Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра компьютерных систем в управлении и проектировании

(КСУП)

РАЗРАБОТКА ПЛАГИНА «ЯЩИК ДЛЯ ДЕТАЛЕЙ»

ДЛЯ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ «AUTODESK INVENTOR 2022»

Пояснительная записка по дисциплине

«Основы разработки САПР» (ОрСАПР)

Выполнил:

Студент гр.588-3

\_\_\_\_\_\_\_Набережнев Н.А.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Руководитель:

к.т.н., доцент

каф. КСУП

\_\_\_\_\_\_\_Калентьев А.А.

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2021 г.

**РЕФЕРАТ**

Лабораторная работа 25 с., 16 рис., 11 источников.

Ключевые слова: AUTODESK INVENTOR, VISUAL STUDIO, C#, INVENTOR ADDIN, ПЛАГИН, САПР.

Целью данной работы является разработка плагина для автоматизации проектирования модели ящика для деталей.

Результатом работы является плагин, осуществляющий построение модели ящика для деталей по заданным пользователям параметрам.

Пояснительная записка выполнена в текстовом редакторе Microsoft Word 2016.

**Содержание**

[1 Введение 4](#_Toc91334109)

[2 Постановка и анализ задачи 5](#_Toc91334110)

[2.1 Описание предмета проектирования 6](#_Toc91334111)

[2.2 Выбор инструментов и средств реализации 7](#_Toc91334112)

[2.3 Назначение плагина 7](#_Toc91334113)

[3 Обзор аналогов 8](#_Toc91334114)

[4 Описание реализации 9](#_Toc91334115)

[4.1 Диаграмма классов 9](#_Toc91334116)

[5 Описание программы для пользователя 14](#_Toc91334117)

[6 Тестирование плагина 17](#_Toc91334118)

[6.1 Функциональное тестирование 17](#_Toc91334119)

[6.2 Модульное тестирование 19](#_Toc91334120)

[6.3 Нагрузочное тестирование 20](#_Toc91334121)

[Заключение 24](#_Toc91334122)

[Список использованных источников 25](#_Toc91334123)

# Введение

Автоматизация моделирования имеет огромное значение для развития науки, техники и производства в современном обществе. В настоящее время автоматизация – основной способ повышения производительности и эффективности труда инженерно-технических работников, занимающихся моделированием сложных устройств. Использование автоматизации в проектировании позволяет создавать все более сложные технические объекты и гибко реагировать на появление новых решений и технологий в той или иной области техники. Она позволяет значительно повысить точность расчетов, выбрать наилучшие варианты для реализации на основе строгого математического анализа всех или большинства вариантов проекта с оценкой технических, технологических и экономических характеристик производства и эксплуатации проектируемого объекта, значительно повысить качество конструкторской документации, существенно сократить сроки проектирования и передачи конструкторской документации в производство, эффективнее использовать технологическое оборудование с программным управлением [1].

Таким образом, целью данной работы является разработка плагина, автоматизирующего построение модели «Ящик для деталей» для системы автоматизированного проектирования Inventor 2022 с помощью интегрированной среды разработки Visual Studio 2019 Сommunity. [2]

Интегрированная среда разработки Visual Studio — это стартовая площадка для написания, отладки и сборки кода, а также последующей публикации приложений. Интегрированная среда разработки (IDE) представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

# Постановка и анализ задачи

В рамках лабораторных работ в соответствии с технически заданием требовалось разработать плагин, который на основе входных параметров, интегрируя с системой Inventor 2022, строит модель «Ящик для деталей» [3].

Ящик для деталей – емкость с ячейками для хранения небольших объектов, таких как: шурупы, гвозди, подшипники, резиновые прокладки, радиодетали для плат и т.д.

Также нужно было реализовать объединение ячеек (дополнительное задание). Необходимо чтобы плагин позволял задавать параметры по умолчанию, а также изменять входные параметры ящика, такие как:

* ширину ящика;
* глубину ящика;
* высоту ящика;
* толщину внешних стенок;
* толщину внутренних перегородок;
* толщину днища ящика;
* количество ячеек на ширину ящика;
* количество ячеек на длину ящика.

## Описание предмета проектирования

Ящик для деталей – емкость с ячейками для хранения небольших объектов, таких как: шурупы, гвозди, подшипники, резиновые прокладки, радиодетали для плат и т.д.

Изображение моделируемого объекта на рисунке 2.1:



Рисунок 2.1 – модель ящика

Измеряемые параметры для плагина:

* W – ширина ящика (150 – 700 мм);
* D – глубина ящика (150 – 700 мм);
* H – высота ящика (50 – 150 мм);
* d1 – толщина внешних стенок (5 – 10 мм);
* w1 – толщина внутренних перегородок (2 – 5 мм);
* h1 – толщина днища ящика (5 – 10 мм);
* N1 – количество ячеек на ширину ящика. Минимум 1, максимум такой, что бы ширина одной ячейки была не менее 10 мм, т.е. должно выполнятся следующее неравенство: , где – введенное пользователем количество ячеек. Если заданное количество не удовлетворяет условию, то при запуске моделирования появится сообщение с предложением сократить число ячеек до максимально доступного при текущих параметрах или отменить моделирование;
* N2 – количество ячеек на длину ящика. Минимум 1, максимум рассчитывается аналогично ширине: индивидуальная ячейка не менее 10 мм, неравенство .

