ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ «ОДИНЦОВСКИЙ «ДЕСЯТЫЙ ЛИЦЕЙ»

(143000, Московская область, г. Одинцово, ул. Любы Новоселовой бул., 8)

КОНКУРСНАЯ РАБОТА

(информатика)

«Планировка квартир при помощи ИИ»

(проект)

Выполнил:

Алхимов Арсений Александрович, 10 класс

Московская область,

г. Одинцово,

ул. Белорусская, д.28, кв.169

Научный руководитель:

Пименова Ольга Рушановна,

учитель информатики и ИКТ

Одинцово

2022

**Оглавление**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Паспорт проекта** | | | | | | |  | | | | | | | | | 3 | | |
| **Введение** | |  | | | | | | | | | | | | | | 4 | | |
| **Основная часть** | | | | | |  | | | | | | | | | | 5 | | |
| Теоретическая часть | | | | | | | |  | | | | | | | | 5 | | |
| Планировка квартиры | | | | | | | | | |  | | | | | | 5 | | |
| Платформа Google Colab | | | | | | | | | | | |  | | | | 5 | | |
| Язык программирования Python | | | | | | | | | | | | | |  | | 6 | | |
| Библиотеки | | | | | | | | |  | | | | | | | 6 | | |
| Библиотека Random | | | | | | | |  | | | | | | | | 6 | | |
| Библиотека Copy | | | | | | | . | | | | | | | | | 7 | | |
| Библиотека Pprint | | | | | | |  | | | | | | | | | 7 | | |
| Библиотека PIL | | | | |  | | | | | | | | | | | 8 | | |
| Библиотека Requests | | | | | | | |  | | | | | | | | 8 | | |
| Библиотека ipywidgets | | | | | | | | | |  | | | | | | 9 | | |
| Библиотека JSON | | | | | | |  | | | | | | | | | 9 | | |
| Генетический алгоритм | | | | | | | | | | |  | | | | | 9 | | |
| База данных в Notion | | | | | | | | |  | | | | | | | 10 | | |
| **Практическая часть** | | | | | | | | |  | | | | | | | 11 | | |
| Написание кода | | | | |  | | | | | | | | | | | 11 | | |
| Загрузка базы планировок | | | | | | | | | | | | |  | | | 11 | | |
| Функция рассчёта этажа | | | | | | | | | | |  | | | | | 11 | | |
| Ввод параметров и запуск рассчёта | | | | | | | | | | | | | | |  | 11 | | |
| Пуск |  | | | | | | | | | | | | | | | 11 | | |
| **Заключение** | | |  | | | | | | | | | | | | | 12 | | |
| **Список использованной литературы** | | | | | | | | | | | | | | | | |  | 13 | |
| **Приложение** | | | |  | | | | | | | | | | | | 14 | | |

**Паспорт проекта**

**Название проекта** – «Планировка квартир при помощи ИИ»

**Автор проекта** – Алхимов Арсений Александрович

**Научный руководитель** – Пименова Ольга Рушановна

**Цель проектной работы** – создать программу для планировки квартир при помощи искусственного интеллекта

**Задачи проекта:**

* Собрать информацию из различных источников по данной теме
* Проанализировать основные онлайн платформы, представленные в среде Интернет и выбрать наиболее эффективный
* Проанализировать основные виды ИИ и выбрать наиболее эффективный
* Составить программу для планировки квартир
* Защитить проект

**Результат проекта** – программа для планировки квартир

**Этапы проектной работы:**

1. Подготовительный

1.1. Поиск литературы по данной теме

1.2. Выбор платформы и языка программирования для создания проекта

1. Исследовательский

2.1. Анализ литературы и интернет ресурсов

1. Аналитический

3.1. Обобщение прочитанного материала

3.2. Выборка самого необходимого из уже обобщенного материала

1. Практический

4.1. Написание кода на платформе

4.2. Загрузка базы данных

**Введение**

В наше время на планете проживает 8 млрд людей, и большая часть из них живет в многоэтажных домах. Сейчас такие дома строятся в больших количествах и для каждой отдельной квартиры нужен свой дизайн и планировка. У некоторых людей нет времени или средств на консультации с дизайнерами. Если бы любой человек мог за пару действий получить планировку своей будущей квартиры, то тогда жизнь стала намного проще, а дизайн не казался бы такой проблемой.

Поэтому я решил помочь человечеству с такой трудной проблемой как планировка квартиры.

**Актуальность**

Программа для планировки квартир обеспечит надежные и эффективные решения для дизайна квартиры, а также ускорит данный процесс и покажет самую оптимальную планировку.

**Цель**

Создать программу для планировки квартир при помощи искусственного интеллекта на онлайн платформе.

**Задачи**

* Собрать информацию из различных источников по данной теме
* Проанализировать основные онлайн платформы, представленные в среде Интернет и выбрать наиболее эффективный
* Проанализировать основные виды ИИ и выбрать наиболее эффективный
* Составить программу для планировки квартир
* Защитить проект

**Практическая значимость**

Проекта заключается в том, что мои наработки могут использоваться в повседневной жизни, чтобы навсегда решить проблему планировки квартир.

**Основная часть**

**Теоретическая часть**

**Планировка квартиры**

Планировка квартиры — это конфигурация ее внутреннего пространства: габариты помещений, их расположение относительно друг друга, высота потолков, форма стен, размер и направление окон. Правильно выбрать планировку не менее важно, чем не ошибиться с районом или площадью жилья.

Общий метраж квартиры часто служит главной характеристикой при выборе жилья. Но не менее важны размеры и пропорции каждого помещения. К примеру, дискомфорт может вызывать квартира с большой гостиной или спальней, но с крошечной ванной или тесной прихожей.

Если в новое жилье планируется перевезти всю имеющуюся мебель, стоит тщательно проверить, как предметы впишутся в габариты нового жилья — в первую очередь речь идет о диванах и массивных шкафах.

