

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):

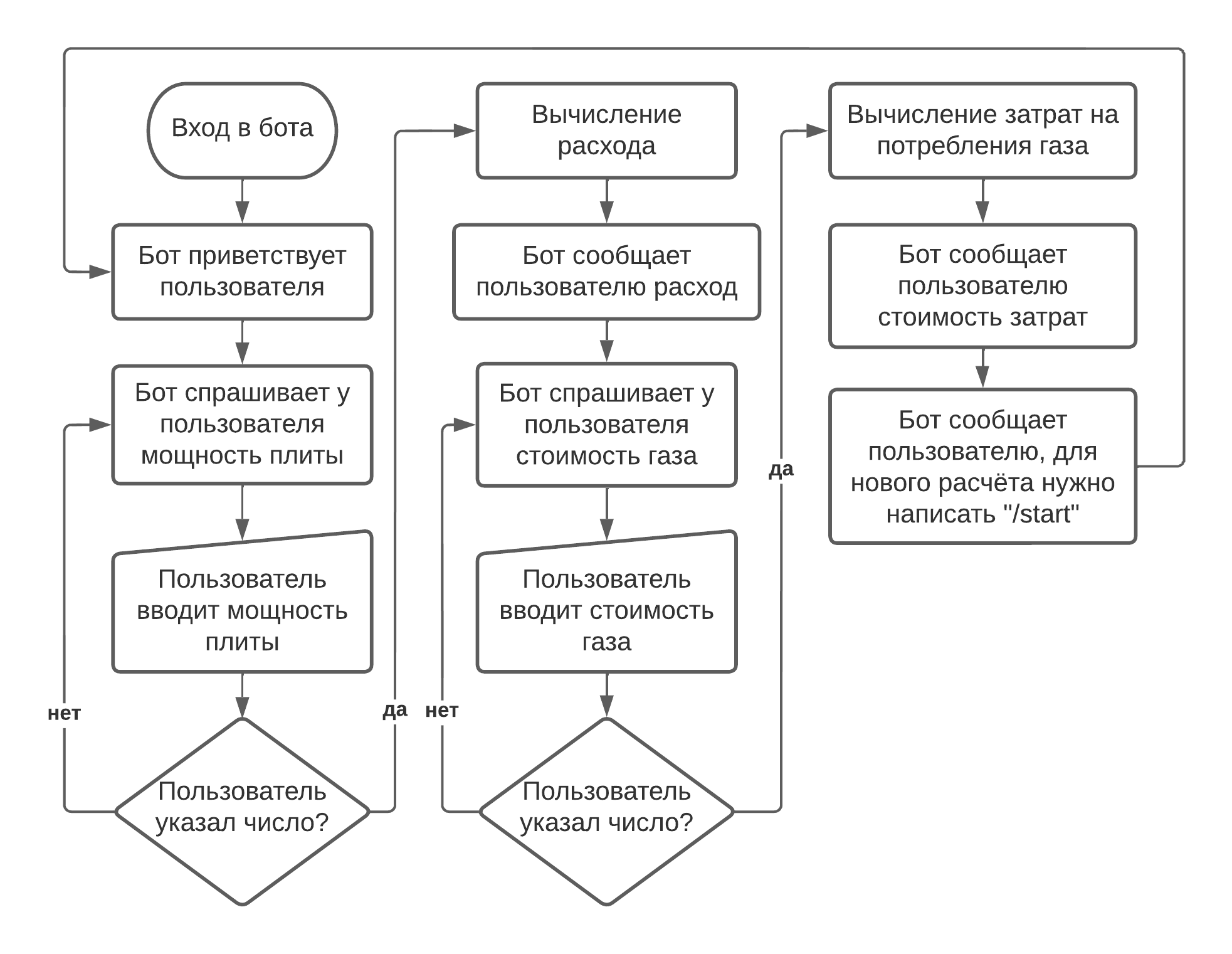


Рисунок 1 - Диаграмма вариантов использования

# **Определение структуры программного продукта**

Под структурой программного продукта понимается совокупность его элементов, их взаимодействие между собой и с внешней средой, а также иерархия и взаимосвязь этих элементов.