## Выбор инструментов и средств реализации

На основе требований к техническому заданию программа выполнена на языке программирования C# в среде Microsoft Visual Studio 2019 с использованием .NET Framework 4.7.2 [2], библиотека «Inventor» [4] для основных операций в САПР Inventor.

Инструментом тестирования и создания модульных тестов был выбран тестовый фреймворк NUnit [5] версии 3.13.2.

Для реализации пользовательского интерфейса использовалась система для построения настольные приложения WPF [6].

## Назначение плагина

Назначение разрабатываемого плагина обусловлено быстрым моделированием ящиков для деталей, рассчитанных на различное количество ячеек. Благодаря данному расширению, проектировщики могут наглядно рассмотреть спроектированную модель, при необходимости перестроить под необходимые им параметры.

# Обзор аналогов

**Расширение “Пресс-формы 3D express” для Компас-3D**

«Компас» — семейство САПР, универсальная система автоматизированного проектирования, позволяющая в оперативном режиме выпускать чертежи изделий, схемы, спецификации, таблицы, инструкции, расчётно-пояснительные записки, технические условия, текстовые и прочие документы. Система ориентирована на оформления документации в соответствии с ЕСКД, ЕСТД, СПДС и международными стандартами [7].

Система Пресс-формы 3D Express предназначена для автоматизации анализа 3D-модели детали и проектирования формообразующих пресс-формы, включая боковые ползуны [8].

Основные функции:

* проведение анализа раскрываемости, наличия нулевых или отрицательных уклонов.
* отнесение поверхностей 3D-модели к подвижной или неподвижной частям пресс-формы, к боковым ползунам;
* построение линии и поверхности раскрывания, заплаток внутренних отверстий;
* получение заготовок формообразующих пресс-формы;
* моделирование литниковой системы.

# Описание реализации

Для графического описания абстрактной модели проекта, а также пользовательского взаимодействия (сценарии действия) использован стандарт UML.

UML язык графического описания для объектного моделирования в области разработки программного обеспечения. UML является языком широкого профиля, это – открытый стандарт, использующий графические обозначения для создания абстрактной модели системы, называемой UML – моделью. UML был создан для определения, визуализации, проектирования и документирования, в основном, программных систем. UML не является языком программирования, но на основании UML возможна генерация кода и наоборот. [9]

При использовании UML были простроена диаграмма классов.

## Диаграмма классов

Диаграмма классов – структурная диаграмма языка моделирования UML, демонстрирующая общую структуру иерархии классов системы, их коопераций, атрибутов (полей), методов, интерфейсов и взаимосвязей между ними.

На рисунке 4.1 представлена изначальная диаграмма классов.

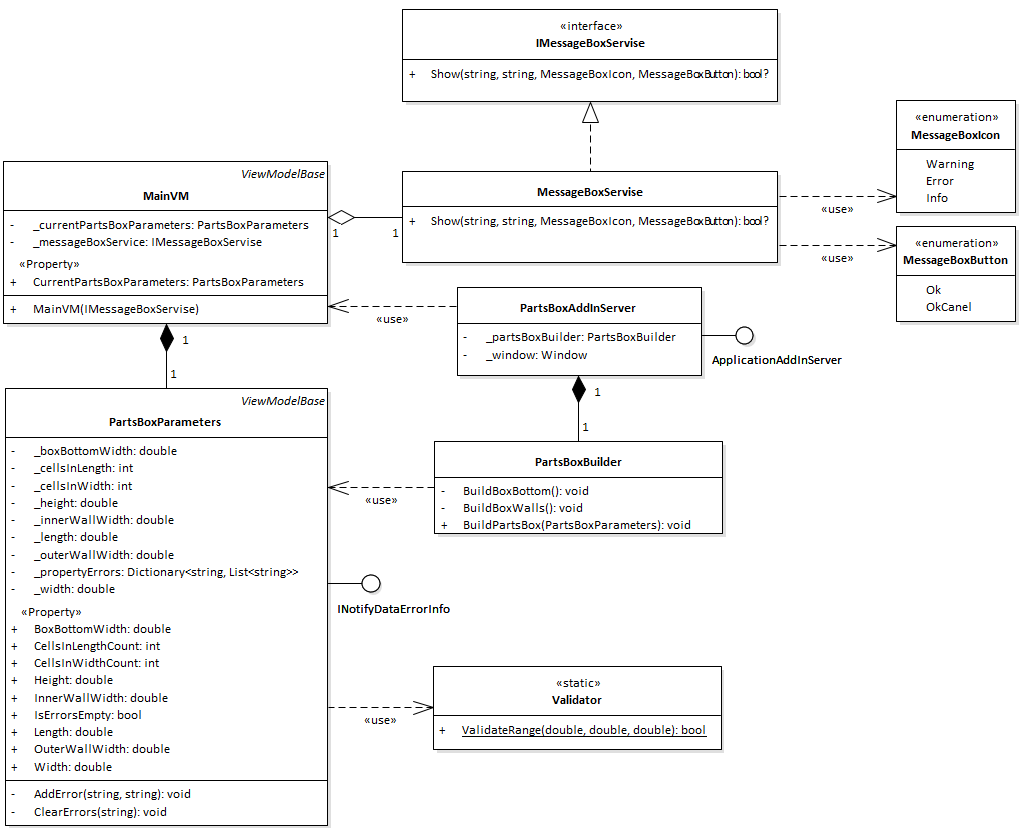


Рисунок 4.1 – Изначальная диаграмма классов

Главной сущностью является PartsBoxAddInServer который реализует интерфейс ApplicationAddInServer обязательный для всех плагинов Inventor. При запуске Inventor с установленным плагином он добавляет в документе новой детали на вкладке “Tools” группу с названием “Parts Box Wizard” с кнопкой “Build”.

