ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

**Московский институт электроники и математики им. А.Н. Тихонова**

Милёхина Елизавета Алексеевна, группа БИТ231

**ТЕЛЕГРАММ-БОТ ДЛЯ РАСЧЁТОВ СИСТЕМЫ ОТОПЛЕНИЯ ОДНОЭТАЖНОГО ЗАГОРОДНОГО ДОМА**

Курсовой проект

по дисциплине «Алгоритмизация и программирование»

студентов образовательной программы бакалавриата  
«Инфокоммуникационные технологии и системы связи»

Студент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись И.О. Фамилия

Научный руководитель

к.т.н., доцент

И.В. Назаров

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись

Москва 2024 г.

**Аннотация**

В ходе работы над курсовым проектом был разработан телеграмм-бот для расчётов системы отопления одноэтажного загородного дома. Особенностью бота является расчёт не только мощности котла, скорости теплоносителя, полный оборот теплоносителя в системе, но и количество секций радиатора в отдельно взятой комнате. Существуют сайты для отдельных расчётов каждой из частей проекта, но бот собирает в себе все эти части и находится во всё больше набирающем популярность Телеграмме.

**Abstract**

During the course project a telegram-bot was developed for calculation of heating system of a one-storey suburban house. The peculiarity of the bot is the calculation not only of the boiler capacity, the heat carrier speed, the full rotation of the heat carrier in the system, but also the number of sections of the radiator in a single room. There are sites for separate calculations of each part of the project, but the bot collects all these parts and is in an increasingly popular Telegram.

**Оглавление**

|  |  |
| --- | --- |
| Обоснование выбора языка программирования и средств разработки | 7 |
| Описание сценария | 8 |
| Спецификация интерфейса | 9 |
| Структура программы | 11 |
| Описание алгоритма | 15 |
| Заключение | 16 |
| Библиографический список | 17 |
|  |  |
|  |  |

**Обоснование выбора языка программирования и средств**

Программа написана на языке программирования Python[1], который сейчас используют во многих областях IT-индустрии, одной из которых является разработка телеграмм-ботов. Python это высокоуровневый язык, который отличается простотой синтаксиса и читаемостью кода, а так же он поддерживает различные парадигмы программирования (например: динамическая типизация). Сейчас Python занимает лидирующие позиции в рейтингах языков программирования TIOBE и PYPL.

В качестве интегрированный среды разработки была выбрана программа PyCharm Community Edition [2], так как она является бесплатной при использовании для некоммерческих целей. Также стоит отметить, что PyCharm обладает функциями автодополнения и анализа кода, в том числе он обладает возможностью удобной навигации по написанному коду.

В качестве платформы, на которой можно увидеть результат программы, я выбрала Telegram [3]. Он с каждым днём набирает всё большую популярность и расширяется круг пользователей. На данной платформе с помощью программы (бота) можно легко коммуницировать с пользователем.

Для написания бота использовалась библиотека Telebot - библиотека Python, которая предоставляет интерфейс для взаимодействия с Telegram Bot API. Она позволяет отправлять сообщения, создавать встроенные клавиатуры, обрабатывать обратные вызовы и выполнять множество других действий с легкостью [4].

**Описание сценария**

Работа программы предусмотрена только для расчётов системы отопления одноэтажных загородных домов, кровля которых имеет малый уклон, что им пренебрегается. Так же считается, что пользователь собирается отапливать всю площадь помещения и вводит данные в метрах. Для расчётов системы отопления дома пользователю требуется отправить ряд параметров (например: количество, толщину элемента, его длину, ширину) и выбрать из предложенного списка (например: материал, погодные условия и тд.).

**Спецификация интерфейса**

Программа предусмотрена для использования веб-версии приложения Telegram (Рис. 1), компьютерной и мобильной версии приложения (Рис. 2).

В левом нижнем углу присутствует, разворачиваемая по кнопке, панель команд, применимых в данном боте с описанием того, что по её вызову выполняется (Рис. 3).

