Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Брянский государственный инженерно-технологический университет»

Кафедра «Информационные технологии»

**Курсовая работа**

по дисциплине «Язык программирования Python»

**Создание программы по расчёту бетонных и железобетонных конструкций на языке Python**

КР – 02068025-09.03.01-018.20

Студент \_\_\_\_\_\_\_\_\_ Шатунов Дмитрий Александрович

Группа ИВТ-201 № зачётной книжки: 20-2.018

Руководитель работы \_\_\_\_\_\_\_\_ к.т.н., доц. Моргунов М.В.

Нормоконтроль \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ к.т.н., доц. Моргунов М.В.

Доступ к защите «\_»\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ к.т.н., доц. Моргунов М.В.

Дата защиты «\_»\_\_\_\_\_\_\_ Оценка \_\_\_\_\_\_\_\_

Члены комиссии: \_\_\_\_\_\_\_\_ к.т.н., доц. Моргунов М.В.

\_\_\_\_\_\_\_\_ к.э.н., доц. Казаков О.Д.

\_\_\_\_\_\_\_\_ к.э.н., доц.Юркова О.Н.

Брянск 2021

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc91071704)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 5](#_Toc91071705)

[1.1 Язык программирования Python. 5](#_Toc91071706)

[1.2 PyQT 6](#_Toc91071707)

[1.3 Расчет железобетонных конструкций без предварительно напряженной арматуры 8](#_Toc91071708)

[2. рассчет прямоугольных колон 12](#_Toc91071709)

[2.1 Тестовые данные 12](#_Toc91071710)

[2.2. Разработка алгоритма программы 17](#_Toc91071711)

[2.3Инструкция по эксплуатации программы 19](#_Toc91071712)

[3. ЗАКЛЮЧЕНИЕ 31](#_Toc91071713)

[4. СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 32](#_Toc91071714)

[**ПРИЛОЖЕНИЕ А** 33](#_Toc91071715)

# ВВЕДЕНИЕ

Python – сравнительно новый язык программирования, который был создан голландским программистом Гвидо ванн Россумом (GuidovanRossum) в начале 90-х годов. Ван Россум участвовал в разработкеязыка ABC, ориентированного на обучение программированию. Сейчас ванн Россум работает в компании Google, но половину своего времени уделяет дальнейшей разработке Python.

Python – популярный язык программирования, используемый как для разработки самостоятельных программ, так идля создания прикладных сценариев в самых разных областях применения. Это мощный, переносимый, простой в использовании и свободно распространяемый язык. Программисты, работающие в самых разных областях, считают, что ориентация Python на эффективность разработки и высокое качество программного обеспечения дает ему стратегическое преимущество как в маленьких, так и в крупных проектах.

В рамках курсовой работы необходимо разработать программу на определенную тематику согласно заданию курсовой работы.

Цели курсовой работы:

* разработка программы, которая будет проводить расчеты железобетонных конструкций без предварительно напряженной арматуры, а в частности: прямоугольные сечения с симметричной арматурой;
* создание графического интерфейса с помощью PyQT.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

## Язык программирования Python.

Определенный интерес представляет рассмотрение сравнительно нового языка программирования Python, который был создан его автором Гвидо ван Россумом в начале 90-х годов.

Отличительные характеристики языка:

* очень низкий порог вхождения, уже после одного дня изучения можно начать писать простые программы;
* минималистичный язык, с небольшим количеством конструкций;
* краткий код;
* прекрасно подходит для создания программ-обёрток, поддерживается импорт Си-библиотек;
* существует большое количество реализаций: CPython (основная реализация); Jython (реализация для JVM); IronPython (CLR); PyPy;
* очень хорошая поддержка математических вычислений (библиотеки NumPy, SciPy);
* используется для обработки естественных языков (NLTK);
* большое количество развитых web-фреймворков (Django, TurboGear, CherryPy, Flask).
* Язык можно рекомендовать всем, кто только начинает программировать, как первый язык программирования в жизни.
* В связи с этим, определенный интерес представляет рассмотрение сравнительно нового языка программирования Python, который был создан его автором Гвидо ван Россумом в начале 90-х годов.

