

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

**«Кубанский государственный университет»**

**(ФГБОУ ВО «КубГУ»)**

**Институт среднего профессионального образования**

**(ИНСПО)**

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе**

**«МДК 03.01 Технология разработки**

**программного обеспечения» на тему:**

**Разработка программы расчета энергоснабжения дома**

Выполнила студентка гр. ПКС-1

Соболева Полина Арсеньевна

Руководитель преподаватель

Трубников Ю.Ю.

Краснодар, 2022

СОДЕРЖАНИЕ

**[ВВЕДЕНИЕ](#_Toc119259791)** [3](#_Toc119259791)

**[1.](#_Toc119259792)****[Анализ задания и выбор технологии, языка и среды разработки](#_Toc119259792)** [4](#_Toc119259792)

**[2.](#_Toc119259793)****[Определение структуры программного продукта](#_Toc119259793)** [9](#_Toc119259793)

**[2.1.](#_Toc119259794)****[Анализ процесса обработки информации и выбор структур данных для ее хранения](#_Toc119259794)** [9](#_Toc119259794)

**[2.2.](#_Toc119259795)****[Выбор методов решения задачи и разработка основных алгоритмов предметной области](#_Toc119259795)** [9](#_Toc119259795)

[2.3 Построение структурной схемы программного продукта 13](#_Toc119259796)

**[3.](#_Toc119259797)****[Разработка интерфейса пользователя](#_Toc119259797)** [14](#_Toc119259797)

**[3.1 Построение диаграммы вариантов использования](#_Toc119259798)** [14](#_Toc119259798)

**[3.2.](#_Toc119259799)****[Разработка форм ввода-вывода информации](#_Toc119259799)** [16](#_Toc119259799)

**[4.](#_Toc119259800)****[Разработка диаграммы классов программы](#_Toc119259800)** [19](#_Toc119259800)

**[5.](#_Toc119259801)****[Разработка диаграммы последовательности](#_Toc119259801)** [20](#_Toc119259801)

# **ВВЕДЕНИЕ**

Развитие человеческой цивилизации на протяжении всей её истории сопровождало непрерывное развитие методов и технологий обработки и использования материалов. Одним из важнейших технологических прорывов для человечества стало изобретение термической обработки пищи, воды, что позволило значительно повысить качество пищи и её усваиваемость.

На пути совершенствования методов термообработки и искусственного обогрева помещений, человечество сменило множество материалов: от брёвен и кизяка, до угля, и, наконец, газа.

Современную жизнь невозможно представить без использования газообразного топлива. Использование природного газа в качестве топлива для отопления и приготовления пищи позволяет не только обеспечить комфортные условия для жизни человека, но и значительно сократить затраты, а также уменьшить выбросы вредных веществ в атмосферу.

Однако массовые потребители данного вида топлива ввиду его дешевизны, могут потреблять сильно больше пропускной способности газового счётчика, таким образом приводя к его неисправности, и в дальнейшем переплачивая на его переустановке.

Чтобы не допустить износа газового оборудования и его дальнейшей замены, верным исходом будет провести подсчёты средних и пиковых расходов топлива. В том числе, это поможет сэкономить не только на амортизации оборудования, но и на непосредственном потреблении газа.

Для массового потребителя газа подобные вычисления могут показаться сложными, т.к. в них достаточно просто допустить ошибку. Поэтому более удобным и надёжным решением в данном вопросе будет использование автоматических вычислений - калькулятор расчётов энергопотребления дома.

# **Анализ задания и выбор технологии, языка и среды разработки**

Задача заключается в разработке приложения «Калькулятор расчёта энергоснабжения дома». Оно должно представлять собой автономного бота, интегрированного с системой чат-ботов мессенджера Telegram.

Согласно документации Telegram, «боты» - это специальные мини-приложения, которые целиком работают внутри приложения Telegram. Работают они благодаря «Bot API» - упрощённой версии Telegram API. Боты обращаются к данному API по HTTPS-протоколу, получают от него сообщения пользователей, которые пользуются им внутри основного приложения Telegram, обрабатывают эти сообщения, и отправляют пользователю ответ.

Для начала разработки необходимо определить технологии программирования, а именно: язык программирования и среду разработки.