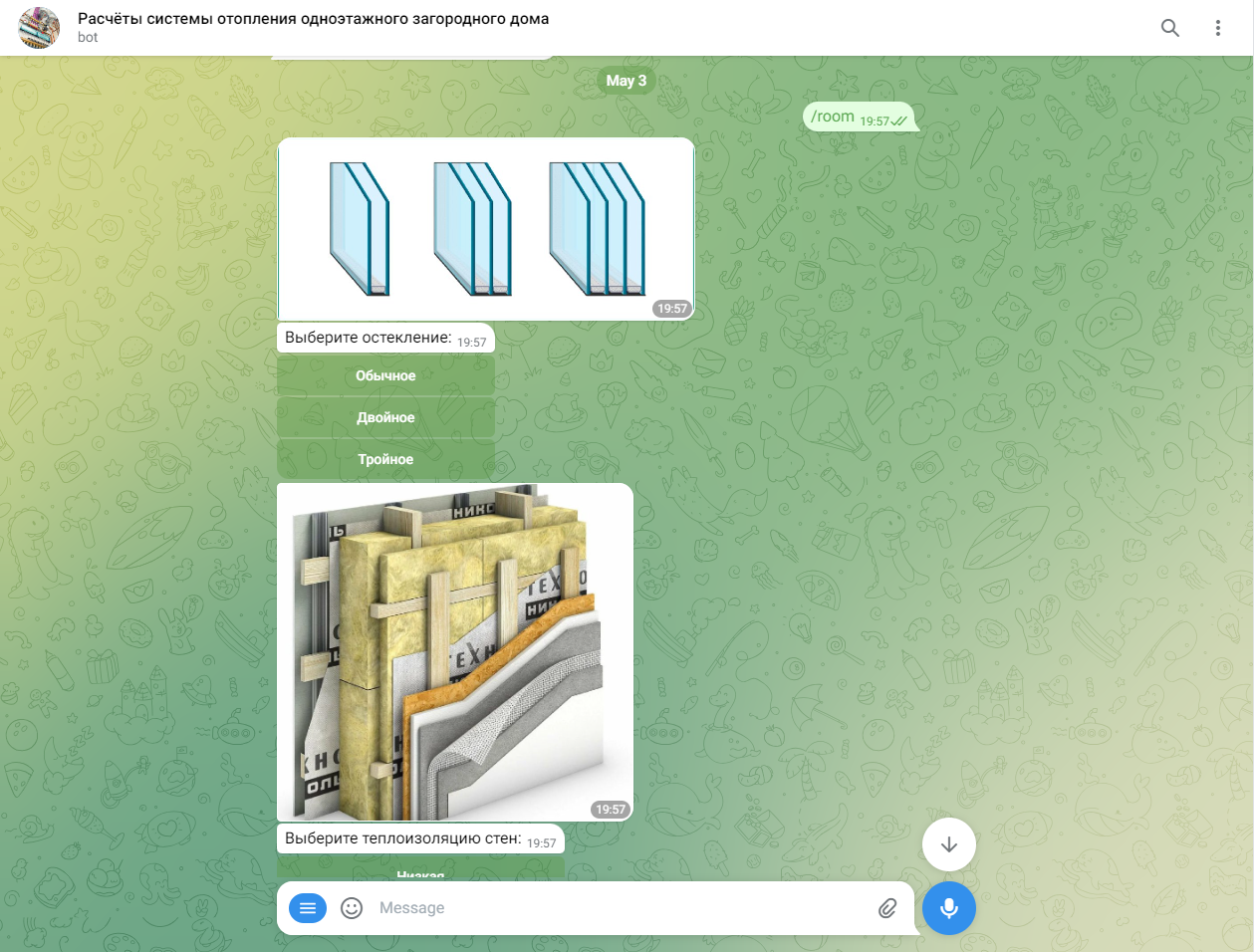
****

Рис. 1 – веб-версия приложения

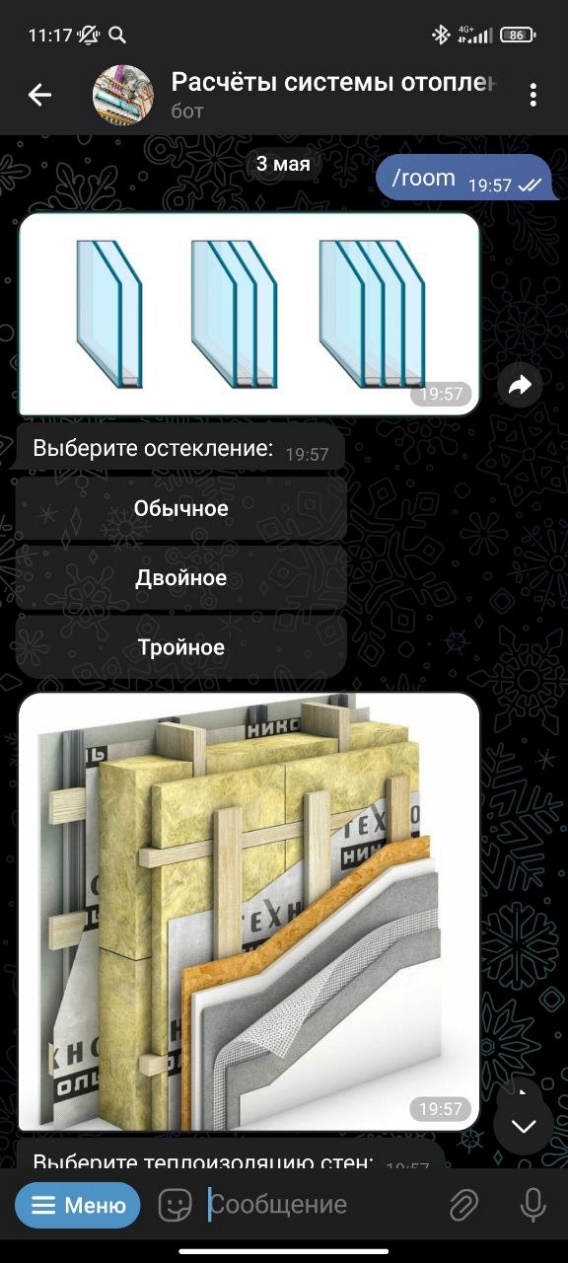


Рис. 3 – меню с командами

Рис. 2 – мобильная версия приложения

**Структура программы**

Программа разделена на 5 файлов, это сделано для удобства и более лёгкой читаемости кода.

Первый файл (boy.py) является основой программы, как раз его мы и запускаем, для полной работы, в нём соединяются функции из других файлов:

1. **info(message)** – функция, срабатывающая при запуске бота, а так же команд info и start, выводит список доступных команд
2. **house(message)** – функция, срабатывающая при отправленной команде house боту
3. House\_(bot) – функция, импортированная из другого файла, включающая в себя расчёт суммарных теплопотерь, мощности котла, скорости теплоносителя и полного оборота всего теплоносителя в системе (описание функций во втором файле)
4. **room(message)** – функция, срабатывающая при отправленной команде room боту
5. Room\_(bot) – функция, импортированная из другого файла, включающая в себя расчёт количества секций радиатора, необходимого для отопления комнаты (описание функций в третьем файле)

Перед тем как рассказать про второй и третий файлы, расскажу про четвёртый и пятый, так как они фигурируют в них.