**Платформа Google Colab**

Colab используют, чтобы создавать код на Python прямо в браузере. Google Colaboratory — бесплатная среда, чтобы писать код в jupyter notebook. Она функционирует по принципу облака, поэтому над одним проектом могут работать одновременно несколько человек. Программа предоставляет доступ к графическим процессорам GPU и TPU. Благодаря их мощности можно исследовать искусственный интеллект и развивать приложения на основе нейросетей. Сервис бесплатный. Единственное ограничение — через 12 часов данные удаляются.

Colab позволяет использовать в одном файле исполняемый код, html-разметку, картинки. Всё будет храниться на гугл-диске. Этими файлами можно делиться: разрешать просматривать, редактировать и оставлять комментарии для совместной работы.

Я выбрал Google Colab, т. к. она лучше всего подойдет для моих целей.

**Язык программирования Python**

Python — это скриптовый язык программирования. Он универсален, поэтому подходит для решения разнообразных задач и многих платформ, начиная с iOS и Android и заканчивая серверными ОС. Это интерпретируемый язык — он не компилируется, то есть до запуска представляет из себя обычный текстовый файл. Программировать можно практически на всех платформах, язык хорошо спроектирован и логичен.

Я выбрал данный язык программирования, потому что знаю его лучше всего и потому что его поддерживает Google Colab.

**Библиотеки**

В данном проекте я буду использовать несколько библиотек, а именно: random, сору, pprint, PIL, requests, json и ipywidgets. Они намного упростят процесс разработки и добавят функции, без которых я бы не смог реализовать большинство удобств и преимуществ своей программы. Я бы хотел рассказать кратко о каждой из них.

**Библиотека Random**

Этот модуль реализует генераторы псевдослучайных чисел для различных распределений.

Для целых чисел существует равномерный выбор из диапазона. Для последовательностей существует равномерный выбор случайного элемента, функция для генерации случайного перестановка списка на месте и функция случайной выборки без замены.

На вещественной линии существуют функции для вычисления равномерного, нормального (гауссовского), логнормальные, отрицательные экспоненциальные, гамма- и бета-распределения. Для генерации распределения углов, распределение фон Мизеса доступно.

Практически все функции модуля зависят от базовой функции random(),которая генерирует случайный поплавок равномерно в полуоткрытом диапазоне. Python использует Mersenne Twister в качестве генератора ядра. Обеспечивает 53-битную точность плавает и имеет период 2\*\*19937-1. Базовая реализация в C как быстрый, так и потокобезопасный. Mersenne Twister является одним из самых обширных протестированные генераторы случайных чисел существуют. Однако, будучи полностью детерминированный, он подходит не для всех целей, и совершенно непригоден для криптографических целей.

**Библиотека Copy**

Операторы присваивания в Python не копируют объекты, они создают привязки между целевым объектом и объектом. Для коллекций, которые являются изменяемыми или содержат изменяемые элементы, иногда требуется копия, чтобы можно было изменить одну копию без изменение другого. Этот модуль обеспечивает общее поверхностное и глубокое копирование операции.

Краткое описание интерфейса:

copy.copy(x)

Возврат неглубокой копии x.

copy.deepcopy(x[, memo])

Возврат глубокой копии x.

exception copy.Error

Вызывается для ошибок, специфичных для модуля.

**Библиотека Pprint**

Модуль pprint предоставляет возможность произвольной «красивой печати» cтруктур данных Python в форме, которая может быть использована в качестве входных данных для интерпретатора. Если форматированные структуры включают объекты, которые не являются фундаментальными Python типы, представление может быть не загружаемым. Это может иметь место, если объекты такие как файлы, сокеты или классы включены, а также многие другие объекты, которые не могут быть представлены в виде литералов Python.

Отформатированное представление сохраняет объекты в одной строке, если это возможно, и разбивает их на несколько строк, если они не вписываются в допустимую ширину. Создайте объекты PrettyPrinter явным образом, если вам нужно настроить ограничение ширины.

Словари сортируются по ключам перед вычислением отображения.

**Библиотека PIL**

PIL — это библиотека изображений Python Фредрика Лунда и участников.

Библиотека изображений Python добавляет возможности обработки изображений в интерпретатор Python.

Эта библиотека обеспечивает обширную поддержку форматов файлов, эффективное внутреннее представление и довольно мощные возможности обработки изображений.

Основная библиотека изображений предназначена для быстрого доступа к данным, хранящимся в нескольких основных форматах пикселей. Это должно обеспечить прочную основу для общего инструмента обработки изображений.

**Библиотека Requests**

Requests — это элегантная и простая HTTP-библиотека для Python, созданная для людей.

Requests позволяют очень легко отправлять запросы HTTP/1.1. Нет необходимости вручную добавлять строки запроса в URL-адреса или для кодирования данных POST в форме. Поддержание активности и пул HTTP-соединений на 100% автоматические, благодаря urllib3.

**Библиотека ipywidgets**

ipywidgets, также известные как jupyter-widgets или просто виджеты, являются интерактивными HTML-виджетами для записных книжек Jupyter и ядра IPython.

Надписи оживают при использовании интерактивных виджетов. Пользователи получают контроль над их данными и могут визуализировать изменения в данных.

Основные виджеты, предоставляемые этой библиотекой, называются core interactive widgets. Демонстрационная записная книжка содержит обзор основных интерактивных виджетов, в том числе:

* ползунки
* индикаторы выполнения
* текстовые поля
* переключатели кнопок и флажков
* области отображения
* и многое другое

**Библиотека JSON**

JSON (JavaScript Object Notation) — это упрощенный формат обмена данными, вдохновленный синтаксисом литералов объектов JavaScript.

json предоставляет API, знакомый пользователям стандартных модулей marshal и pickle библиотеки.