## PyQT

PyQt – набор привязок графического фреймворкаQtдля языка программирования Python, выполненный в виде расширения Python. PyQtработает на всех платформах, поддерживаемых Qt: Linuxи другие UNIX-подобные OC, MacOSXи Windows. Существует 2 версии: PyQt5, поддерживающий Qt 5, и PyQt4, поддерживающий Qt4. PyQtпрактически полностью реализует возможности Qt. А это более 600 классов, более 6000 функций и методов, включая:

* cуществующий набор виджетов графического интерфейса;
* cтили виджетов;
* доступ к базам данных с помощью SQL;
* qScintilla, основанный на Scintillaвиджет текстового редактора;
* поддержку интернационализации (i18n);
* парсерXML;
* поддержку SVG;
* интеграцию с WebKit, движком рендеринга HTML;
* поддержку воспроизведения видео и аудио.

PyQtтакже включает в себя QtDesigner–дизайнер графического интерфейса пользователя. Программа pyuicгенерирует Pythonкод из файлов, созданных в QtDesigner. Это делает PyQtочень полезным инструментом для быстрого прототипирования. Кроме того, можно добавлять новые графические элементы управления, написанные на Python, в QtDesigner.

Обзор основных модулей:

* QtCore — основные не графические классы: система сигналов и слотов, платформонезависимые абстракции для Unicode, потоков, разделяемой памяти, регулярных выражений и т. д.
* QtGui — компоненты графического интерфейса (элементы управления), основанные на визуальном представлении.
* QtNetwork — классы для сетевого программирования. Например, клиентов и серверов через UDP и TCP.
* QtOpenGL — классы, позволяющие использовать OpenGL и 3D-графику в приложениях PyQt.
* QtScript — классы, позволяющие использовать встроенный в Qt интерпретатор JavaScript для управления приложением.
* QtSql — классы для интеграции с базами данных с помощью SQL.
* QtSvg — классы для отображения векторной графики в формате SVG.
* QtXml — классы, реализующие обработку XML.
* uic — реализация обработки XML-файлов, созданных в QtDesigner, для генерации из них Python-кода графического интерфейса

## Расчет железобетонных конструкций без предварительно напряженной арматуры

Учет влияния прогиба элементов

1. Расчетная длина *l*0принимается равной:

а) при вычислении коэффициента h*v*, а также при расчете элемента на действие продольной силы со случайным эксцентриситетом для элементов:

с шарнирным опиранием на двух концах - 1,0*l*;

с шарнирным опиранием на одном конце, а на другом конце:

с жесткой заделкой – 0,7*l*;

с податливой заделкой - 0,9*l*;

с заделкой на двух концах:

жесткой - 0,5*l*;

податливой – 0,8*l*;

с податливой заделкой на одном конце и с жесткой заделкой на другом - 0,7*l*;

б) при вычислении коэффициента h*h* для элементов:

с шарнирным опиранием на одном конце, а на другом конце

с жесткой заделкой - 1,5*l*;

с податливой заделкой - 2,0*l*;

с заделкой на двух концах:

жесткой - 0,8*l*;

податливой - 1,2*l*;

с податливой заделкой на одном конце и с жесткой заделкой на другом - *l*;

с жесткой заделкой на одном конце и незакрепленным другим концом (консоль) – 2*l.*

Здесь *l* - расстояние между концами элемента.

Для конкретных конструкций и сооружений можно принимать иные значения l0.

3.2.46 Проверку прочности прямоугольных сечений с несимметричной

арматурой производят из условия (3.91), определяя высоту сжатой зоны по формуле (3.100)

при этом если (3.2.4), высоту сжатой зоны корректируют, вычисляя по формуле(3.101)

3.91 . где М – момент относительно центра тяжести сечения, определяемый с учетом прогиба элементов согласно 3.2.40-3.2.42;

х – высота сжатой зоны, принимаемая равной.

а) при (рисунок 3.27) – х = ;

б) при

где определяется по формуле

Здесь ; – согласно 3.2.4.