PartsBoxAddInServer композирует PartsBoxBuilder – класс содержащий методы построения модели ящика для деталей.

Так же класс PartsBoxAddInServer содержит поле с экземпляром главного окна приложения.

MainVM – модель-представления главного окна, реализует ViewModelBase из библиотеки **MvvmLightLib**. Служит для связи между видом окна и моделью.

PartsBoxParameters – модель, хранит параметры ящика для деталей. Реализует ViewModelBase и интерфейс INotifyDataErrorInfo для проверки введенных данных на форме.

IMessageBoxService – сервис для работы с окнами сообщений.

MessageBoxService реализует IMessageBoxService. В классе содержится метод Show принимающий в качестве аргументов заголовок и текст сообщения, а также элементы перечислений MessageBoxIcon и MessageBoxButton, которые определяют тип картинки и кнопок соответственно.

Статический класс Validator производит проверку присваиваемых в PartsBoxParameters данных.

В итоговом проекте созданы следующие классы и методы, которые отображены на итоговой диаграмме классов (рисунок 4.2).

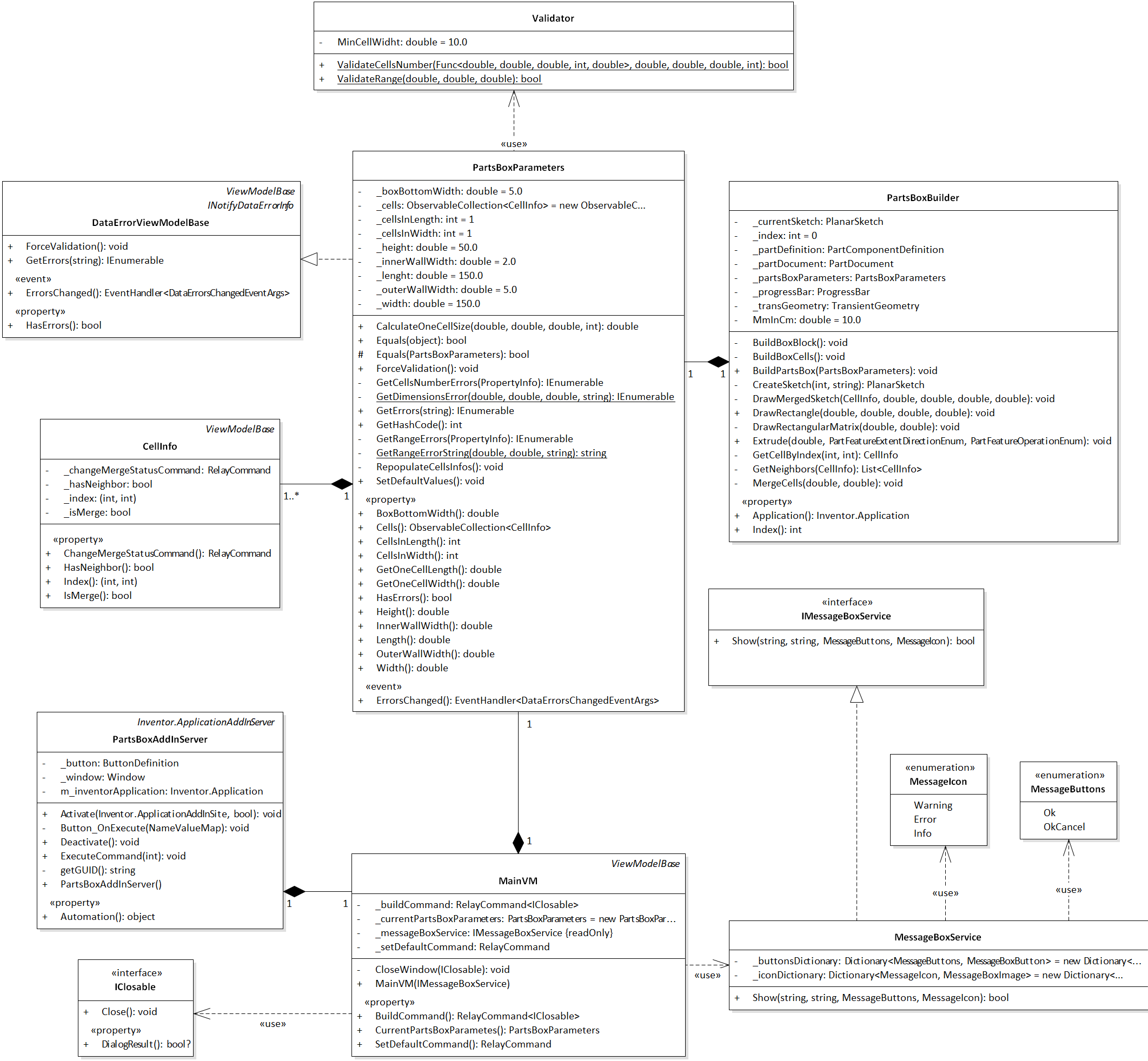


Рисунок 4.2 – Итоговая диаграмма классов

Был добавлен интерфейс IClosable, который реализует главное окно. Данный интерфейс служит для создания возможности закрытия окна через экземпляр его модели представления.