**Генетический алгоритм**

Генетический алгоритм — это эвристический алгоритм поиска, используемый для решения задач оптимизации и моделирования путём последовательного подбора, комбинирования и вариации искомых параметров с использованием механизмов, напоминающих биологическую эволюцию. Является разновидностью эволюционных вычислений. Отличительной особенностью генетического алгоритма является акцент на использование оператора «скрещивания», который производит операцию рекомбинации решений-кандидатов, роль которой аналогична роли скрещивания в живой природе.

Данный алгоритм я буду использовать для написания искусственного интеллекта в своей программе. Он поможет улучшить результаты по окончанию программы и подобрать наиболее подходящий вариант планировки.

**База данных в Notion**

Notion — приложение, которое предоставляет такие компоненты, как базы данных, доски канбан, вики, календари и напоминания. Пользователи могут подключать эти компоненты для создания собственных систем управления знаниями, ведения заметок, управления данными, управления проектами и другими. Эти компоненты и системы могут использоваться индивидуально или совместно с другими.

Я решил создать базу данных в Notion, т. к. в ней по моему мнению удобнее всего создать базу данных для моей программы. В моей базе данных конкретно будут характеристики разнообразных квартир и их планировки. Это сделано для того, чтобы искусственный интеллект смог подобрать несколько наиболее подходящих вариантов для введённых характеристик(рис.1).

**Практическая часть**

**Написание кода**

Для облегчения работы и понимания кода я разделил программу на четыре части, а именно: Загрузка базы планировок, Функция рассчёта этажа, Ввод параметров и запуск рассчёта, Пуск.

**Загрузка базы планировок**

В данном фрагменте кода происходит загрузка и раскодировка данных с базы планировок для того, чтобы при активации финального фрагмента кода программа смогла подобрать планировки квартир(рис.2).

**Функция рассчёта этажа**

В этом отрезке кода описана функция рассчёта этажа при помощи генетического алгоритма. Грубо говоря это и есть “мозг” программы, из-за которого она и может выбрать самые подходящие решения для планировки квартиры.

**Ввод параметров и запуск рассчёта**

В данном отрывке кода прописан интерфейс для финального кусочка программы, чтобы данные было удобнее вводить.

**Пуск**

Это финальный фрагмент кода. В нём выводится интерфейс и вводятся данные квартиры. В результате заданного количества поколений программа выдает данные, с которыми сравниваются данные из базы планировок и выбираются наиболее подходящие изображения.

**Заключение**

Можно с уверенностью сказать, что основные цели и задачи моей работы были достигнуты. В ходе своей работы я создал программу для планировки квартир, которая может помочь многим людям с выбором дизайна. Так же я изучил новые для себя библиотеки и набрался опыта в этой отрасли. Я получил много теоретических и практических знаний, которые смогу применить в работе над дальнейшими проектами.

**Список использованной литературы**

1. Официальный сайт Python - <https://www.python.org/doc/>
2. Официальный сайт Google Colab - <https://colab.research.google.com/>
3. Документация к библиотеке Random - <https://docs.python.org/3/library/random.html>
4. Документация к библиотеке Copy - <https://docs.python.org/3/library/copy.html>
5. Документация к библиотеке Pprint - <https://docs.python.org/3/library/pprint.html>
6. Документация к библиотеке PIL - <https://pillow.readthedocs.io/en/stable/>
7. Документация к библиотеке Requests - <https://pypi.org/project/requests/>
8. Документация к библиотеке JSON - <https://docs.python.org/3/library/json.html>
9. Документация к библиотеке ipywidgets - <https://github.com/jupyter-widgets/ipywidgets#readme>
10. Генетический алгоритм - <https://ru.wikipedia.org/wiki/%D0%93%D0%B5%D0%BD%D0%B5%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC>
11. Платформа Notion - <https://ru.wikipedia.org/wiki/Notion_(%D0%BF%D1%80%D0%B8%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5)>