3.2.45 Расчет по прочности прямоугольных сечений внецентренно сжатых элементов с арматурой, расположенной у противоположных в плоскостях изгиба сторон сечения, при эксцентриситете продольной силы , , при допускается производить из условия

*), (3.99)*

где φ – коэффициент, коэффициент, принимаемый при длительном действии нагрузки по таблице 3.5 в зависимости от гибкости элемента; при кратковременном действии нагрузки значения φ определяют по линейному закону, принимая φ = 0,9 при = 10 и φ = 0,85 при = 20.

# рассчет прямоугольных колон

# Тестовые данные

Тема данной курсовой работы: Расчет железобетонных конструкций без предварительно напряженной арматуры(прямоугольные сечения с симметричной арматурой) .

Пример 1:

Дано: поскольку колонна закрепленаколонна среднего этажа связевого каркаса с сечением размерами ; тяжелый бетон класса В25 (= 14,5МПа);продольная арматура класса А400( == 350 МПа); продольные силы и изгибающие моменты: от вертикальных нагрузок в опорном сечении: от всех нагрузок = 2200 кН, = 20кН .м, постоянных и длительных = 1980кН, = 0кН .м.; высота этажа H = 6 м.

Требуется определить площадь сечения продольной арматуры.

Расчет. Поскольку колонна закреплена с обоих концов шарнирно опертыми ригелями, принимаем согласно 3.4.42, а) расчётную длину равной = H = 6 м. Тогда = 6/0,4 = , т.е. учет прогиба колонны обязателен.

Эксцентриситет продольной силы от всех нагрузок равен

Поскольку h/30 = 400/30 = 13,3 > /600 = 6000/600 = 10мм, согласно 3.2.46 случайный эксцентриситет принимаем равным = 13,3 мм . Следовательно, расчет колонны производим на действие продольной силы с эксцентриситетом согласно 3.2.45.

Из таблицы 3.5 при длительном действии нагрузки при кратковременном действии нагрузки

Из условия (3.99) находим при кратковременном действии нагрузки:

Из условия (3.99) находим при длительном действии нагрузки:

Окончательно принимаем = 1018 (418).

Пример 2.

Дано: поскольку колонна закрепленаколонна среднего этажа связевого каркаса с сечением размерами ; тяжелый бетон класса В25 (= 14,5МПа);продольная арматура класса А400( == 350 МПа); продольные силы и изгибающие моменты: от вертикальных нагрузок в опорном сечении: от всех нагрузок = 100 кН, = 100 кН .м, постоянных и длительных = 100кН, = 100кН .м.; высота этажа H = 6 м.

Требуется проверить прочность опорного сечения колонны.

Таблица 2 Результаты вычислений

|  |  |
| --- | --- |
| Переменные | Посчитанные значения |
|  | 6 м |
| /h | 15мм |
|  | 1000 мм |
|  | 13мм |
| при длительном действии | 0,83 |
| при кратковременном действии | 0,875 |
| при кратковременном действии | -6302 |
| при длительном действии | -5721 |

Пример 3.

Дано: поскольку колонна закрепленаколонна среднего этажа связевого каркаса с сечением размерами ; тяжелый бетон класса В25 (= 14,5МПа);продольная арматура класса А400( == 350 МПа); продольные силы и изгибающие моменты: от вертикальных нагрузок в опорном сечении: от всех нагрузок = 9900кН, = 9900 кН .м, постоянных и длительных = 9900 кН, = 9900кН .м.; высота этажа H = 6 м.

Требуется проверить прочность опорного сечения колонны.

Таблица 3 Результаты вычислений

|  |  |
| --- | --- |
| Переменные | Посчитанные значения |
|  | 6 м |
| /h | 15мм |
|  | 1000 мм |
|  | 13мм |
| при длительном действии | 0,83 |
| при кратковременном действии | 0,875 |
| при кратковременном действии | 25698 |
| при длительном действии | 28113 |

# 2.2. Разработка алгоритма программы

В данной курсовой работе использовали различные методы програмирования, были отображены пользовательские функции описанные ниже.

В классе Calcфункция setting\_calc используется для того, чтобы дать имя окну и загрузить ui файл.