Для реализации дополнительной функциональности был добавлен класс CellInfo, который хранит информацию с данными для объединения ячеек.

Для более простой и удобно поддерживаемой реализации проверки ввода данных был создан класс DataErrorViewModelBase, который реализует интерфейс INotifyDataErrorInfo. PartsBoxParameters реализует данный класс и переопределяет метод проверки и генерации сообщений об ошибках для передачи их визуальной части плагина.

# Описание программы для пользователя

Пользовательский интерфейс состоит из окна, в котором вводятся данные для построения модели коробки для деталей. Если все данные были введены корректно, то при нажатии кнопки “Build” происходит построение модели. При наличии ошибок в вводе параметров построение не произойдет и пользователю выведется сообщение об ошибке. Кнопка “Clear All” очищает все поля параметров.

Так же под списком параметров расположен элемент для объединения ячеек. При нажатии на ячейку она отмечается как “ячейка для объединения” и меняет свой цвет на зеленый. При повторном нажатии выделение снимается. При построении детали ячейки, выделенные зеленым цветом, будут объединены.

На рисунке 5.1 представлен макет интерфейса программы.

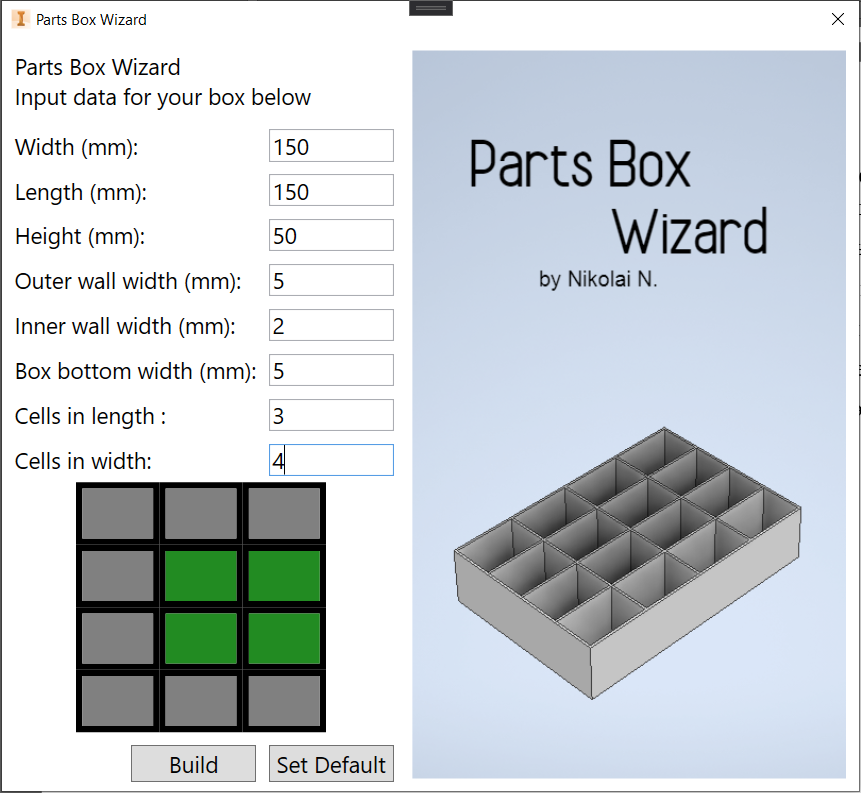


Рисунок 5.1 – Пользовательский интерфейс PartsBoxBuilder

Если ввести неверные параметры, то кнопка Build будет заблокирована для нажатия, а поля с неправильными данными будут выделены красным цветом и при наведении на них курсором мыши будет появляться сообщение с описанием ошибки. На рисунке 5.2 представлен макет интерфейса с некорректно введенными данными.