Выбор языка программирования будет основываться на уже имеющихся в открытых источниках публичных библиотеках для эффективной разработки Telegram бота, перечисленных в официальной документации к разработке Telegram.

Данные библиотеки предоставляют высокоуровневые абстракции над низкоуровневой логикой (запросы к API, обработка сообщений), что позволяет разработчику сфокусироваться на основной логике работы приложения.

Также они позволяют разработчику создавать более сложные и гибкие приложения, ориентированные на конечного пользователя. Данные библиотеки доступны для следующих платформ, в порядке убывания популярности: PHP, Go, Python, Rust, Kotlin, Node.js, .NET, Swift, TypeScript, Java, Scala, Ruby и т.д.

Разберём плюсы и минусы использования первых трёх из них, а именно:

1. PHP;
2. Go;
3. Python.

PHP - универсальный язык веб-программирования, который можно использовать как для написания сайтов, так и запускаемых приложений, выполняемых на сервере.

Преимущества PHP:

1. Развитое сообщество разработчиков. Язык PHP появился в 1995 году, и на момент 2022 года, согласно статистике W3Techs, на нём написано около 77.6% всех сайтов в мире;
2. Кроссплатформенность. PHP-приложения работают на любой платформе, от Windows, Mac OS, или Linux;
3. Скорость выполнения. Вопреки расхожему мнению, начиная с версии PHP 7, в языке была проведена большая работа по оптимизации скорости выполнения программы.
4. Большое количество инструментов. Ввиду широкого распространения данного языка, для поддержки всей существующей продукции на данном языке было написано большое количество библиотек и фреймворков.

Недостатки PHP:

1. Слабая динамическая типизация. Это позволяет языку производить неявные преобразования типов переменных, например, строк с числами;
2. Ориентированность в основном на создание сайтов. Хотя язык и позволяет разрабатывать серверные приложения по примеру чат-ботов, язык был изначально разработан исключительно как препроцессор гипертекста для сайтов;
3. Поддерживаемость кода и снижение популярности. Согласно статистике Github, общее количество создаваемых изменений (Pull Request) в программах, написанных на PHP составляет около 6.3% от всех изменений среди всех языков программирования. В сравнении с 11% на момент 2014-го года, можно сделать вывод, что на данном языке постепенно всё меньше пишут новых программ.

Go — компилируемый язык с открытым исходным кодом, созданный в Google в качестве альтернативы C++ для решения проблем слишком долгого процесса компиляции, нагруженности языка C++, и ускорения процесса разработки.

Преимущества Go:

1. Статическая типизация. На этапе компиляции выявляются ошибки, допущенные программистом в неправильном использовании данных. Статическая проверка проверяет правильность типов данных, а также правильность использования переменных, констант, функций и т.д.
2. Скорость компиляции. Основным отличием Go от ряда компилируемых языков программирования, как C, C++, является быстрая компиляция в исполняемый файл;
3. Конкурентность. Вместо выделения у оперативной системы настоящего потока, минимальный размер которого 1 мегабайт, Go использует собственную систему легковесных потоков выполнения «горутин», минимальный размер которого 4 килобайта. Это позволяет программе эффективно обрабатывать сотни тысяч операций на одной машине, на одноядерном процессоре.

Недостатки Go:

1. Ограниченный функционал. Применение языка Go - сетевые и серверные приложения. Из-за этого у языка есть проблемы с созданием графических интерфейсов;
2. Низкая выразительность. Ввиду стремления языка к простоте, он не обладает некоторыми общими концепциями из других языков программирования в привычном смысле, как, например: классы, наследование, полиморфизм;
3. На сегодняшний день Go, возможно, является самым новым языком программирования. Но по сравнению с другими языками, Go не является широко распространенным, и в этом его недостаток.

Python – это один из наиболее распространённых скриптовых языков общего назначения. Этот язык универсален, он может быть использован в самых разных сферах деятельности.

Преимущества Python:

1. Простой синтаксис. Один из основных принципов Python - простота, что позволяет просто писать читабельные программы;
2. Сильная динамическая типизация. Несмотря на динамическую по природе типизацию, язык не позволяет производить неявное преобразование типов. Также стоит отметить наличие в языке начиная с версии Python 3.6 аннотаций типов - особых меток в коде, позволяющих разработчику видеть типы переменных, типы аргументов в функциях, и возвращаемых значений функций.
3. Богатая стандартная библиотека и множество сторонних библиотек.

