sub

Private Sub Cmb\_Manuf\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_Manuf.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_V\_graf\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_V\_graf.SelectedIndexChanged

Obnov()

End Sub

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

CO.open()

Dim Manufacturer As String, Weight As Double, Color As String, Model As String, Graphic\_unit As String, Frequency As Integer, Turbofrequency As Integer, DLSS As Boolean, V\_GPU As Integer, Type\_of\_memory As String, Type\_of\_ports As String, Kol\_ports As Integer, Interface\_q As String, Dop\_power As String, Type\_of\_cooling As String, LED As Boolean, Cost As Decimal

Dim choose\_GPU As GPU

Dim id As Integer = Nud\_id.Value

Dim command\_8 = $"SELECT \* FROM GPU WHERE Id = '6\_{id}'"

Dim COMMAND8 As New System.Data.OleDb.OleDbCommand(command\_8, CO)

Dim DATA8 As System.Data.OleDb.OleDbDataReader = COMMAND8.ExecuteReader(CommandBehavior.CloseConnection)

Try

DATA8.Read()

Manufacturer = DATA8.GetValue(2)

Weight = DATA8.GetValue(3)

Color = DATA8.GetValue(4)

Model = DATA8.GetValue(5)

Graphic\_unit = DATA8.GetValue(6)

Frequency = DATA8.GetValue(7)

Turbofrequency = DATA8.GetValue(8)

DLSS = If(DATA8.GetValue(9) = "Да", True, False)

V\_GPU = DATA8.GetValue(10)

Type\_of\_memory = DATA8.GetValue(11)

Type\_of\_ports = DATA8.GetValue(12)

Kol\_ports = DATA8.GetValue(13)

Interface\_q = DATA8.GetValue(14)

Dop\_power = DATA8.GetValue(15)

Type\_of\_cooling = DATA8.GetValue(16)

LED = If(DATA8.GetValue(17) = "Да", True, False)

Cost = DATA8.GetValue(18)

choose\_GPU = New GPU(id, Manufacturer, Weight, Color, Model, Graphic\_unit,

Frequency, Turbofrequency, DLSS, V\_GPU, Type\_of\_memory,

Type\_of\_ports, Kol\_ports, Interface\_q, Dop\_power, Type\_of\_cooling, LED, Cost)

Me.ResultObject = choose\_GPU

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex.ToString)

Finally

CO.close()

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

End Class

Imports System.ComponentModel.Design

Imports Windows.Win32.System

Public Class Motherboard\_Frame

Private \_resultObject As Motherboard

Dim count1, count2, count3, count4, all(0 To 3), kol\_m2(), max\_speed() As Integer

Dim manu(), type\_ram() As String

Dim intriger As Integer = 0

Public Property ResultObject() As Motherboard ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As Motherboard)

\_resultObject = value

End Set

End Property

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

CO.open()

Dim Manufacturer As String, Weight As Double, Color As String, Length As Double, Width As Double, Soket As String, Type\_RAM As String, Max\_V\_RAM As Integer, Max\_speed\_RAM As Integer, Kol\_slots\_RAM As Integer, PCI\_version As String, Kol\_M2 As Integer, USB As Integer, Video\_port As String, Form\_factor As String, Esernet\_adapter As String, Esernet\_speed As String, Cost As Decimal

Dim choose\_Motherboard As Motherboard

Dim id As Integer = Nud\_id.Value

Dim command\_8 = $"SELECT \* FROM MATHERBOARD WHERE Id = '2\_{id}'"

Dim COMMAND8 As New System.Data.OleDb.OleDbCommand(command\_8, CO)

Dim DATA8 As System.Data.OleDb.OleDbDataReader = COMMAND8.ExecuteReader(CommandBehavior.CloseConnection)

Try

DATA8.Read()

Manufacturer = DATA8.GetValue(2)

Weight = DATA8.GetValue(3)

Color = DATA8.GetValue(4)

Length = DATA8.GetValue(5)

Width = DATA8.GetValue(6)

Soket = DATA8.GetValue(7)

If Soket <> Sborka.computer.CPU.Soket Then

Throw New Exception("Сокет материнской платы не соответствует выбранному процессору! Поменяйте выбор из предложенных.")