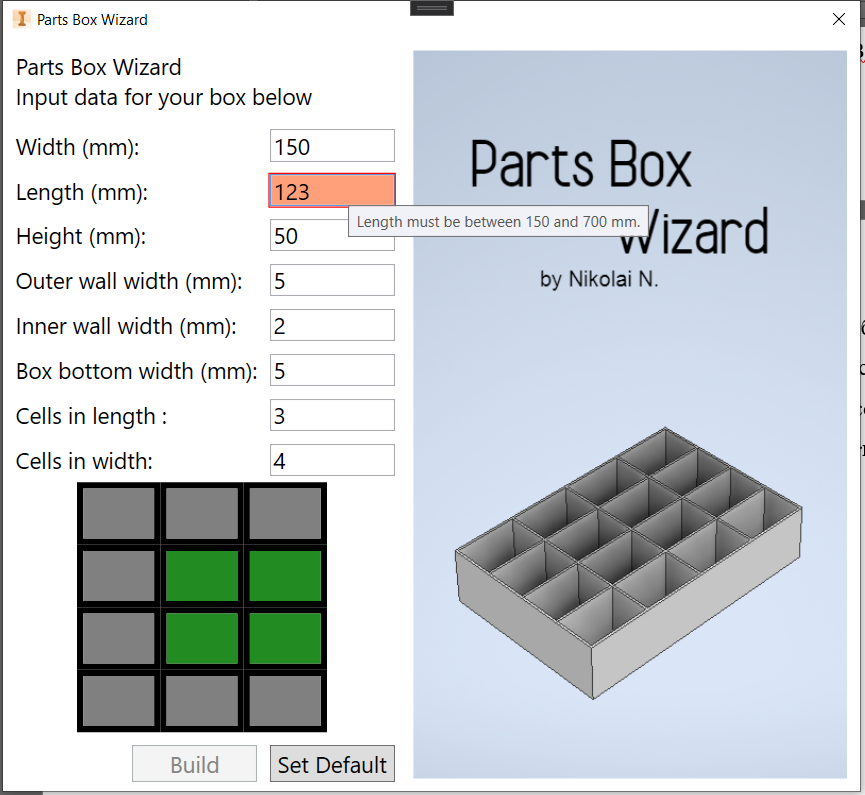


Рисунок 5.1 – Макет интерфейса с неверно введенными данными

После ввода корректных параметров кнопка Build будет снова доступна и при ее нажатии будет построена модель ящика для деталей.

Ящик, построенный по заданным параметрам по умолчанию представлен на рисунке 5.3

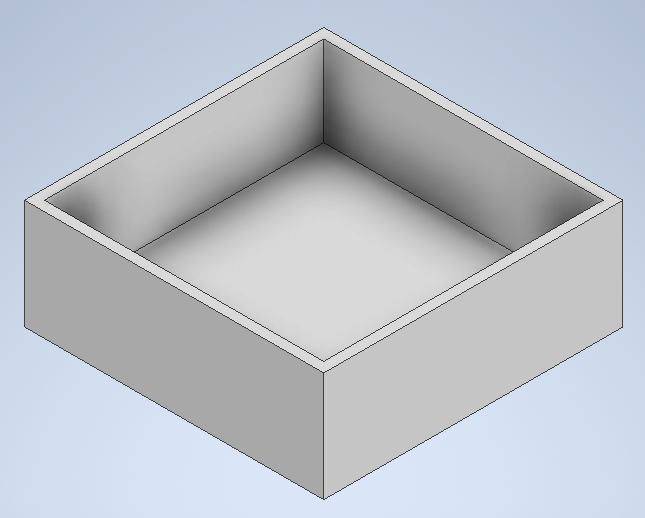


Рисунок 5.3 – Модель ящика, построенная

по параметрам по умолчанию

Модель ящика, построенная с объединением ячеек представлена на рисунке 5.4

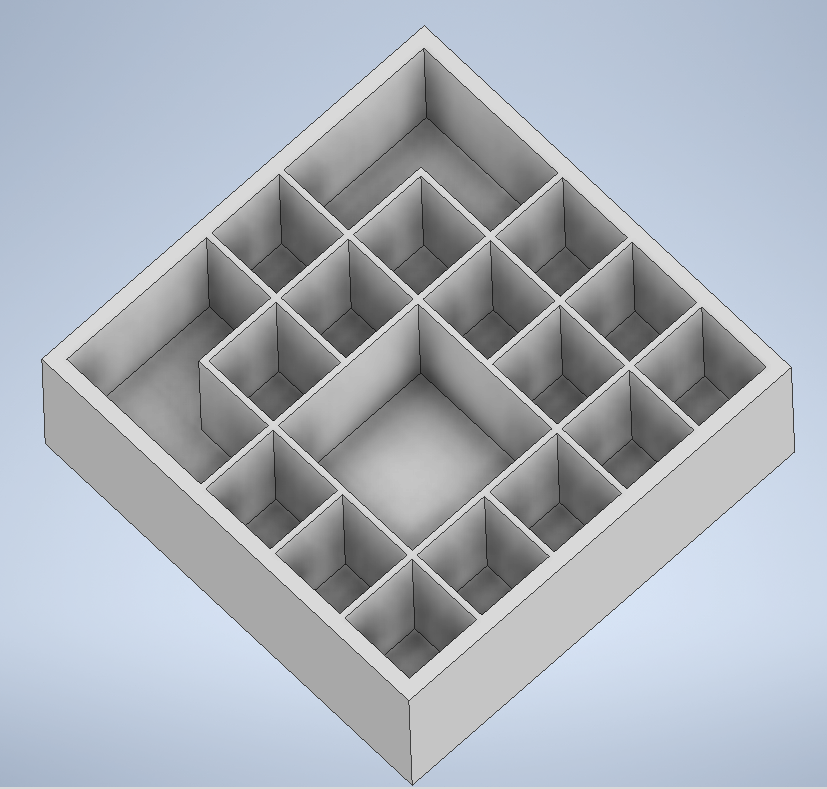


Рисунок 5.4 – Модель ящика с объединёнными ячейками

# Тестирование плагина

Тестирование позволяет убедиться в работоспособности программы, выявлять ошибки при изменении какого-либо функционала.