Недостатки Python:

1. Низкая производительность. Это основной недостаток языка, однако в версии Python 3.11 были произведены значительные оптимизации работы интерпретатора, что в среднем ускорило производительность на 10-60%;
2. Повышенное потребление памяти. Продолжением низкой производительности является и повышенное потребление памяти, вследствие того, что для создания переменных используется динамическая область памяти, которую в процессе работы программы также требуется очищать, чем занимается «сборщик мусора».

Для разработки Telegram бота было решено использовать язык Python, благодаря высокой выразительности, простоте, распространённости. В данном случае PHP слишком направлен на разработку сайтов. Go однако проигрывает Python в области выразительности, а также задача не требует от инструмента использование асинхронных вычислений.

# **Разработка диаграммы вариантов использования**

Перед разработкой также требуется разработать диаграмму вариантов использования, т.е. выбрать необходимые для реализации функциональности элементы, их связи и последовательность их исполнения.

Согласно техническому заданию, программа должна решать следующие функции:

1. Ввод пользователем параметров (мощность плиты, стоимость газа);
2. Проведение расчёта среднесуточного, среднемесячного и среднегодового расхода газа;
3. Вывод пользователю вычисленных значений затрат газа.

Адаптируя данные требования возможностей системы к принципам коммуникации с Telegram-ботом, получаем следующую диаграмму (см. Рисунок 1):