Функция returnres выводит результат вычисления в textedit и записывает его текстовый файл.

ВклассеHelpфункцияsetting\_helpиспользуется для того, чтобы дать имя окну и загрузить ui файл, указать работу кнопок.

Функции info\_about, limit\_mean, instruction\_use созданы для работы кнопок, при нажатии на которых выводится соответствующая информация.

В классе Asform функция setting\_asform используется для того, чтобы дать имя окну, загрузить ui файл.

В классе HistoryCalc функция settingHistoryCalc используется для того, чтобы дать имя окну, загрузить ui файл.

Функция concatStr читает из текстового файла и выводит информацию в окне История вычислений.

В классе BuildingCalc функция settingBuildCalc используется для того, чтобы дать имя окну, загрузить ui файл и указать работу кнопок, заполнения comb класса бетона и арматуры.

Функция selectcombobeton заполняет переменные и в зависимости от выбора класса бетона.

Функция selectcomboarmatura заполняет переменные и в зависимости от выбора класса арматуры.

Функция take берёт результат вычисления для дальнейшей передачи в функцию вывода результата вычислений.

Функция help\_form используется для открытия окна help.

Функция input\_A\_sиспользуется для открытия окна с таблицей значений .

Функция show\_historyиспользуется для открытия окна История вычислений.

Функция clear\_data применяется для очищения всех введённых ранее полей в главном окне.

Функция clear\_file очищает текстовый файл с историей вычислений.

Функция print\_result производит проверку на корректность ввода, вычисляет и передаёт строку в окно с результатом вычислений.

# 2.3Инструкция по эксплуатации программы

Таблица1. Основные действия программы

|  |  |
| --- | --- |
| Нажатие на клавишу | Действие |
| Вычислить с текущими данными | Вызов функции, которая выполняет одну из основных задач про-граммы: проводит расчет прямоугольного сечения с симметричной арматурой |
| Очистить текстовый файл | Вызов функции, которая удаляет данные из текстового файла |
| Содержимое текстового файла | Выводит на экран новое окно, в котором записана вся история вычислений |
| Очистить заполненные поля | Вызов функции, которая удаляет данные аргументов |
| Помощь | Выводит на экран новое окно, в котором отображается помощь в эксплуатации программы, информация о допустимых значениях и общая информация о переменных. |

На рис 1-13описана работа программы PyQT coursework

При запуске в среде PyCharmфайла PyQTcoursework открывается окно «Проверка прочности опорного сечения колонны» – меню, показанное на Рис. 1.

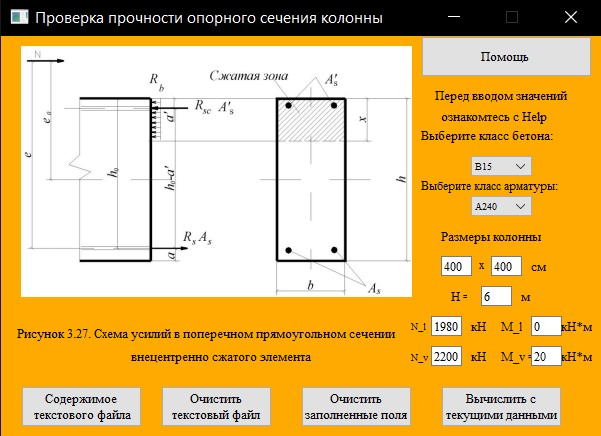


Рис. 1 –Окно при запуске cyrsovaya.py через среду PyCharm.

После нажатия на кнопку Помощь переходим на окно Помощь.

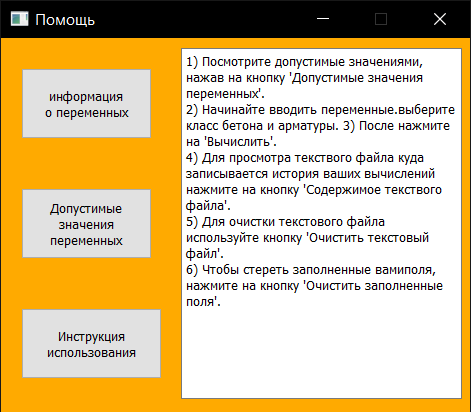


Рис. 2 –Окно Помощь

Чтобы узнать допустимые значения и кратко о переменных созданы кнопки «информация о переменных» Рис. 3 и «Допустимые значения переменных» Рис.4.