## Функциональное тестирование

При функциональном тестировании проверялось корректность работы плагина «Ящик для деталей», а именно, соответствие полученного результата в виде трехмерной модели, с входными параметрами. [10]

Проведено тестирование максимальных и минимальных параметров модели.

На рисунке 6.1 представлена модель с минимально введенными параметрами.

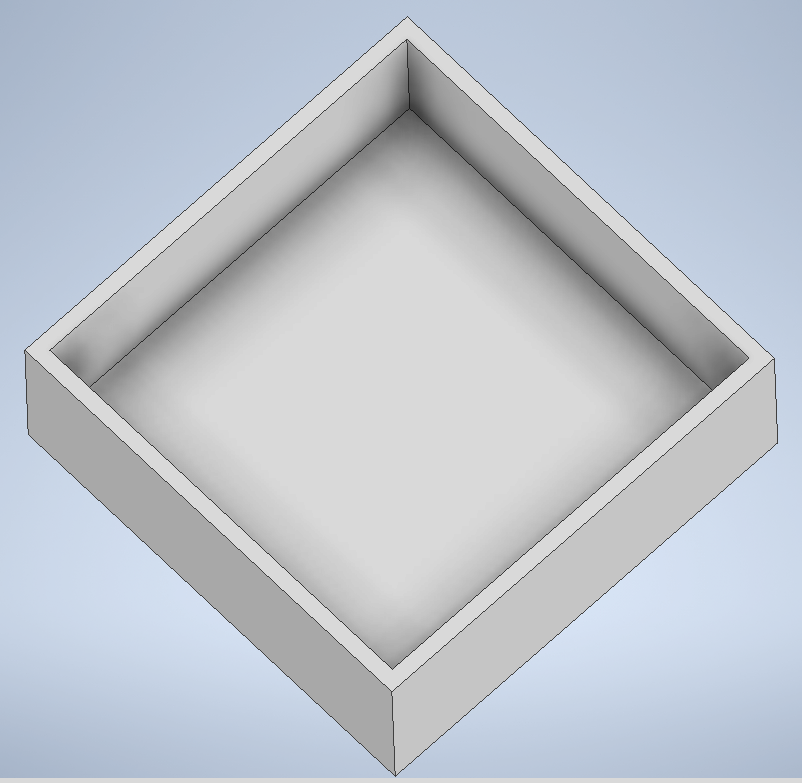


Рисунок 6.1 – Модель ящика с минимальными введенными параметрами.

На рисунке 6.2 представлена модель ящика с максимальными введенными параметрами.

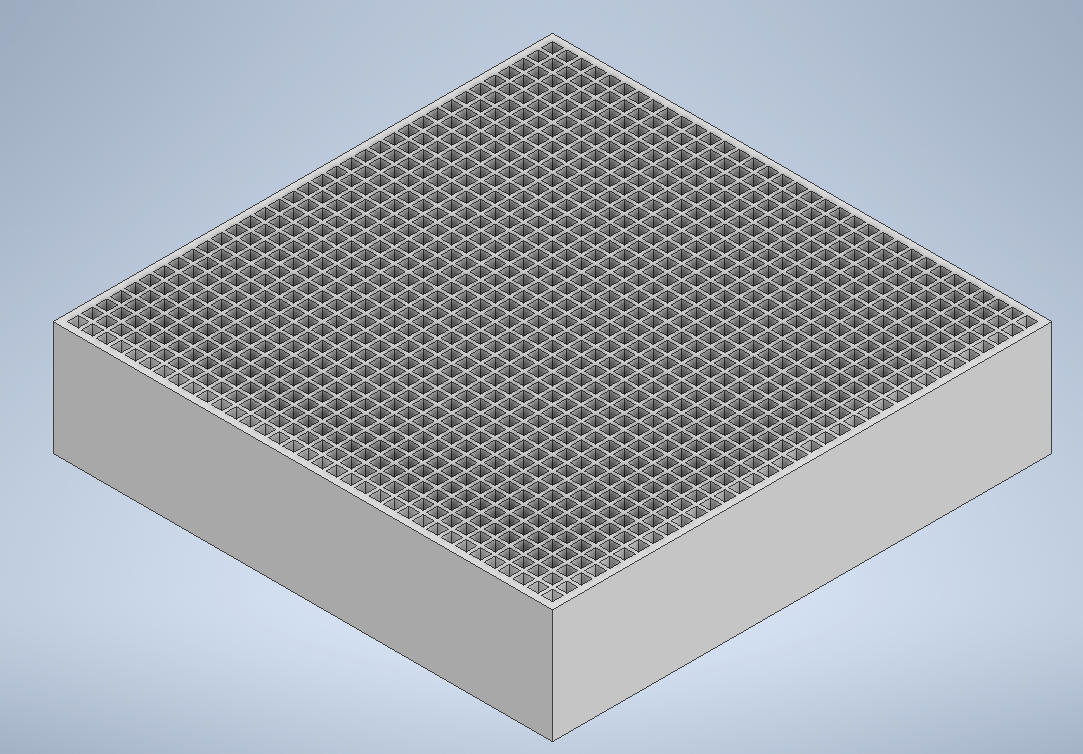


Рисунок 6.2 – Модель ящика с максимальными параметрами

На рисунке 6.3 представлена модель ящика с объединёнными ячейками.