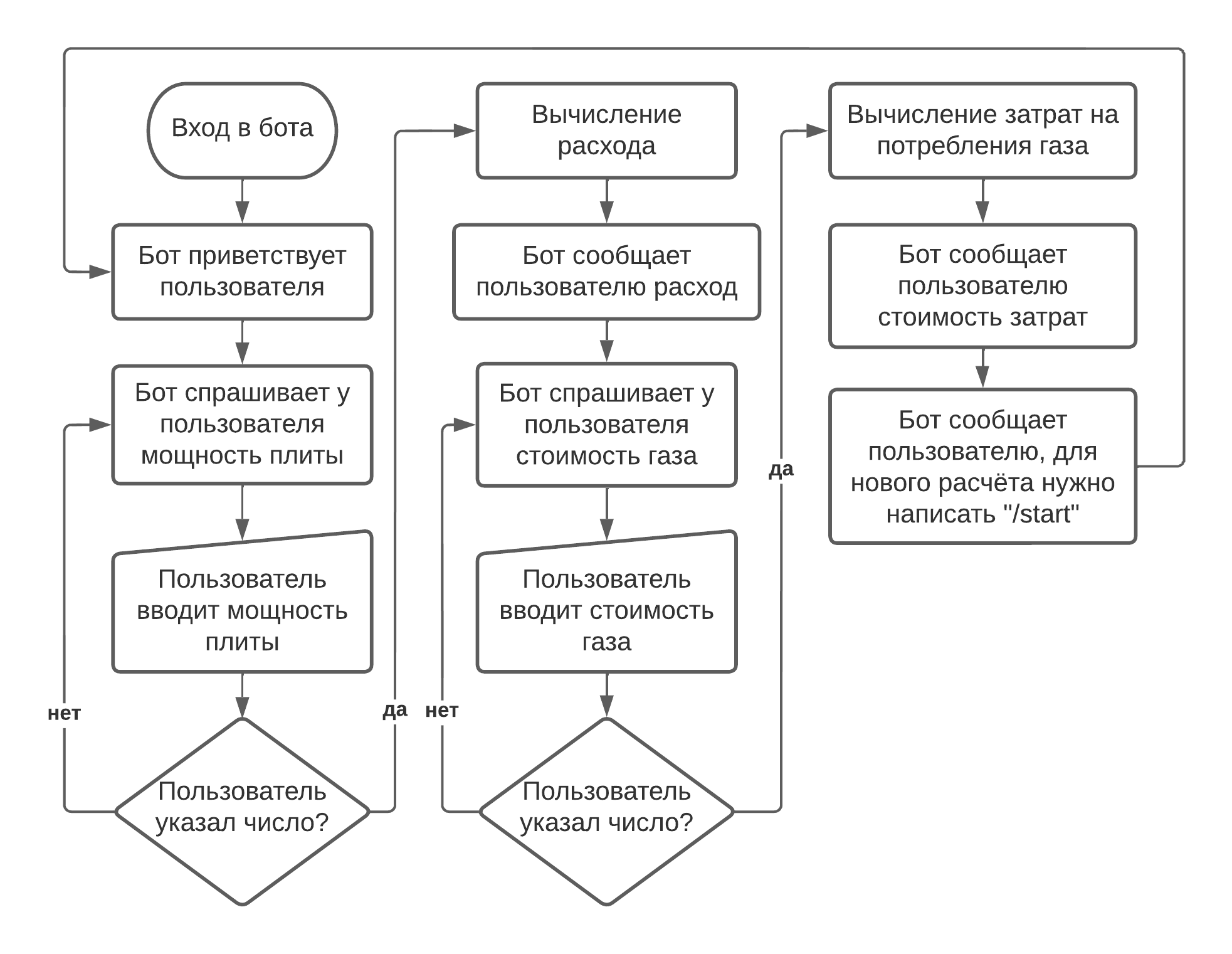


Рисунок 1 - Диаграмма вариантов использования

# **Определение структуры программного продукта**

Исходя из выше построенной диаграммы (см. Рисунок 1) видно, что процесс работы с ботом также состоит из трёх основных процессов: спросить у пользователя мощность и стоимость газа, провести вычисления, и вывести пользователю результат вычислений. Данные три процесса можно разделить на процессы взаимодействия с ботом и процесс вычисления.

Таким образом, целесообразно выделить для работы с вычислениями отдельный класс, единственной ответственностью которого будут вычисления. За взаимодействие с вводом-выводом данных будет отвечать сущность бота, реализованного при помощи сторонней библиотеки.

Перед разработкой методов класса необходимо выработать методику вычисления нужных физических величин.

## **Методы вычислений**

Конечной целью пользователя является получение стоимости потребления газа в день, месяц и год. Т.е. средний расход газа за единицу времени (м3/час), умноженный на тариф (руб/м3).

Согласно ГОСТ 33998-2016 «Приборы газовые бытовые для приготовления пищи. Общие технические требования, методы испытаний и рациональное использование энергии», номинальная мощность устройства вычисляется по следующей формуле:

где Vn — объёмный расход сухого газа при номинальной тепловой мощности и стандартных условиях испытаний, м3/ч;

Hs — теплота сгорания эталонного газа, МДж/м3.

0.278 — константа для преобразования МДж в кВт·час

Из формулы выше можно вывести формулу объёмного расхода газа:

Однако данная формула не учитывает КПД плиты. Поэтому её необходимо преобразовать, умножить HS на значение КПД устройства. Согласно вышеупомянутому ГОСТу, КПД плиты состоит из среднего значения КПД всех горелок (комфорок). Нижняя планка соответствия устройства ГОСТу может считаться пройденной, если КПД открытых горелок плиты равно не менее 52%. Для закрытых (духовой шкаф) не менее 25-35%. Таким образом, формула минимального КПД устройства будет вычисляться следующим образом:

где no — количество открытых горелок;

nз — количество закрытых горелок.

Таким образом, у современной стандартной плиты с 4 открытыми горелками, проходящей требования ГОСТ 33998-2016, минимальное значение КПД составляет 47.6%.

Для упрощения процесса вычислений и уменьшения требований к знаниям пользователя, данное число принимается за константу в вычислениях.

Подставив КПД в формулу объёмного расхода, получаем:

где Qn — номинальная мощность плиты, указанная в паспорте на устройство, кВт.

Данный расход включает в себя одновременное использование всех имеющихся горелок устройства на полную мощность. Однако в хозяйстве редко когда используются одновременно все горелки, поэтому данное значение будет уместно разделить на 2 для получения среднего значения.

В качестве теплоты сгорания эталонного газа, в вычислениях будут использоваться табличные значения из вышеупомянутого ГОСТа для газа с обозначением G20 (чистый метан) c теплотой сгорания 34.02 МДж/м3.

К системе необходимо предъявить ряд требований, выполнение которых обеспечит достижение оптимальных значений всех параметров целевой функции.