Исходя из выше построенной диаграммы (см. Рисунок 1) можно увидеть, 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.

Также до процесса вычисления необходимо сделать проверку на то, что данные, которые ввёл пользователь, не содержат ошибок. Так, если пользователь ввёл не число, а какой-либо иной текст, то бот должен распознать это и попросить пользователя ввести именно число. Или, если пользователь ввёл число, содержащее ошибку (например, содержит запятую вместо точки), то программа также должна уведомить пользователя об ошибке и предложить ему ввести правильный вариант записи.

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. Это важно, потому что от КПД, т.е. эффективности сгорающего газа передавать тепло на поверхность посуды, зависит время приготовления пищи, следовательно, времени использования газа. 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%.

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 c теплотой сгорания 34.02 МДж/м3.

Таким образом, формула для расчёта среднесуточной цены на потребление газа будет иметь следующий вид:

где T — тариф на газ, руб/м3.

Vср — средний расход газа в час.

Используя данную формулу, получаем среднемесячный расход:

А также среднегодовой расход:

## **Декомпозиция элементов системы**

В результате анализа диаграммы вариантов использования и разработки методов вычислений, была спроектирована структурная схема будущего программного продукта (см. Рисунок 2):

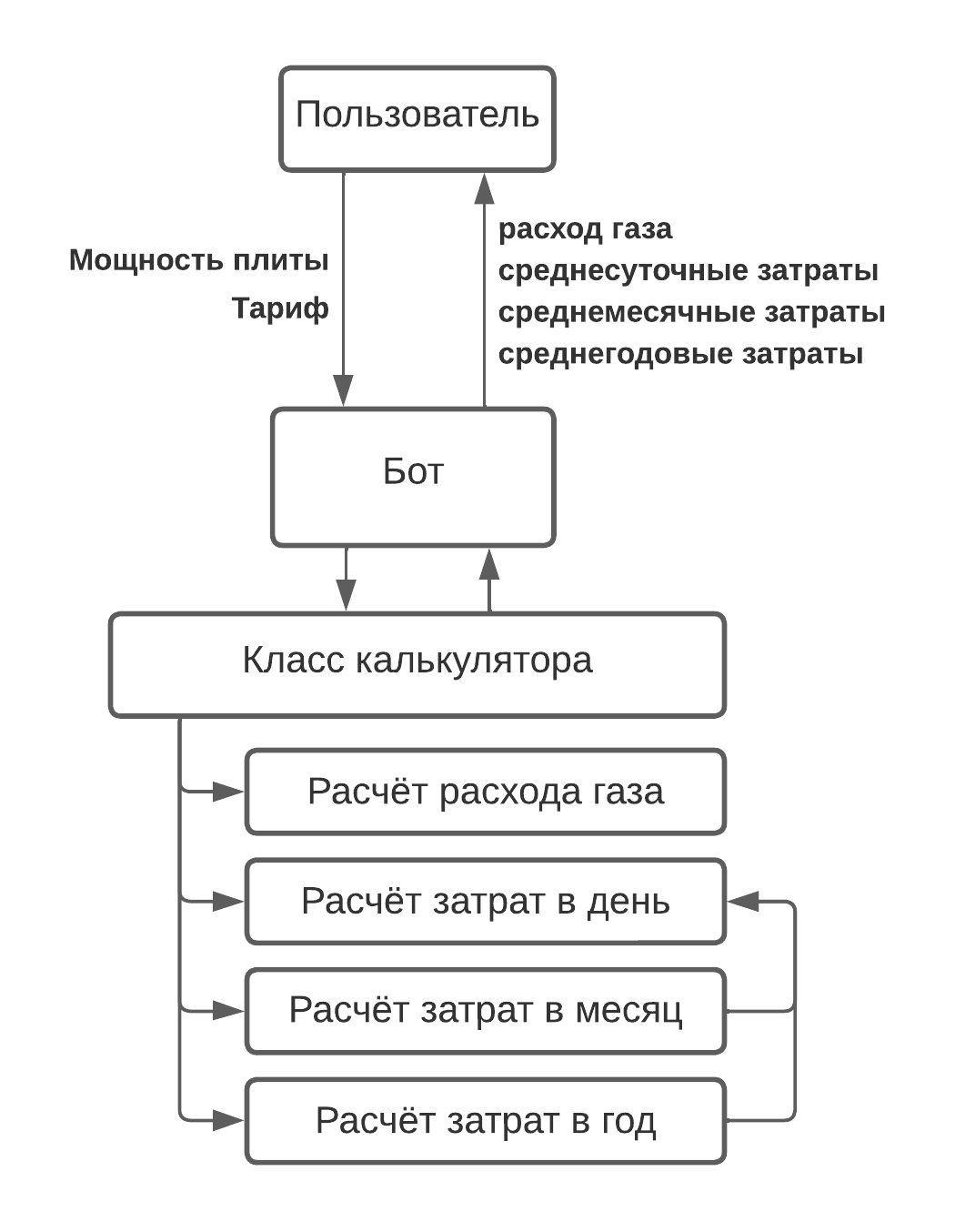
****

Рисунок 2 Структурная схема программного продукта