End If

Type\_RAM = DATA8.GetValue(8)

Max\_V\_RAM = DATA8.GetValue(9)

Max\_speed\_RAM = DATA8.GetValue(10)

Kol\_slots\_RAM = DATA8.GetValue(11)

PCI\_version = DATA8.GetValue(12).ToString + ".0"

Kol\_M2 = DATA8.GetValue(13)

USB = DATA8.GetValue(14)

Video\_port = DATA8.GetValue(15)

Form\_factor = DATA8.GetValue(16)

Esernet\_adapter = DATA8.GetValue(17)

Esernet\_speed = DATA8.GetValue(18)

Cost = DATA8.GetValue(19)

choose\_Motherboard = New Motherboard(id, Manufacturer, Weight, Color, Length, Width, Soket,

Type\_RAM, Max\_V\_RAM, Max\_speed\_RAM, Kol\_slots\_RAM, PCI\_version,

Kol\_M2, USB, Video\_port, Form\_factor, Esernet\_adapter, Esernet\_speed, Cost)

Me.ResultObject = choose\_Motherboard

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex.ToString)

Finally

CO.close()

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

Public Sub Obnov()

Dim whereClauses As New List(Of String)()

Dim command\_7 As String

Dim soket As String = Sborka.computer.CPU.Soket

If intriger < 4 Then

intriger += 1

Return

Else

CO.Open()

' Формируем условия WHERE

If Cmb\_manuf.Text() <> "Все" Then

whereClauses.Add($"Manufacturer = '{Cmb\_manuf.SelectedItem}'")

End If

If Cmb\_Type\_RAM.Text() <> "Все" Then

whereClauses.Add($"Type\_RAM = '{Cmb\_Type\_RAM.SelectedItem}'")

End If

If Cmb\_Max\_speed\_RAM.Text() <> "Все" Then

whereClauses.Add($"Max\_speed\_RAM <= {Cmb\_Max\_speed\_RAM.SelectedItem}")

End If

If Cmb\_kol\_m2.Text <> "Все" Then

whereClauses.Add($"Kol\_M2 = {Cmb\_kol\_m2.SelectedItem}")

End If

' Собираем итоговый SQL

command\_7 = "SELECT \* FROM MATHERBOARD" & " WHERE " & $"Soket = '{soket}'"

If whereClauses.Count > 0 Then

command\_7 = command\_7 & " AND " & String.Join(" AND ", whereClauses)

End If

Dim COMMAND7 As New System.Data.OleDb.OleDbCommand(command\_7, CO)

Dim DATA7 As System.Data.OleDb.OleDbDataReader = COMMAND7.ExecuteReader(CommandBehavior.CloseConnection)

Dim dataTable As New DataTable()

dataTable.Load(DATA7)

DataGV.DataSource = dataTable

CO.Close

End If

End Sub

Private Sub Motherboard\_Frame\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_manuf.Items.Clear()

Cmb\_Type\_RAM.Items.Clear()

Cmb\_kol\_m2.Items.Clear()

Cmb\_Max\_speed\_RAM.Items.Clear()

Dim command\_1 As String = "SELECT COUNT(\*)

FROM (SELECT DISTINCT Manufacturer FROM MATHERBOARD)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT Max\_speed\_RAM FROM MATHERBOARD)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT Kol\_M2 FROM MATHERBOARD)

UNION ALL

Select COUNT(\*) FROM (Select DISTINCT Type\_RAM FROM MATHERBOARD);"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To 3

DATA1.Read()

all(i) = DATA1.GetValue(0)

Next

count1 = all(0)

count4 = all(1)

count3 = all(2)

count2 = all(3)

ReDim manu(0 To count1 - 1), type\_ram(0 To count2 - 1), kol\_m2(0 To count3 - 1), max\_speed(0 To count4 - 1)

Dim command\_2 As String = "Select DISTINCT Manufacturer FROM MATHERBOARD"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_manuf.Items.AddRange(manu)

Cmb\_manuf.Items.Add("Все")

Cmb\_manuf.SelectedItem = "Все"

Dim command\_3 As String = "Select DISTINCT Type\_RAM FROM MATHERBOARD"

Dim COMMAND3 As New System.Data.OleDb.OleDbCommand(command\_3, CO)

Dim DATA3 As System.Data.OleDb.OleDbDataReader = COMMAND3.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count2 - 1

DATA3.Read()

type\_ram(i) = DATA3.GetValue(0)

Next

Cmb\_Type\_RAM.Items.AddRange(type\_ram)

Cmb\_Type\_RAM.Items.Add("Все")

Cmb\_Type\_RAM.SelectedItem = "Все"

Dim command\_4 As String = "Select DISTINCT Kol\_M2 FROM MATHERBOARD"

Dim COMMAND4 As New System.Data.OleDb.OleDbCommand(command\_4, CO)