Для хранения информации была выбрана структура данных типа массив. Массив — это структура данных с фиксированным и упорядоченным набором однотипных элементов (компонентов). Доступ к какому-либо из элементов массива осуществляется по имени и номеру (индексу) этого элемента. Количество индексов определяет размерность массива.

Так как программа не имеет сложных данных в больших объемах, массива будет достаточно для хранения и обработки информации.

Определим следующие переменные в программе:

|  |  |
| --- | --- |
| Тип и название | Назначение |
| string[] sizeForWood | Хранение массива размеров для вида «Саморезы по дереву» |
| string[] sizeListForRoofing |  |
| string[] sizeForUni |  |
| string[] sizeForSecret |  |
| string[] sizeForMetal |  |
| string[] sizeForPSH |  |
| string[] sizeForGun |  |
| string[] sizeForNails |  |
|  |  |
|  |  |

## **Выбор методов решения задачи и разработка основных алгоритмов предметной области**

Задачами программы является:

1. Расчет количества гвоздей и саморезов, выбранного вида и размера. Исходные данные – введенный пользователем вес.
2. Расчет веса крепежных материалов, выбранного вида и размера. Исходные данные – введенное пользователем количество (в штуках).

В зависимости от задачи, выбранной пользователем программы будет реализован алгоритм действий и расчетов.

Таблица 1 – общий вариант выполнения приложения

|  |  |
| --- | --- |
| Действия исполнителя | Отклик программы |
| 1. Пользователь выбирает вид гвоздей или саморезов. | 1. Система отображает необходимые данные и выводит в поле «Размер» значения доступные для данного вида |
| 1. Пользователь выбирает размер крепежных материалов | 1. Система отображает необходимые данные |
| 1. Пользователь выбирает вид перевода (из килограмм в штуки или из штук в килограммы) | 1. Система отображает необходимые данные, а также меняет значения зависимых полей |
| 1. Пользователь вводит необходимое количество(штук или килограмм) в специальное поле | 1. Система отображает необходимые данные. |
| 1. Пользователь кликает на кнопку расчета | 1. Программа выбирает необходимый алгоритм подсчета и выводит результат на экран. |

Альтернатива 1:

1.Пользователь вводит новое выражение, или изменяет старое.

2.Система переходит к пункту 8 типичного хода событий.

Альтернатива 2:

1.Пользователь выбирает возврат к варианту «Вид» и изменяет данные.

2. Система переходит к пункту 2 типичного хода событий

Альтернатива 3:

1. Пользователь выбирает возврат к варианту «Размер» и изменяет данные.
2. Система переходит к пункту 4 типичного хода событий

Альтернатива 4:

1. Пользователь выбирает возврат к пункту «Тип перевода» и изменяет данные
2. Система переходит к пункту 6 типичного хода событий.

Рассмотрим каждый вариант работы программы более подробно.

Таблица 2 – вариант работы программы при выборе типа перевода «из килограмм в штуки»

|  |  |
| --- | --- |
| Действия исполнителя | Отклик программы |
| 1. Пользователь выбирает вид гвоздей или саморезов. | 1. Система отображает необходимые данные и выводит в поле «Размер» значения доступные для данного вида |
| 1. Пользователь выбирает размер крепежных материалов | 1. Система отображает необходимые данные |
| 1. Пользователь выбирает вид перевода(из килограмм в штуки или из штук в килограммы) | 1. Система отображает необходимые данные, а также меняет значения зависимых полей |
| 1. Пользователь вводит необходимое количество веса в килограммах. | 1. Система отображает необходимые данные и выбирает массив значений для дальнейшего расчета. |
| 1. Пользователь кликает на кнопку расчета | 1. Программа выполняет необходимый алгоритм подсчета и выводит результат на экран. |

Данный алгоритм показывает, как программа будет работать при выборе пользователем варианта «Перевод из килограмм в штуки».