**Приложение**

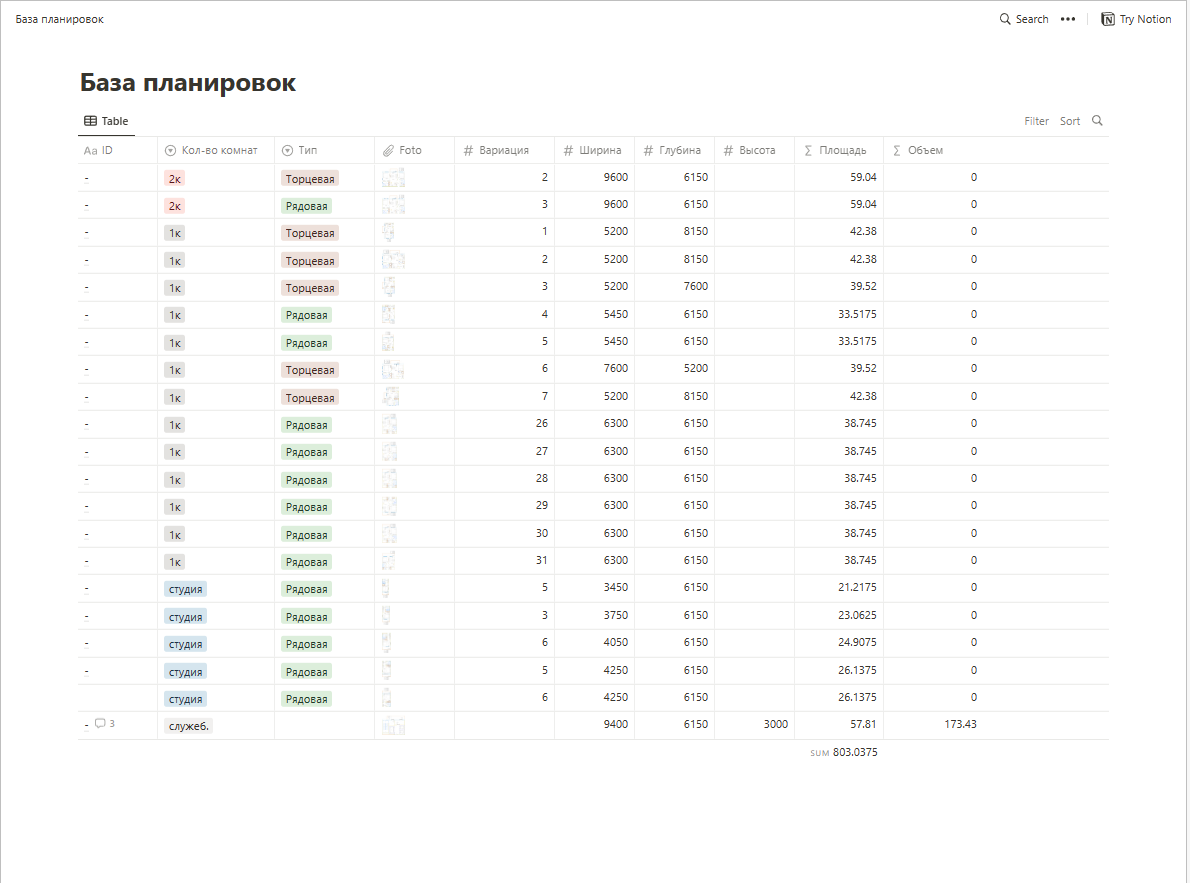
Рис.1 