Четвёртый файл (oop.py) является объектно-ориентированной частью программы и включает в себя три класса:

1. **House** – класс для дома, включающий в себя только одну подфункцию:
2. \_\_init\_\_(self) – функция, отвечающая за хранение переменных, относящихся к дому и расчётов для него
3. **House\_multip** – класс, состоящий из функций, для различных расчётов:
4. Square(self, len, wide) – функция, рассчитывающая площадь
5. Square\_wall(self, len, wide, high, s\_win, s\_d) – функция, рассчитывающая площадь стен с вычетом площадей окон и дверей
6. Volume(self, high, s) – функция, рассчитывающая объём дома
7. Heat\_loss(self, k, s, thick) – функция, рассчитывающая теплопотери отдельного элемента
8. Power(self, s, Q) – функция, рассчитывающая мощность котла
9. **Room** – класс для комнаты, включающий в себя только одну подфункцию:
10. \_\_init\_\_(self) – функция, отвечающая за хранение переменных, относящихся к комнате и расчётов для неё

Пятый файл(check.py) содержит в себе код проверки:

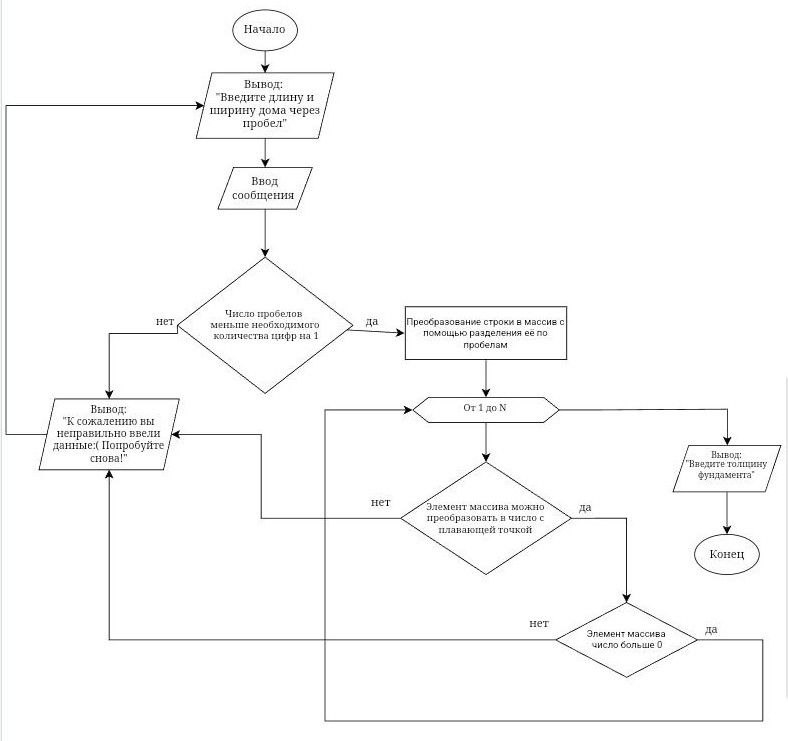
1. **is\_num(n)** – функция, проверяющая является ли числом вводимая строка (через ValueError)
2. **is\_nvalue(str, n) –** функция, проверяющая подходит ли данная строка для дальнейшей работы с ней(положительные числа, правильное количество вводимых чисел, числа ли вообще)