На данной схеме изображено структурное деление программы на компоненты в виде сущности бота, класса калькулятора и его методов. Также в данной схеме также участвует пользователь, который подаёт на вход к программе мощность плиты и тариф на газ, и получает от него ответ.

Однонаправленные параллельные стрелки между сущностями отображают продолжение коммуникации бота с пользователем. Т.е. мощность плиты и тариф проходят через бота, бот передаёт эти свойства в класс калькулятора, а класс калькулятора возвращает обработанные значения обратно в бота, и бот передаёт их пользователю в удобочитаемом виде.

# **Описание реализации программного продукта**

В соответствии со структурной схемой (см. Рисунок 2), была реализована рабочая программа в виде сервера чат-бота и класса калькулятора для вычислений.

## **Разработка чат-бота**

Для разработки Telegram бота использовалась библиотека pyTelegramBotAPI 4.8.0, дающая простой и минималистичный, но в то же время расширяемый интерфейс для задач коммуникации с Telegram Bot API. Также данная библиотека по названию модуля называется «Telebot».

Перед разработкой любого Telegram бота, необходимо зарегистрировать его в системе. Это возможно при помощи официального бота «BotFather». Бот предоставит ключ доступа, который потребуется для дальнейшей инициализации нового бота.

Инициализация выглядит следующим образом (см. Рисунок 3):

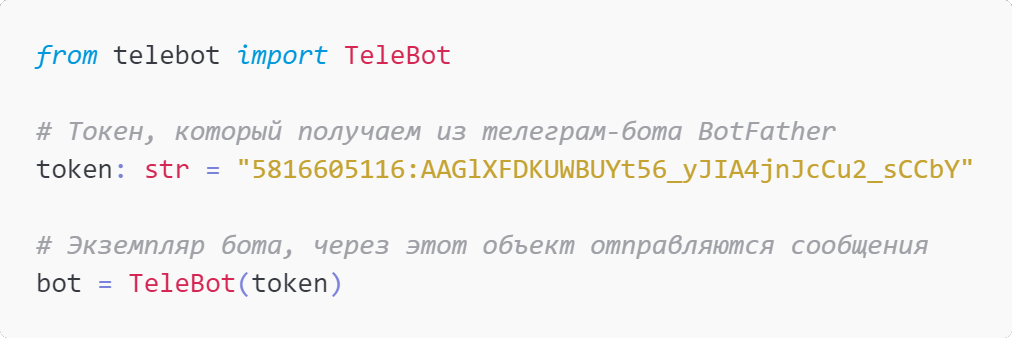


Рисунок 3 Инициализация бота

Для создания экземпляра бота используется класс TeleBot, и в него передаётся ключ доступа. В дальнейшем библиотека будет использовать ключ доступа для получения и отправки сообщений.