Таблица 3 - вариант работы программы при выборе типа перевода «из штук в килограммы»

|  |  |
| --- | --- |
| Действия исполнителя | Отклик программы |
| 1. Пользователь выбирает вид гвоздей или саморезов. | 1. Система отображает необходимые данные и выводит в поле «Размер» значения доступные для данного вида |
| 1. Пользователь выбирает размер крепежных материалов | 1. Система отображает необходимые данные |
| 1. Пользователь выбирает вид перевода(из килограмм в штуки или из штук в килограммы) | 1. Система отображает необходимые данные, а также меняет значения зависимых полей |
| 1. Пользователь вводит необходимое количество веса в килограммах. | 1. Система отображает необходимые данные и выбирает массив значений для дальнейшего расчета. |
| 1. Пользователь кликает на кнопку расчета | 1. Программа выполняет необходимый алгоритм подсчета и выводит результат на экран. |

Альтернативы соответствуют альтернативам общего алгоритма приложения.

Данный алгоритм отображает работу программы при выборе пользователем варианта «перевод из штук в килограммы». При подсчете используются нормы ГОСТов для каждого крепежного материала.

## Построение структурной схемы программного продукта

Структурной называют схему, отражающую состав и взаимодействие по управлению частей разрабатываемого ПО.

Структурными компонентами программной системы или программного комплекса могут служить программы, подсистемы, базы данных, библиотеки ресурсов. Структурная схема программного комплекса демонстрирует передачу управления от программы-диспетчера (главной программы) соответствующей программе(методу).

Следующая схема отражает структуру и взаимодействие компонентов программного продукта «программы расчета веса и количества

саморезов и гвоздей»:

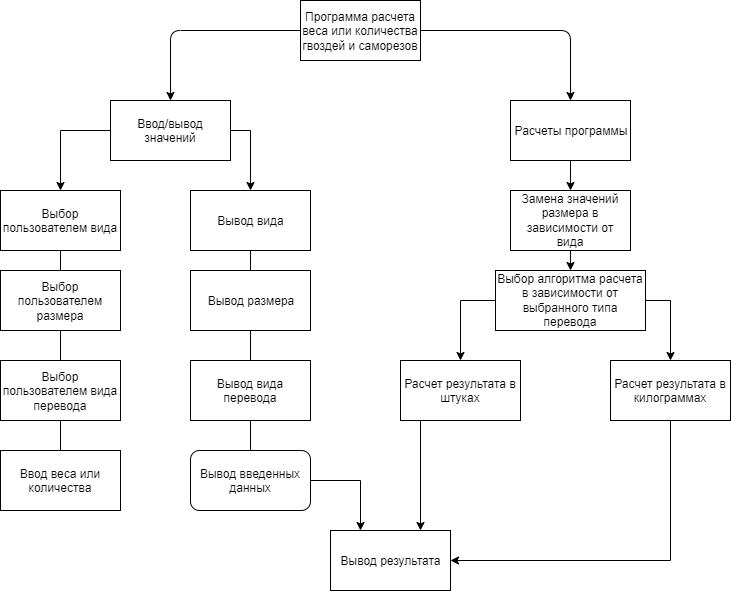


Рисунок 1 – структурная схема программного продукт

# **3. Разработка интерфейса пользователя**

## **3.1 Построение диаграммы вариантов использования**

UML — это сокращение от Unified Modeling Language, он является стандартизированным языком моделирования, состоящим из интегрированного набора диаграмм, разработанных, чтобы помочь разработчикам систем и программного обеспечения в определении структуры и взаимодействии пользователя с системой.

UML представляет собой набор лучших инженерных практик, которые доказали свою эффективность в моделировании больших и сложных систем и является очень важной частью разработки объектно-ориентированного программного обеспечения.

UML использует в основном графические обозначения, чтобы выразить дизайн программных проектов. Использование UML помогает проектным группам общаться, изучать потенциальные проекты и проверять архитектурный дизайн программного обеспечения.

UML-диаграммы - это способ наглядно описать архитектуру, проектирование и реализацию комплексных программных систем.

Диаграммы вариантов использования показывают, как пользователи(агенты) взаимодействуют с системой. Они служат обобщенной моделью связей между агентами и системами и объясняют устройство системы для аудитории без особой технической подготовки.