Dim DATA4 As System.Data.OleDb.OleDbDataReader = COMMAND4.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count3 - 1

DATA4.Read()

kol\_m2(i) = DATA4.GetValue(0)

Next

Cmb\_kol\_m2.Items.AddRange(String.Join(", ", kol\_m2).Split(New String() {", "}, StringSplitOptions.None))

Cmb\_kol\_m2.Items.Add("Все")

Cmb\_kol\_m2.SelectedItem = "Все"

Dim command\_5 As String = "Select DISTINCT Max\_speed\_RAM FROM MATHERBOARD"

Dim COMMAND5 As New System.Data.OleDb.OleDbCommand(command\_5, CO)

Dim DATA5 As System.Data.OleDb.OleDbDataReader = COMMAND5.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count4 - 1

DATA5.Read()

max\_speed(i) = DATA5.GetValue(0)

Next

Cmb\_Max\_speed\_RAM.Items.AddRange(String.Join(", ", max\_speed).Split(New String() {", "}, StringSplitOptions.None))

Cmb\_Max\_speed\_RAM.Items.Add("Все")

Cmb\_Max\_speed\_RAM.SelectedItem = "Все"

CO.Close

End Sub

Private Sub Cmb\_manuf\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_manuf.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_Type\_RAM\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_Type\_RAM.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_Max\_speed\_RAM\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_Max\_speed\_RAM.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_kol\_m2\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_kol\_m2.SelectedIndexChanged

Obnov()

End Sub

End Class

Public Class Powerplant\_Frame

Private \_resultObject As Powerplant

Dim count1, count2, all(0 To 1) As Integer

Dim manu(), type\_cooling() As String

Dim intriger As Integer = 0

Public Property ResultObject() As Powerplant ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As Powerplant)

\_resultObject = value

End Set

End Property

Public Function sertificate(sert As String)

Select Case sert

Case "80+ Bronze"

Return 0.81

Case "80+ Gold"

Return 0.88

Case "80+ Platinum"

Return 0.9

Case "80+ Titanium"

Return 0.94

Case "80+ White"

Return 0.85

Case "Нет"

Return 0.8

Case Else

Return 0.8

End Select

End Function

Public Sub Obnov()

Dim whereClauses As New List(Of String)()

Dim command\_7 As String

If intriger < 2 Then

intriger += 1

Return

Else

CO.Open()

' Формируем условия WHERE

If Cmb\_manuf.Text() <> "Все" Then

whereClauses.Add($"Manufacturer = '{Cmb\_manuf.SelectedItem}'")

End If

If Cmb\_type.Text() <> "Все" Then

whereClauses.Add($"Sertif = '{Cmb\_type.SelectedItem}'")

End If

If Sborka.computer.CPU IsNot Nothing And Sborka.computer.GPU IsNot Nothing Then

whereClauses.Add($"Сapacity >= {Fix((Sborka.computer.CPU.TDP + 400) / sertificate(Cmb\_type.Text))}")

ElseIf Sborka.computer.CPU IsNot Nothing And Sborka.computer.GPU Is Nothing Then

whereClauses.Add($"Сapacity >= {Fix((Sborka.computer.CPU.TDP + 100) / sertificate(Cmb\_type.Text))}")

End If

' Собираем итоговый SQL

command\_7 = "SELECT \* FROM POWER\_BLOK"

If whereClauses.Count > 0 Then

command\_7 = command\_7 & " WHERE " & String.Join(" AND ", whereClauses)

End If

Dim COMMAND7 As New System.Data.OleDb.OleDbCommand(command\_7, CO)

Dim DATA7 As System.Data.OleDb.OleDbDataReader = COMMAND7.ExecuteReader(CommandBehavior.CloseConnection)

Dim dataTable As New DataTable()

dataTable.Load(DATA7)

DataGV\_1.DataSource = dataTable

CO.Close

End If

End Sub

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Private Sub Cmb\_manuf\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_manuf.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_type\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_type.SelectedIndexChanged

Obnov()

End Sub

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

CO.open()

Dim Manufacturer As String, Weight As Double, Color\_1 As String, Сapacity As Integer, Form\_factor As String, Power\_CPU As String, Power\_GPU As String, l\_15\_pin\_SATA As String, l\_4\_pin\_Molex As String, Cooling\_system As String, Sertif As String, Cost As Decimal

Dim choose\_Powerplant As Powerplant

Dim id As Integer = Nud\_id.Value

Dim command\_8 = $"SELECT \* FROM POWER\_BLOK WHERE Id = '4\_{id}'"