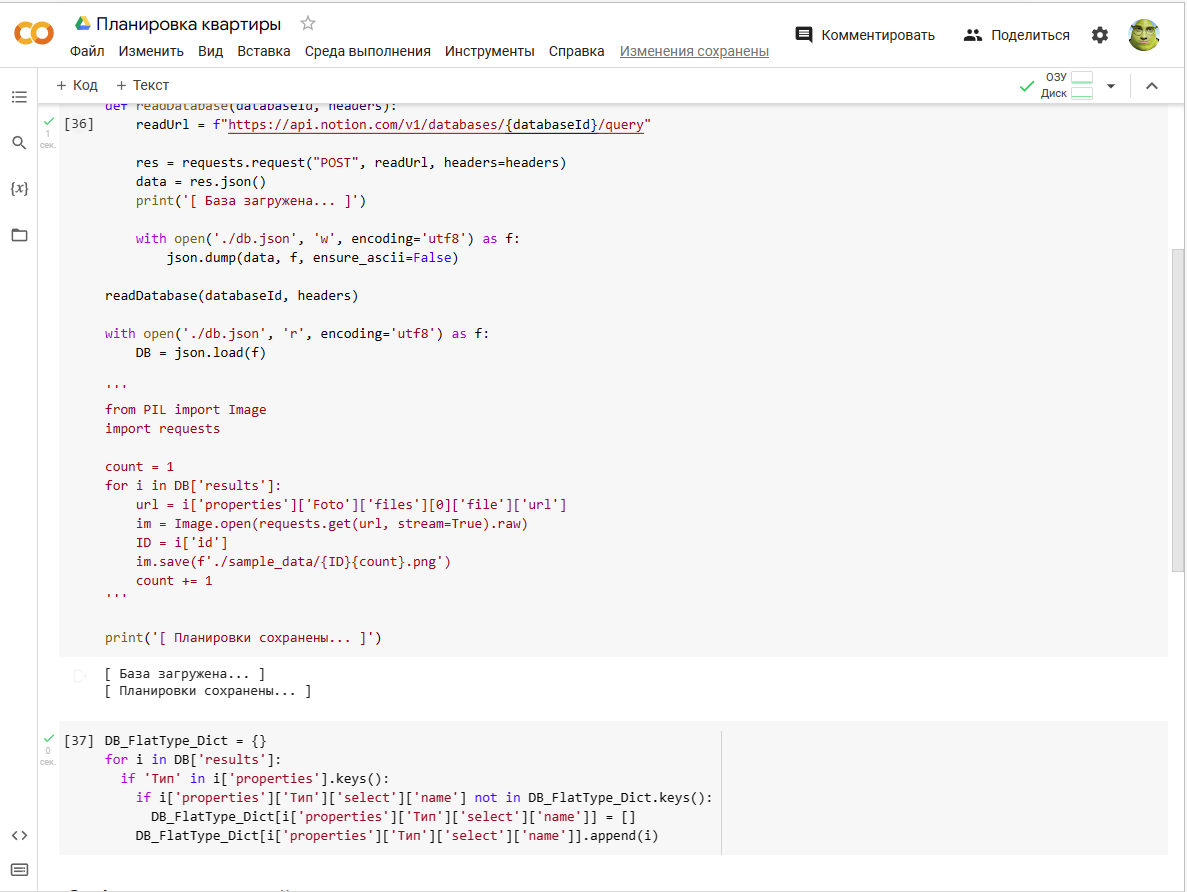
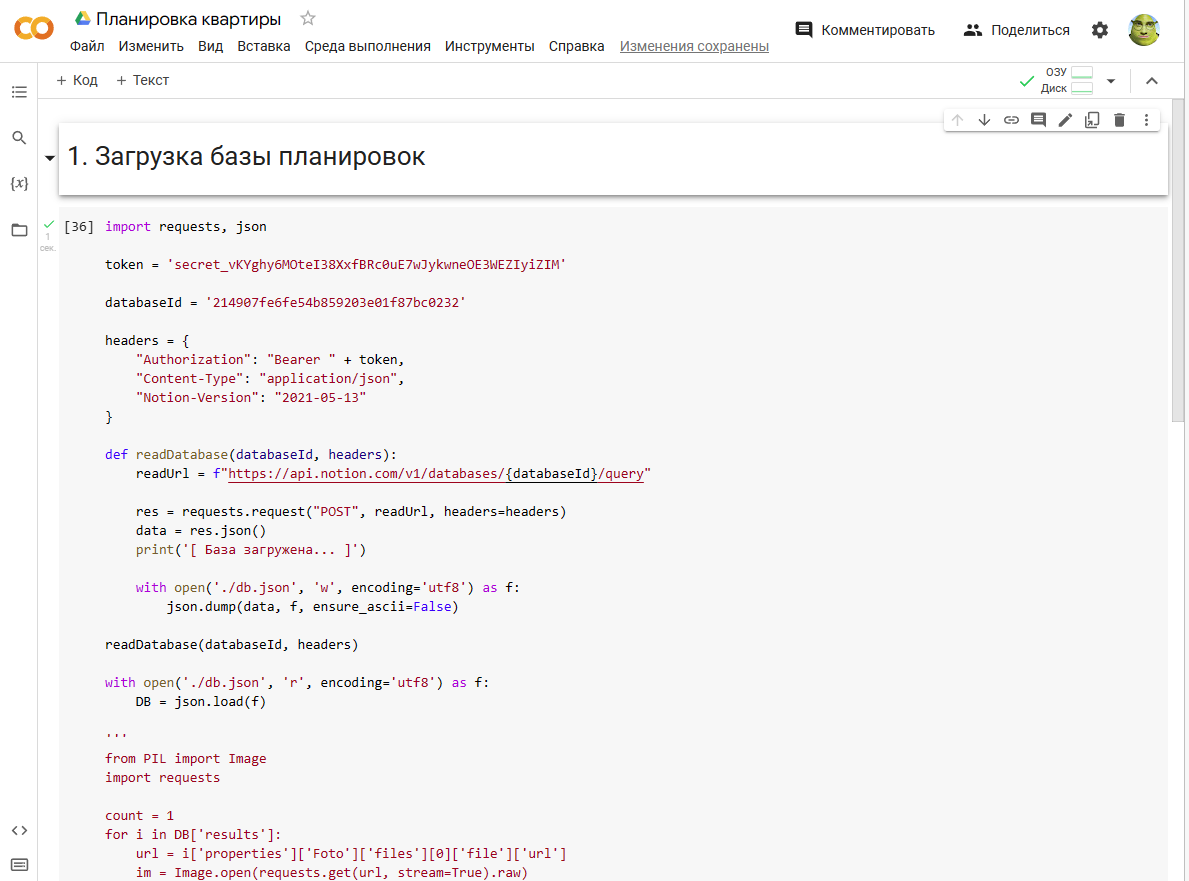