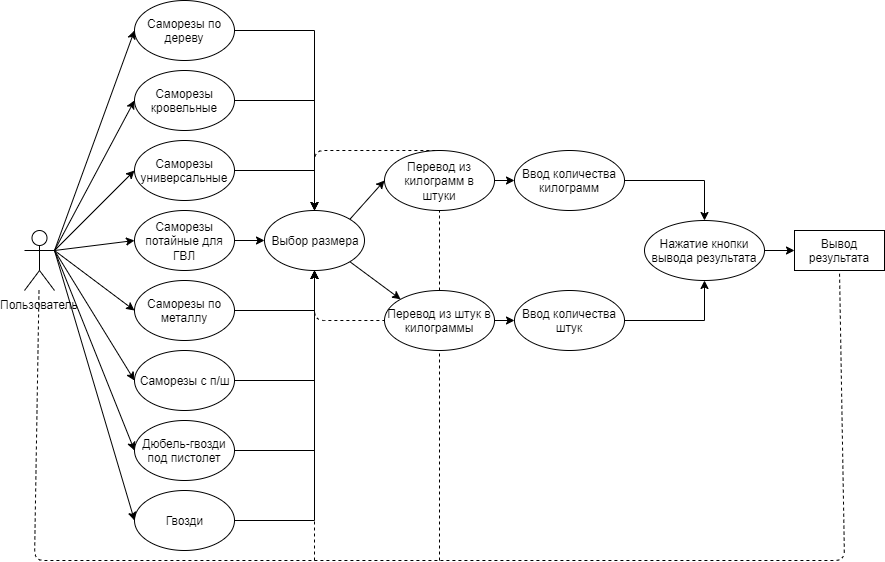


Рисунок 2 – Диаграмма вариантов использования

Таблица 4 – Условные обозначения для UML-диаграммы

|  |  |
| --- | --- |
| Условное обозначение | Пояснение |
|  | Агент(пользователь) приложения |
|  | Вариант взаимодействия пользователя с системой |
|  | Входные/выходные данные |
|  | Альтернативные варианты использования |
|  | Основные варианты использования |

## **Разработка форм ввода-вывода информации**

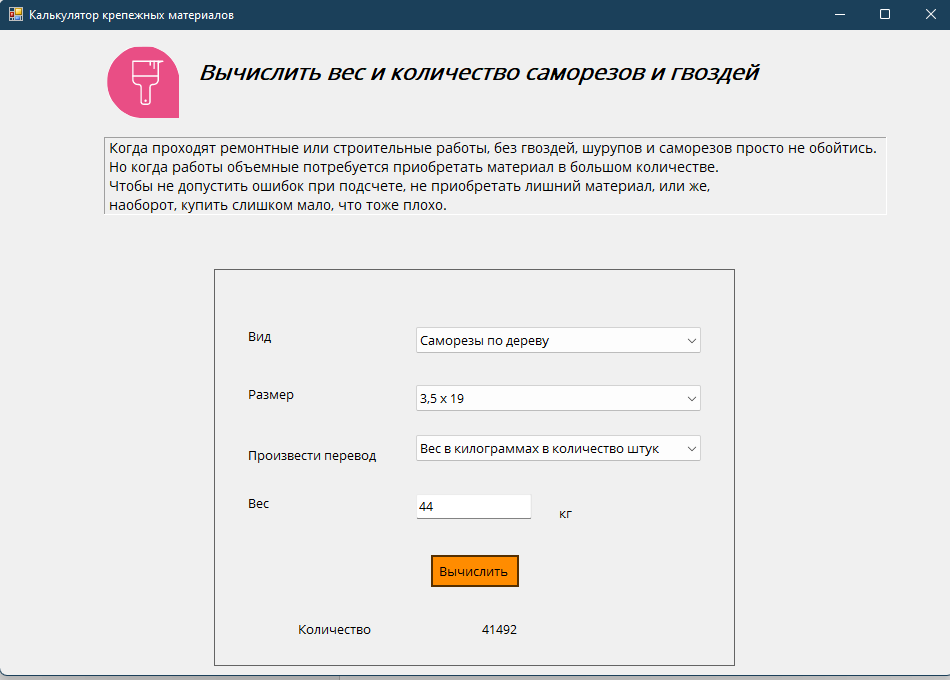


Рисунок 3 – общий вид пользовательской формы

Программа имеет одну пользовательскую форму, в которой происходит всё взаимодействие пользователя с программой. Форма является главной и содержит следующие поля:

* Заголовок приложения
* Описание
* Вид
* Размер
* Произвести перевод
* Вес/количество
* Кнопка «Вычислить»
* Поле вывода результата

Разберем каждое поле подробнее.