Dim COMMAND8 As New System.Data.OleDb.OleDbCommand(command\_8, CO)

Dim DATA8 As System.Data.OleDb.OleDbDataReader = COMMAND8.ExecuteReader(CommandBehavior.CloseConnection)

Try

DATA8.Read()

Manufacturer = DATA8.GetValue(2)

Weight = DATA8.GetValue(3)

Color\_1 = DATA8.GetValue(4)

Сapacity = DATA8.GetValue(5)

Form\_factor = DATA8.GetValue(6)

Power\_CPU = DATA8.GetValue(7)

Power\_GPU = DATA8.GetValue(8)

l\_15\_pin\_SATA = DATA8.GetValue(9)

l\_4\_pin\_Molex = DATA8.GetValue(10)

Cooling\_system = DATA8.GetValue(11)

Sertif = DATA8.GetValue(12)

Cost = DATA8.GetValue(13)

choose\_Powerplant = New Powerplant(id, Manufacturer, Weight, Color\_1, Сapacity, Form\_factor,

Power\_CPU, Power\_GPU, l\_15\_pin\_SATA, l\_4\_pin\_Molex, Cooling\_system, Sertif, Cost)

Me.ResultObject = choose\_Powerplant

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex.ToString)

Finally

CO.close()

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

Private Sub Powerplant\_Frame\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_manuf.Items.Clear()

Cmb\_type.Items.Clear()

Dim command\_1 As String = "SELECT COUNT(\*)

FROM (SELECT DISTINCT Manufacturer FROM POWER\_BLOK)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT Sertif FROM POWER\_BLOK);"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To 1

DATA1.Read()

all(i) = DATA1.GetValue(0)

Next

count1 = all(0)

count2 = all(1)

ReDim manu(0 To count1 - 1), type\_cooling(0 To count2 - 1)

Dim command\_2 As String = "SELECT DISTINCT Manufacturer FROM POWER\_BLOK"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_manuf.Items.AddRange(manu)

Cmb\_manuf.Items.Add("Все")

Cmb\_manuf.SelectedItem = "Все"

Dim command\_3 As String = "SELECT DISTINCT Sertif FROM POWER\_BLOK"

Dim COMMAND3 As New System.Data.OleDb.OleDbCommand(command\_3, CO)

Dim DATA3 As System.Data.OleDb.OleDbDataReader = COMMAND3.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count2 - 1

DATA3.Read()

type\_cooling(i) = DATA3.GetValue(0)

Next

Cmb\_type.Items.AddRange(type\_cooling)

Cmb\_type.Items.Add("Все")

Cmb\_type.SelectedItem = "Все"

CO.Close

End Sub

End Class

Public Class RAM\_Frame

Private \_resultObject As RAM

Dim count1, count2, all(0 To 1) As Integer

Dim manu(), V\_graf() As String

Dim intriger As Integer = 0

Public Property ResultObject() As RAM ' Public свойство для доступа к объекту

Get

Return \_resultObject

End Get

Set(value As RAM)

\_resultObject = value

End Set

End Property

Public Sub Obnov()

Dim whereClauses As New List(Of String)()

Dim command\_7 As String

If intriger < 2 Then

intriger += 1

Return

Else

CO.Open()

' Формируем условия WHERE

If Cmb\_One\_RAM.Text() <> "Все" Then

whereClauses.Add($"One\_RAM = {Cmb\_One\_RAM.SelectedItem}")

End If

If Cmb\_Kol\_RAM.Text() <> "Все" Then

whereClauses.Add($"Kol\_RAM = {Cmb\_Kol\_RAM.SelectedItem}")

End If

If Sborka.computer.Motherboard IsNot Nothing Then

whereClauses.Add($"Speed\_RAM <= {Sborka.computer.Motherboard.Max\_speed\_RAM}")

whereClauses.Add($"Type\_RAM = '{Sborka.computer.Motherboard.Type\_RAM}'")

End If

' Собираем итоговый SQL

command\_7 = "SELECT \* FROM RAM"

If whereClauses.Count > 0 Then

command\_7 = command\_7 & " WHERE " & String.Join(" AND ", whereClauses)

End If

Dim COMMAND7 As New System.Data.OleDb.OleDbCommand(command\_7, CO)

Dim DATA7 As System.Data.OleDb.OleDbDataReader = COMMAND7.ExecuteReader(CommandBehavior.CloseConnection)

Dim dataTable As New DataTable()

dataTable.Load(DATA7)