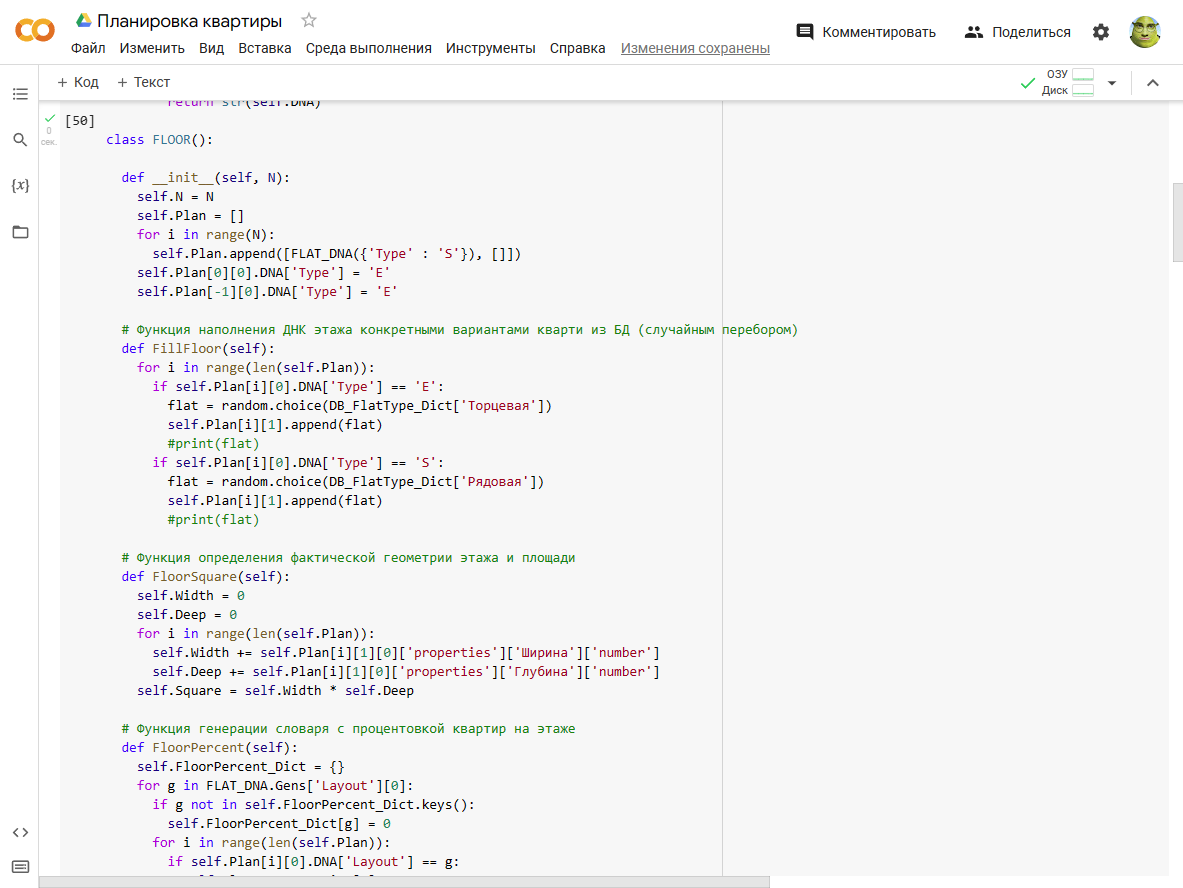
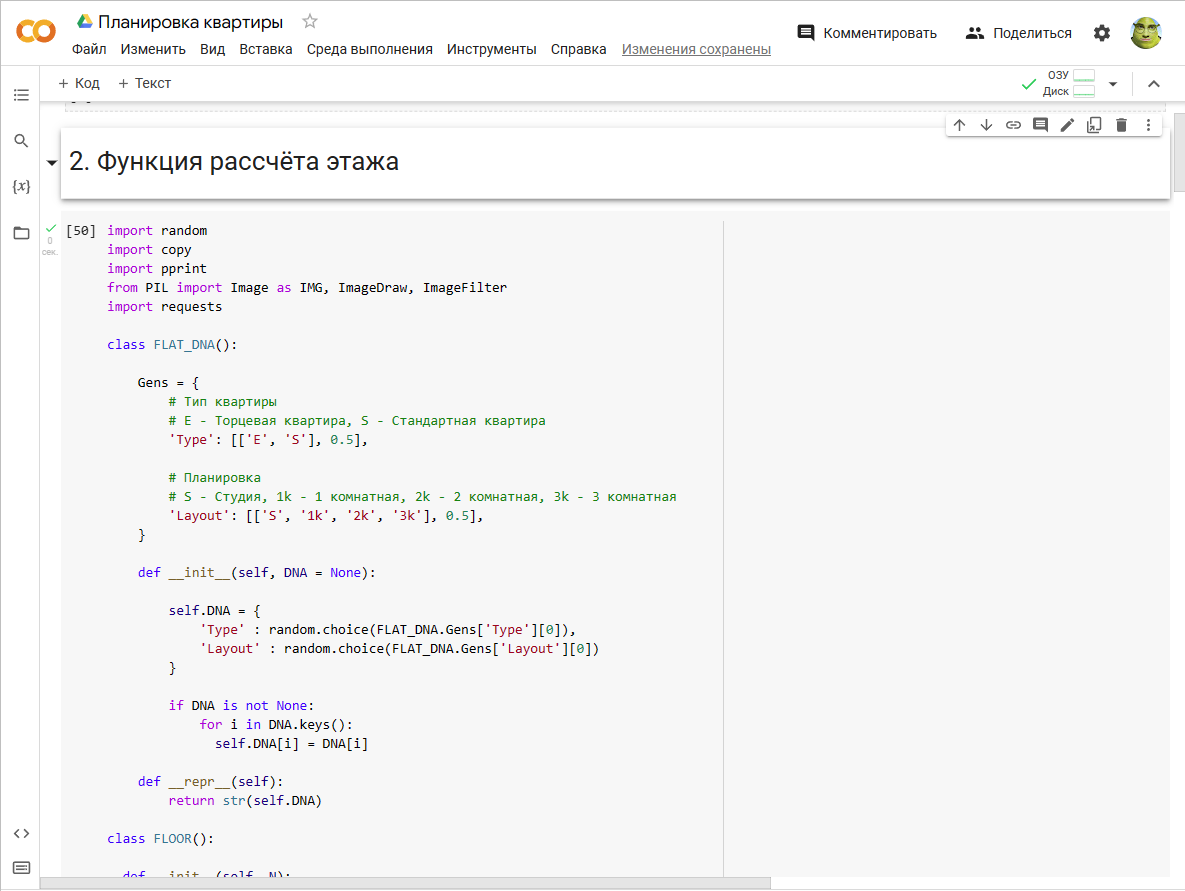