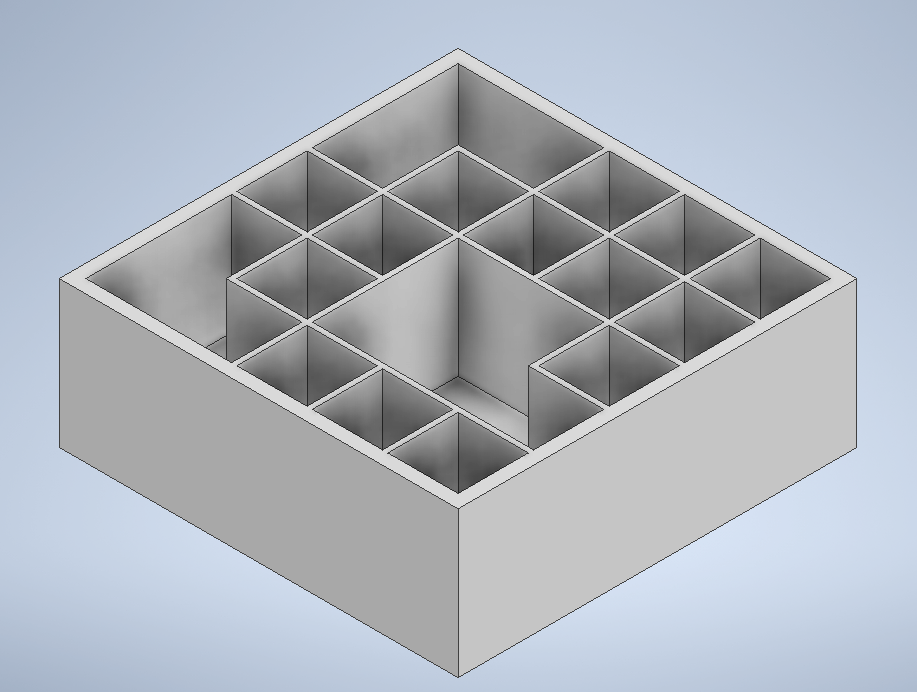


Рисунок 6.3 – Модель ящика с объединёнными ячейками

## Модульное тестирование

В целях проверки корректности работы методов и свойств классов при помощи тестового фреймворка NUnit версии 3.13 проведено модульное тестирование, проверялись открытые поля, свойства и методы. Были протестированы классы модели: PartsBoxParameters, CellInfo, Validator. На рисунке 6.4 представлена информация о модульном тестировании программы. Покрытие модели тестами составило сто процентов.

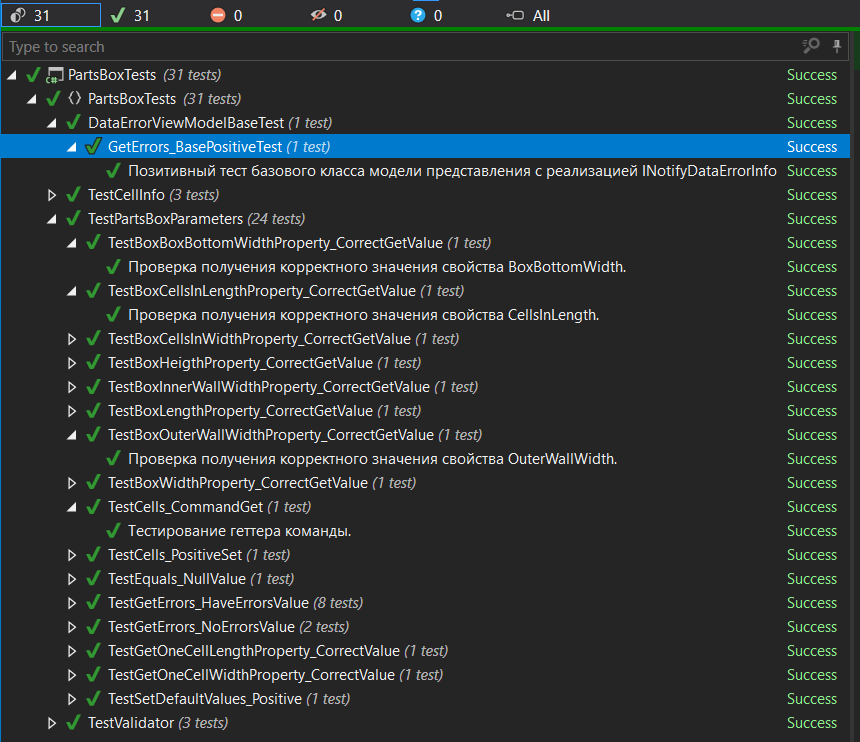


Рисунок 6.4 – Модульное тестирование плагина

## Нагрузочное тестирование

В целях проверки производительности работы плагина, было проведено нагрузочное тестирование. Тестирование производилось на ПК со следующей конфигурацией:

– ЦП Intel Core i3-8145U 2.10 ГГц;

– 8 ГБ ОЗУ;

– графический процессор объемом памяти 4 ГБ.

Для нагрузочного тестирования был создан проект консольного приложения с бесконечным циклом построения детали. Для измерения времени был использован класс Stopwatch. Первое тестирование заключалось в построении детали с минимальными параметрами. На рисунке 6.5 представлен график зависимости времени от количества построенных деталей.