Второй файл (house.py) содержит в себе функцию по расчёту мощности котла, суммарных теплопотерь, количество теплоносителя и его полный оборот в системе:

1. **House\_(bot)** – функция, состоящая из множества подфункций:
2. house(message) – функция, после проверки входных данных, записывает параметры дома в класс House()
3. floor(message) – функция, после проверки входных данных, записывает параметры пола в класс House()
4. floor\_mat(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент теплопроводности пола в класс House()
5. wall(message) – функция, после проверки входных данных, записывает параметры стен в класс House()
6. wall\_mat(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент теплопроводности стены в класс House()
7. roof(message) – функция, после проверки входных данных, записывает параметры крыши в класс House()
8. roof\_mat(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент теплопроводности крыши в класс House()
9. window(message) – функция, после проверки входных данных, записывает параметры окон в класс House()
10. door(message) – функция, после проверки входных данных, записывает параметры дверей в класс House()
11. door\_mat(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент теплопроводности двери в класс House(), а так же в ей производятся окончательные расчёты мощности котла, количества теплоносителя и его полный оборот в системе, записываются результаты подсчёта так же в класс House()

Третий файл (room.py) аналогично второму содержит функцию, для расчётов количества секций радиатора, необходимых для отопления комнаты:

1. **Room\_(bot) –** функция, состоящая из множества подфункций:
2. window\_mat(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент, зависящий от остекления окон и необходимый для дальнейших расчётов в класс Room()
3. isolation(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент, зависящий от теплоизоляции стен и необходимый для дальнейших расчётов в класс Room()
4. window(message) – функция, после проверки входных данных, записывает параметры окон в класс House()
5. room(message) – функция, после проверки входных данных, записывает параметры комнаты в класс Room()
6. temperature(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент, зависящий от средней температуры зимой и необходимый для дальнейших расчётов в класс Room()
7. wall\_outside(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент, зависящий от количества стен в комнате, выходящих наружу, и необходимый для дальнейших расчётов в класс Room()
8. hight(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент, зависящий от высоты потолка в комнате и необходимый для дальнейших расчётов в класс Room()
9. radiator(c) – функция, использующая информацию о нажатой пользователем кнопке, записывает коэффициент, зависящий от материала радиатора и необходимый для дальнейших расчётов в класс Room(), и производит окончательные расчёты количества секций радиатора, необходимого для отопления данной комнаты, результат так же дублируется в класс Room()

**Описание алгоритма**

Данная часть кода, не считая первого вывода предложения и последнего, используется несколько раз в коде, для проверки вводимых данных на соответствие формату, для их дальнейшей обработки.

**Заключение**

В ходе курсовой работы был разработан Telegram Bot для выполнения расчётов системы отопления одноэтажного загородного дома на основе языка программирования Python с использованием библиотеки telebot, а именно расчёты суммарных теплопотерь, мощности котла, количество теплоносителя в системе и его полный оборот, а также количество секций радиатора, необходимого для отопления комнаты.

Программа реализует ввод необходимых данных для расчётов и проверку их на соответствие формату для дальнейшей обработки.

Программа может быть усовершенствована и преобразована в отдельное приложение с графическим построением дома с любым количеством этажей и определённой площадью, предназначенной для отопления.

**Библиографический список**

1. Любанович Б.Простой Python. Современный стиль программирования. 2-е изд. — СПб.: Питер, 2021 — 592 с.: ил. — (Серия «Бестселлеры O’Reilly»).
2. Всё о PyCharm: https://www.jetbrains.com/ru-ru/pycharm/
3. Шабаршин А. А. Элементарный TELEGRAM. Все, что нужно знать о самом перспективном мессенджере страны, чтобы на нем зарабатывать. — Москва: Эксмо, 2023 — 224 с.: ил. — (Маркетинг для немаркетологов).
4. Документация Telebot: https://pytba.readthedocs.io/ru/latest/index.html