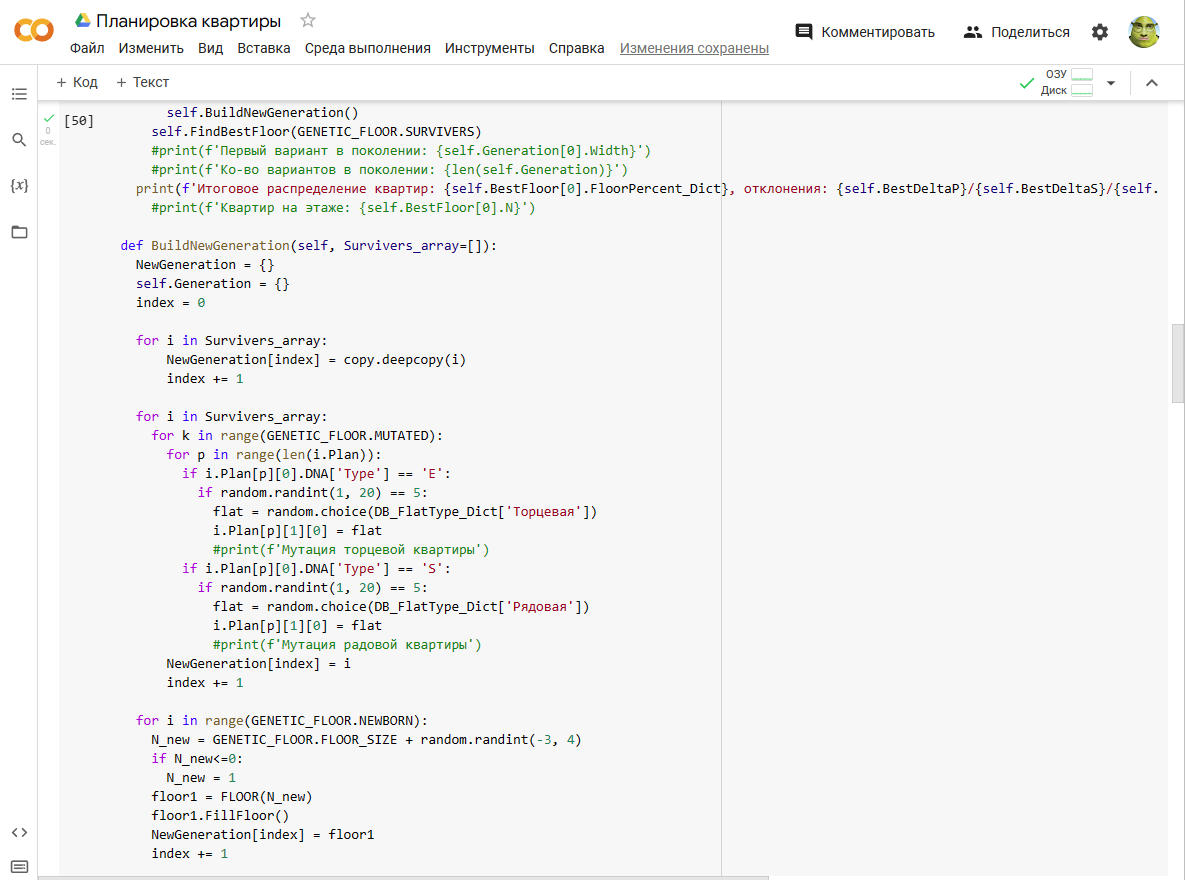
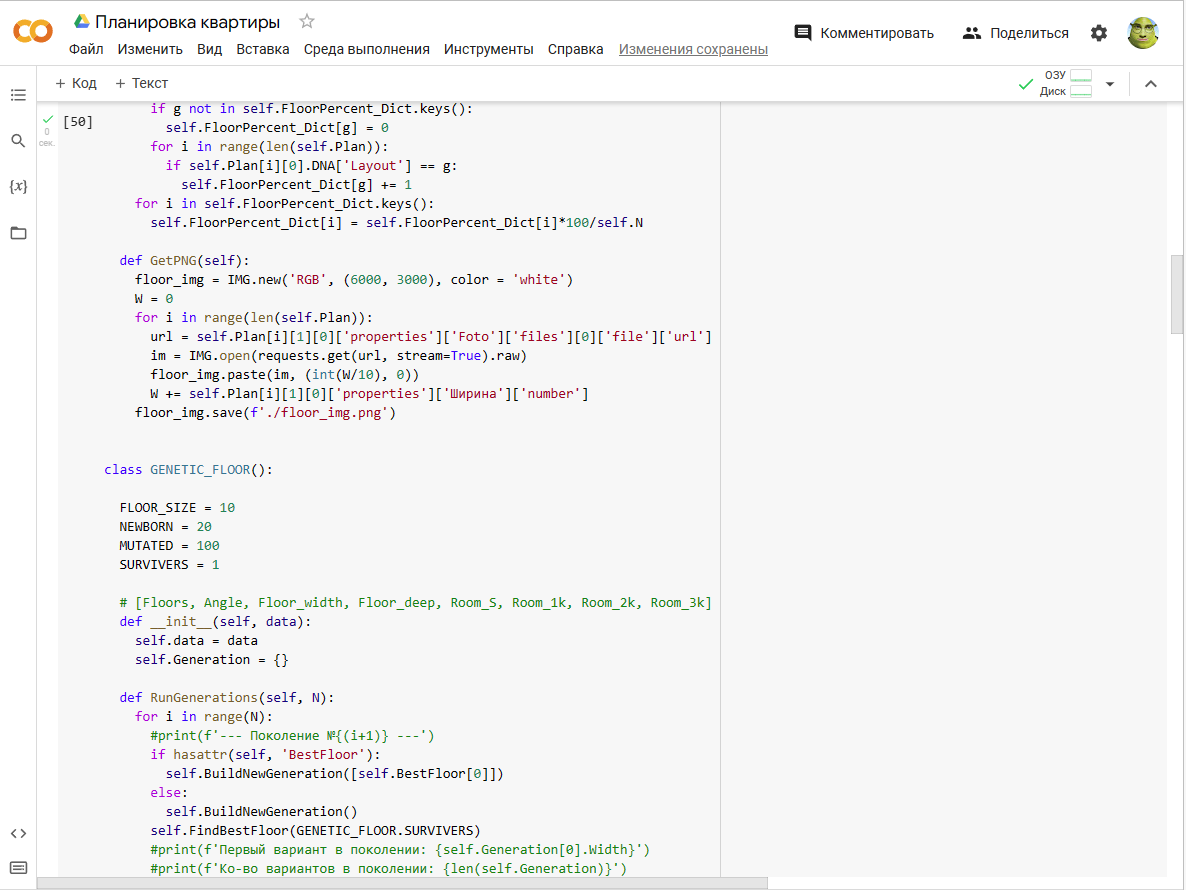
Рис.2