Рисунок 6.5 – График зависимости времени построения

от количества деталей

На рисунке 6.6 представлен график зависимости загруженности памяти от количества построенных деталей

Рисунок 6.6 – График зависимости загруженности памяти

от количества построенных деталей

Тестирование длилось два часа десять минут, было построено 1896 моделей ящика с минимальными параметрами. Исходя из графика, представленного на рисунке 6.5 можно выделить линейную зависимость времени построения детали от ее номера. Из графика 6.6 какой-либо зависимости в данном варианте тестирования выделить нельзя. Все резкие скачки потребления памяти обусловлены открытием и закрытием других приложений во время тестирования (большую часть ОЗУ занимала работа браузера Google Chrome).

Так же немаловажным фактором является использование файла подкачки, а точнее переход в виртуальную память. Виртуальная память — метод [управления памятью](https://ru.wikipedia.org/wiki/%D0%A3%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C%D1%8E) [компьютера](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80), позволяющий выполнять программы, требующие больше [оперативной памяти](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%B8%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D1%8C), чем имеется в компьютере, путём автоматического перемещения частей программы между основной памятью и вторичным хранилищем (например, [твердотельным](https://ru.wikipedia.org/wiki/%D0%96%D1%91%D1%81%D1%82%D0%BA%D0%B8%D0%B9_%D0%B4%D0%B8%D1%81%D0%BA) накопителем)[11].

Исходя из этого было решено провести второе нагрузочное тестирование изменив следующие параметры:

* увеличить количество ячеек с одной до двадцати пяти;
* отключить файл подкачки.

На рисунке 6.7 представлен график зависимости времени от количества построенных деталей для тестирования построения ящика с двадцатью пятью ячейками.

Рисунок 6.7 – График зависимости времени от

количества построенных деталей для ящика

с двадцатью пятью ячейками

На рисунке 6.8 представлен график зависимости загруженности памяти от количества построенных деталей для модели ящика с двадцатью пятью ячейками

Рисунок 6.8 – График зависимости загруженности памяти

от количества построенных деталей для ящика

с двадцатью пятью ячейками

Из графика 6.7, как и в первом варианте нагрузочного тестирования можно отметить линейную зависимость времени построения детали от её номера.

Так же можно отметить, что большее количество ячеек увеличивает среднее время построения модели. Финальный скачок на данном графике обусловлен запуском браузера Google Chrome с большим количеством открытых вкладок, который был сделан с целью увеличить нагрузку на систему и оценить влияние работы других приложений на построение модели. Сделан вывод о том, что работающие сторонние приложения, занимающие большое количество памяти ОЗУ резко увеличивают время построения одной детали.

Из графика 6.8 можно сделать вывод о линейной зависимости потребляемой памяти ОЗУ от количества деталей.

# Заключение

В ходе выполнения лабораторных работ были изучены предметная область проектирования, предмет проектирования, аналоги предмета проектирования, API, функциональное и нагрузочное тестирование и на основании полученных данных были спроектированы UML диаграммы классов, разработан плагин для создания 3D моделей «Ящик для деталей» в САПР Inventor 2022 и проведено функциональное и нагрузочное тестирование плагина.

# Список использованных источников

1. Автоматизация вычислительных процедур в прикладных задачах инженерного проектирования [Электронный ресурс]. – URL: https://scienceforum.ru/2014/article/2014000201 (дата обращения: 10.12.2021).
2. Visual Studio [Электронный ресурс]. – URL: https://visualstudio.microsoft.com/ru/ (дата обращения: 10.12.2021).
3. Autodesk Inventor— Википедия. [Электронный ресурс]. — Режим доступа: https://ru.wikipedia.org/wiki/Autodesk\_Inventor (дата обращения 22.10.2021).
4. Разработка приложений для Inventor - Autodesk. [Электронный ресурс]. – URL: https://www.autodesk.ru/autodesk-developer-network/software-platform-russian/develop-inventor (дата обращения: 10.12.2021).
5. NUnit [Электронный ресурс]. – URL: https://nunit.org/ (дата обращения: 10.12.2021).
6. Что такое Windows Presentation Foundation (WPF) [Электронный ресурс]. – URL: https://docs.microsoft.com/ru-ru/visualstudio/designers/getting-started-with-wpf?view=vs-2022 (дата обращения: 10.12.2021).
7. Компас 3D [Электронный ресурс]. – Режим доступа: https://kompas.ru/kompas-3d/about/ Дата обращения (27.10.2021)
8. Пресс-формы 3D express [Электронный ресурс]. - Режим доступа: https://kompas.ru/kompas-3d/application/machinery/press-formy-3d-express/ Дата обращения (27.10.2021)
9. UML. [Электронный ресурс]. – Режим доступа: http://www.uml.org/ (дата обращения 10.12.2021).
10. Функциональное тестирование [Электронный ресурс]. – URL: https://daglab.ru/funkcionalnoe-testirovanie-programmnogo-obespechenija/ (дата обращения: 10.12.2021).
11. Виртуальная память — Википедия. [Электронный ресурс] — Режим доступа: https://ru.wikipedia.org/wiki/Виртуальная память (дата обращения 17.04.2021)