Поле «Вид» представлено в виде выпадающего списка видов саморезов и гвоздей. Пользователь не может редактировать значения поля, возможен только выбор одного из представленных значений.



Рисунок 4 – поле «Вид»

Поле «Размер» представлено в виде выпадающего списка размеров. Перечень размеров изменяется в зависимости от выбора в предыдущем поле «Вид». Редактирование поля пользователем невозможно. Возможен выбор готового значения.



Рисунок 5 – поле «Размер»

Поле «Произвести перевод» также представлено в виде выпадающего списка, состоящего из двух значений «Вес в килограммах в количество штук» и «Количество штук в килограммы».



Рисунок 6 – поле «Произвести перевод»

Поле «Вес»/ «Количество» представлено в виде строки для ввода пользовательских значений. Оно может принимать только числовые значения. Подпись к полю изменяется в соответствии с выбором пользователя в поле «Произвести перевод».



Рисунок 7 – поле «Вес»/ «Количество»

Кнопка «Вычислить» запускает метод, вычисляющий итоговое значение по формуле необходимой для выбранной пользователем совокупности значений.

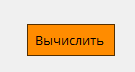


Рисунок 9 – кнопка «Вычислить»

Поле вывода результата имеет подпись «Вес» или «Количество». Подпись зависит от значения выбранного в поле «Произвести перевод», если выбрано значение «Вес в килограммах в количество штук», то значением подписи становится текст «Количество:», если выбрано значение «Количество штук в килограммы», то значением подписи становится текст «Вес:».



Рисунок 10 – поле вывода результата

# **Разработка диаграммы классов программы**

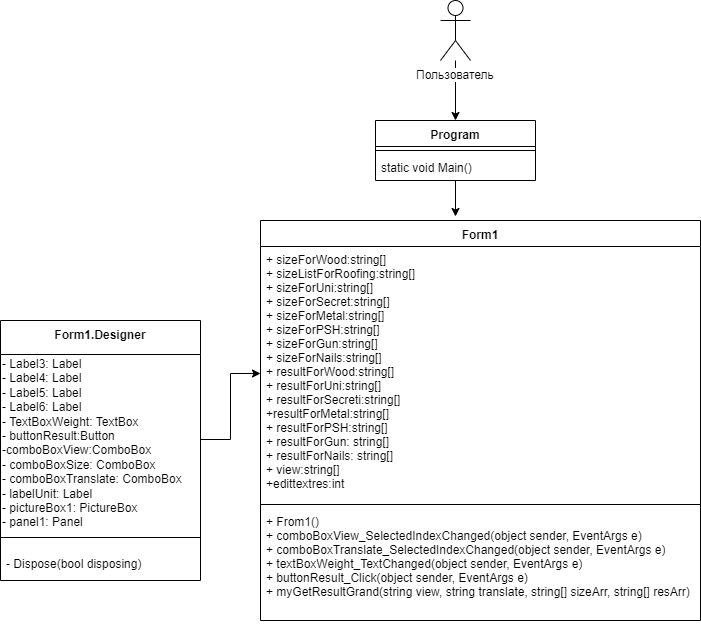


Рисунок 11 – UML диаграмма классов

Программа состоит из трёх основных классов. После запуска программы пользователем запускается класс Program, в котором вызывается метод Main(), вызывающий класс Form1, и запускающий графическую форму приложения.

Класс Form1 содержит следующие поля: sizeForWood:string[],sizeListForRoofing:string[],sizeForUni: string[], sizeForSecret:string[], sizeForMetal: string[], sizeForPSH:string[],sizeForGun:string[], sizeForNails:string[] – данные поля содержат в себе массив значений размеров для различных типов саморезов и гвоздей.

resultForWood:string[], resultForUni:string[],resultForSecret:string[], resultForMetal:string[], resultForPSH:string[], resultForGun:string[],resultForNails:string[] – данные поля содержат в себе массив значений для вычисления конечного результата для каждого из видов гвоздей и саморезов.

view:string[] – поле содержащее в себе массив со значениями видов саморезов и гвоздей.

edittextres: int – поле предназначено для хранения числа веденного пользователем с клавиатуры.

Класс Form1.Designer содержит поля, инициализирующие поля формы.

# **Разработка диаграммы последовательности**