Рис.3



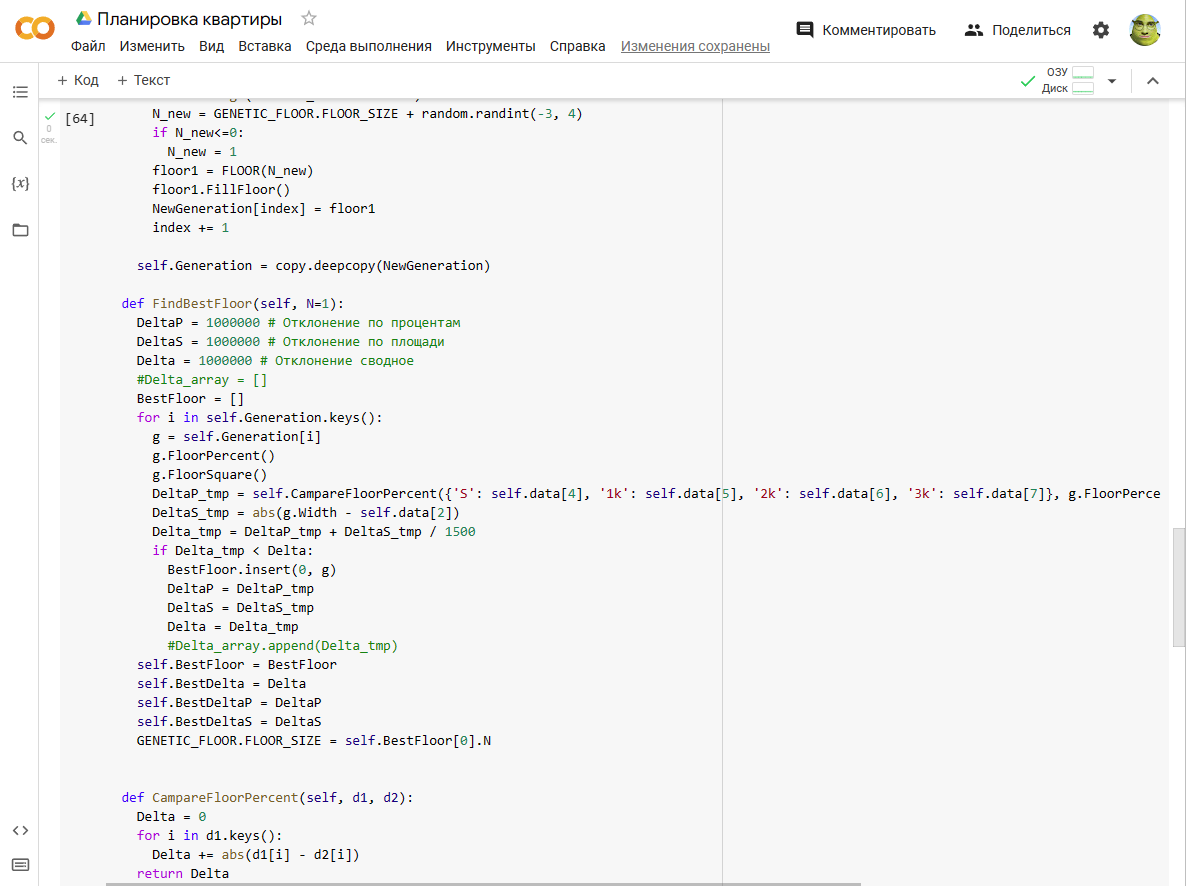
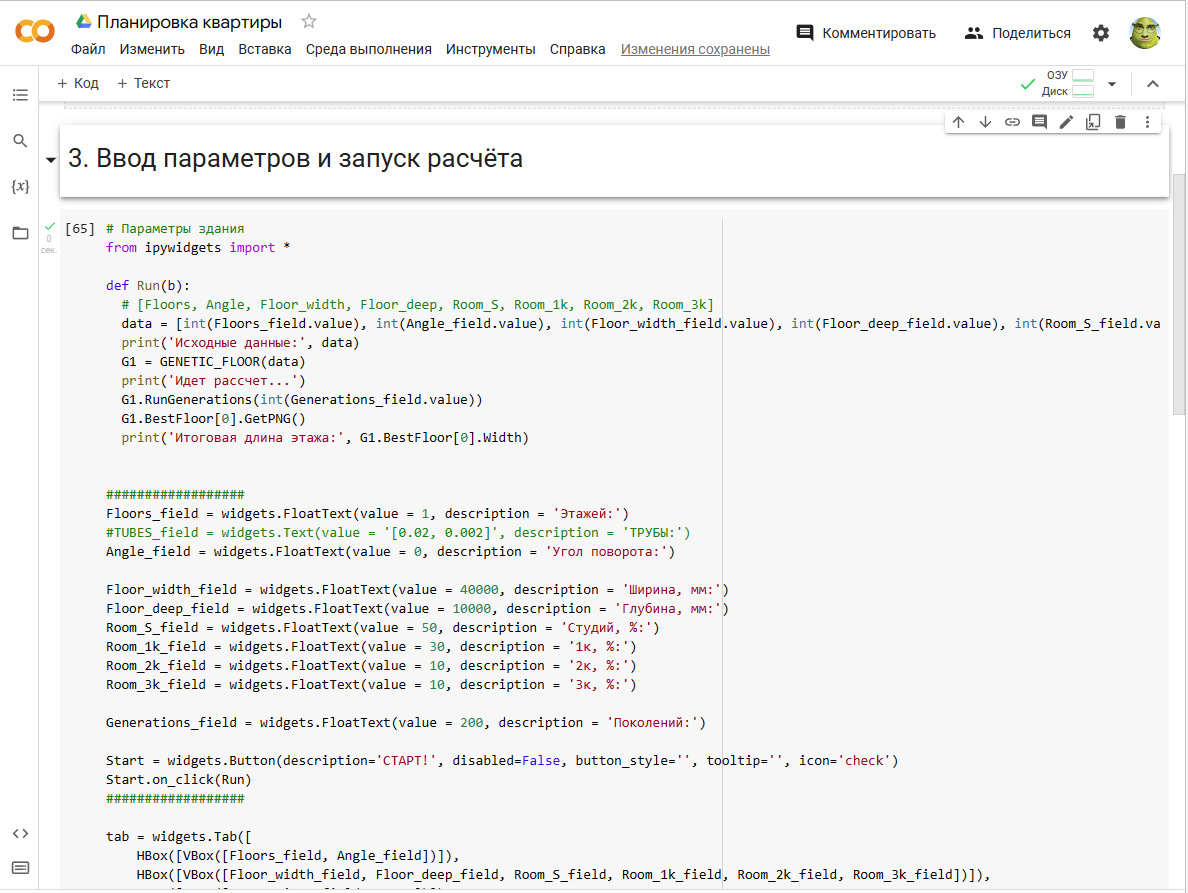


Рис.4

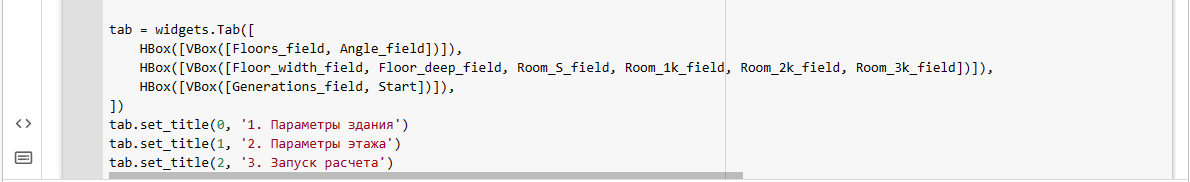


Рис.5