В данной библиотеке обмен сообщениями пользователя с ботом реализован через обработчики.

Обработчики - специальные функции, которые вызываются в зависимости от того, какое сообщение было отправлено пользователем. Пример обработчика изображён на рисунке 4:

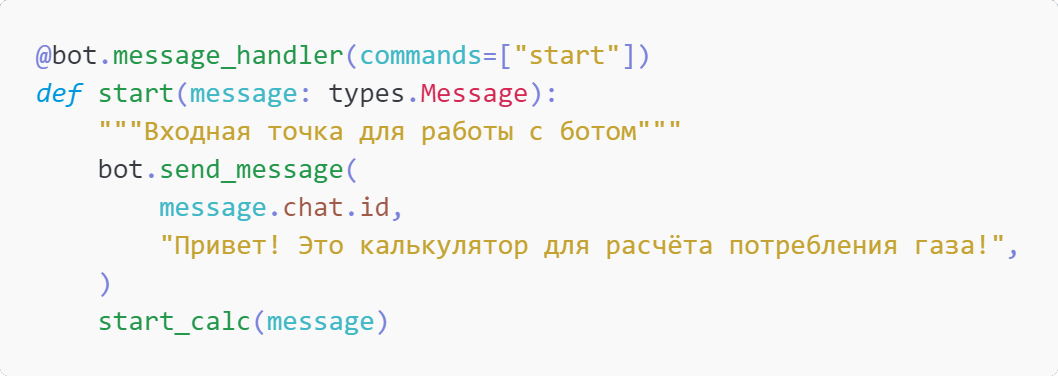


Рисунок 4 Входная точка для начала переписки с ботом

В данном примере (см. Рисунок 4), функция start имеет декоратор bot.message\_handler.

Декоратор в Python - обёртка над функцией, которая изменяет работу того, к чему они применяются. Декоратор bot.message\_handler изменяет поведение так, что функция start вызывается если сообщение пользователя содержит текст «/start».

Затем бот отправляет сообщение при помощи функции bot.send\_message, передавая в неё идентификатор чата с пользователем и текст сообщения. После чего функция вызывает start\_calc, передавая в неё текущее сообщение.

Приведённый выше фрагмент кода зарегистрирует в качестве обработчика на открытие бота функцию «start». Для пользователя это будет выглядеть следующим образом (см. Рисунок 5):

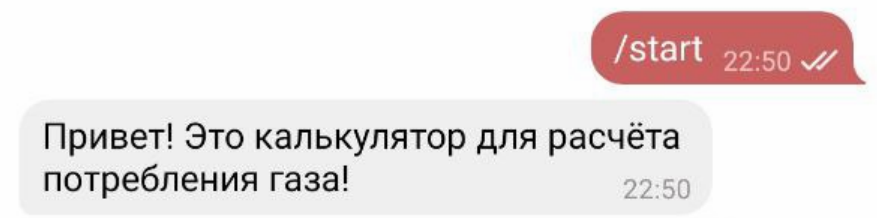


Рисунок 5 Ответ бота пользователю на сообщение «/start»

Далее бот просит пользователя написать мощность плиты. Затем происходит обработка следующего сообщения на то, что оно содержит число. Число может быть как целым, так и дробным.

Для этого в языке Python у переменных с типом строки есть метод isdigit(). Используя данный метод можно определить, можно ли конвертировать введённое пользователем сообщение к типу int. Однако если пользователь ввёл не целое число, необходимо проверить, содержится ли в тексте знак точки. Если он содержится, и обе части справа и слева от неё являются числами, то строка может быть преобразована к типу float.