DataGV.DataSource = dataTable

CO.Close

End If

End Sub

Dim CO = New System.Data.OleDb.OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0;Data Source=|DataDirectory|\Resources\Kurs.accdb;Persist Security Info=False;")

Private Sub Cmb\_One\_RAM\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_One\_RAM.SelectedIndexChanged

Obnov()

End Sub

Private Sub Cmb\_Kol\_RAM\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles Cmb\_Kol\_RAM.SelectedIndexChanged

Obnov()

End Sub

Private Sub butt\_save\_Click(sender As Object, e As EventArgs) Handles butt\_save.Click

CO.open()

Dim Manufacturer As String, Color As String, Type\_RAM As String, Sum\_RAM As Integer, One\_RAM As Integer, Kol\_RAM As Integer, Speed\_RAM As Integer, Cooler\_RAM As Boolean, Cost As Decimal

Dim choose\_RAM As RAM

Dim id As Integer = Nud\_id.Value

Dim command\_8 = $"SELECT \* FROM RAM WHERE Id = '3\_{id}'"

Dim COMMAND8 As New System.Data.OleDb.OleDbCommand(command\_8, CO)

Dim DATA8 As System.Data.OleDb.OleDbDataReader = COMMAND8.ExecuteReader(CommandBehavior.CloseConnection)

Try

DATA8.Read()

Manufacturer = DATA8.GetValue(2)

Color = DATA8.GetValue(3)

Type\_RAM = DATA8.GetValue(4)

Sum\_RAM = DATA8.GetValue(5)

One\_RAM = DATA8.GetValue(6)

Kol\_RAM = DATA8.GetValue(7)

Speed\_RAM = DATA8.GetValue(8)

Cooler\_RAM = If(DATA8.GetValue(9) = "Да", True, False)

Cost = DATA8.GetValue(10)

choose\_RAM = New RAM(id, Manufacturer, Color, Type\_RAM, Sum\_RAM, One\_RAM, Kol\_RAM, Speed\_RAM, Cooler\_RAM, Cost)

Me.ResultObject = choose\_RAM

Me.DialogResult = DialogResult.OK

Catch ex As Exception

MessageBox.Show(ex.ToString)

Finally

CO.close()

End Try

End Sub

Private Sub butt\_close\_Click(sender As Object, e As EventArgs) Handles butt\_close.Click

If (CO.State = ConnectionState.Open) Then

CO.Close()

End If

Me.DialogResult = DialogResult.Cancel

End Sub

Private Sub RAM\_Frame\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

CO.Open

Cmb\_One\_RAM.Items.Clear()

Cmb\_Kol\_RAM.Items.Clear()

Dim command\_1 As String = "SELECT COUNT(\*)

FROM (SELECT DISTINCT One\_RAM FROM RAM)

UNION ALL

Select COUNT(\*)

FROM (SELECT DISTINCT Kol\_RAM FROM RAM);"

Dim COMMAND As New System.Data.OleDb.OleDbCommand(command\_1, CO)

Dim DATA1 As System.Data.OleDb.OleDbDataReader = COMMAND.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To 1

DATA1.Read()

all(i) = DATA1.GetValue(0)

Next

count1 = all(0)

count2 = all(1)

ReDim manu(0 To count1 - 1), V\_graf(0 To count2 - 1)

Dim command\_2 As String = "SELECT DISTINCT One\_RAM FROM RAM"

Dim COMMAND2 As New System.Data.OleDb.OleDbCommand(command\_2, CO)

Dim DATA2 As System.Data.OleDb.OleDbDataReader = COMMAND2.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count1 - 1

DATA2.Read()

manu(i) = DATA2.GetValue(0)

Next

Cmb\_One\_RAM.Items.AddRange(manu)

Cmb\_One\_RAM.Items.Add("Все")

Cmb\_One\_RAM.SelectedItem = "Все"

Dim command\_3 As String = "SELECT DISTINCT Kol\_RAM FROM RAM"

Dim COMMAND3 As New System.Data.OleDb.OleDbCommand(command\_3, CO)

Dim DATA3 As System.Data.OleDb.OleDbDataReader = COMMAND3.ExecuteReader(CommandBehavior.CloseConnection)

For i = 0 To count2 - 1

DATA3.Read()

V\_graf(i) = DATA3.GetValue(0)

Next

Cmb\_Kol\_RAM.Items.AddRange(V\_graf)

Cmb\_Kol\_RAM.Items.Add("Все")

Cmb\_Kol\_RAM.SelectedItem = "Все"

CO.Close

End Sub

End Class