Кратко:

что такое send\_message, register\_next\_…, Влепить скриншоты из телеграмма

## **Разработка класса калькулятора**

# **Выбор стратегии тестирования и отладка программного средства.**

Кратко:

Были использованы юниттесты, библиотека pytest, самая распространённая. У нас есть всего 1 класс, там 8 методов, и 1 функция для проверки преобразования строки в число

Влепить скрин, что все тесты хорошо пройдены

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной работы было рассмотрено историческое развитие рынка труда в России, рассмотрено современное состояние на рынке труда, было разработано кроссплатформенное программное обеспечение для анализа вакансий сайта HeadHunter в сфере информационных технологий.

Было проанализировано около 20 тысяч вакансий, на основе чего составлены списки навыков, и статистическая информация по состоянию заработной платы, опыта работы, сферы деятельности, графика работы, и т.д.

На основе анализа данных вакансий был выделен следующий список навыков, требуемых работодателям от соискателей на рынке IT: SQL, Git, Linux, Английский язык, JavaScript (а также HTML, CSS), Python, Java, Работа в команде, PostgreSQL, Управление Проектами, ООП, PHP, 1C, MySQL.

Таким образом, на реальных данных по спросу навыков был составлен портрет среднего разработчика: это в первую очередь человек, который должен уметь работать в команде. Ему не обойтись без инструментов командной разработки, как Git. Для работы с данными нужно знать язык запросов SQL, уметь работать с СУБД PostgreSQL, MySQL, MSSQL. Он должен уметь работать с Linux на уровне пользователя или администратора. Также немаловажным является знание английского языка, т.к. в работе специалисту часто требуется читать большие объемы информации на данном языке. В идеале каждый веб-разработчик должен знать базовую вёрстку на HTML, язык стилей CSS, знать минимум 1 JavaScript-фреймворк (например, React.js, Angular, Vue).

# СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Учебный курс «Государственное регулирование рынка труда” НИУ ВШЭ [Электронный ресурс]: https://economics.hse.ru/depe/gosreg/
2. Сачук Татьяна Викторовна. Рынок труда в России последней трети XIX - начале XX вв. : Дис. ... канд. экон. наук : 08.00.03 : СПб., 1998 168 c. РГБ ОД, 61:98-8/780-3
3. Тохтиева Л.Н., Карташов С.А. Возникновение организаций по содействию занятости населения в дореволюционной России // Вестник Рязанского государственного университета им. С. А. Есенина. 2018. №3 (60). URL: https://cyberleninka.ru/article/n/vozniknovenie-organizatsiy-po-sodeystviyu-zanyatosti-naseleniya-v-dorevolyutsionnoy-rossii (дата обращения: 27.05.2022).
4. Карташов С.А., Одегов Ю.Г., Никонова Т.В., Журавлев П.В. История бирж труда // Поиск работы: проблемы и решения (теоретические основы и практические рекомендации). М. : Изд-во Рос. экон. акад., 1995. С. 62.
5. [АТАЯН И.М. Биржи труда в 20-е годы: опыт государственного трудового посредничества с. 117-121 // Социологические исследования. 2000. №4.](https://www.isras.ru/files/File/Socis/04-2000/018.ATAYAN.pdf" \o "https://www.isras.ru/files/File/Socis/04-2000/018.ATAYAN.pdf)
6. Исаев А. Об основных принципах работы бирж труда // Вопросы труда. 1923. № 5-6. С. 19.
7. Государственный сайт Работа России [Электронный ресурс]: Режим доступа: https://trudvsem.ru/
8. Публикации об анализе рынка труда HeadHunter [Электронный ресурс]: Режим доступа: [https://stats.hh.ru/blog](https://stats.hh.ru/blog" \o "https://stats.hh.ru/blog)
9. Документация API HeadHunter [Электронный ресурс]: [https://github.com/hhru/api](https://github.com/hhru/api" \o "https://github.com/hhru/api)
10. Список изменений в Python версии 3.5 [Электронный ресурс]: [https://docs.python.org/3.5/whatsnew/3.5.html](https://docs.python.org/3.5/whatsnew/3.5.html" \o "https://docs.python.org/3.5/whatsnew/3.5.html)