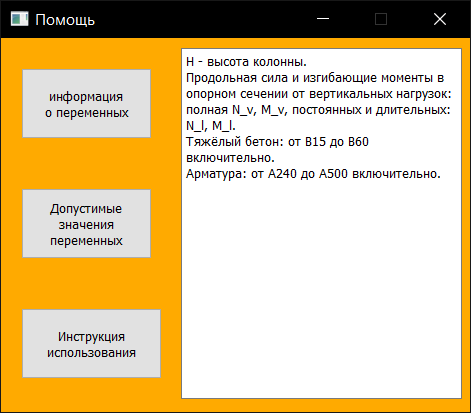


Рис. 3 – Окно Помощь после нажатия на кнопку «информация о переменных»

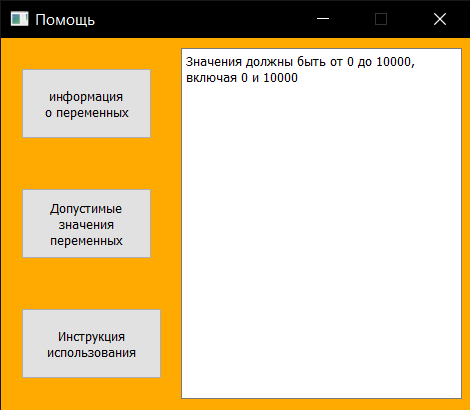


Рис 4 – Окно Помощь нажатия на кнопку «Допустимые значения переменных»

При нажатия на кнопку «Вычислить с текущими данными» откроется окно «Результат» Рис. 5.

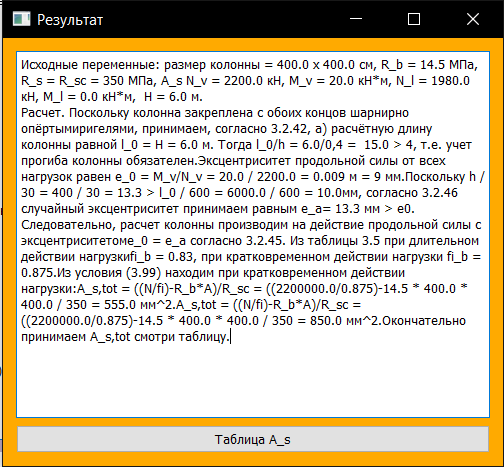


Рис 5. – Окно результат.

После нажатия на кнопку «Содержимое текстового файла» откроется окно Содержимое текстового файла Рис. 6

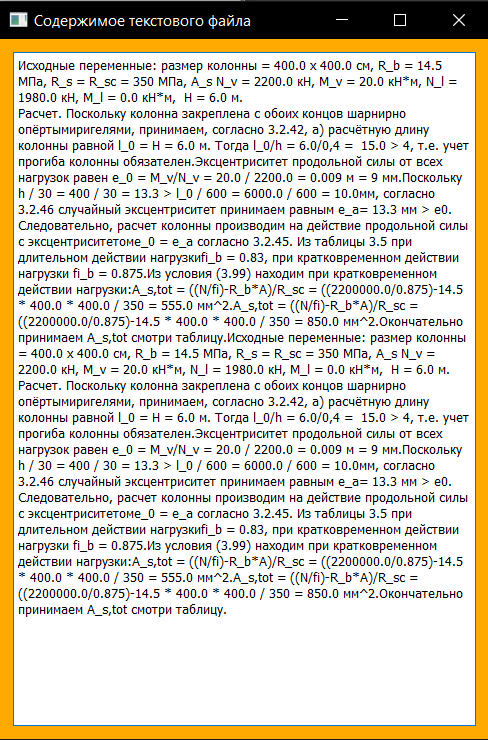


Рис 6 – Окно История вычислений.

При нажатия на кнопку «Очистить текстовый файл» очищается текстовый файл и окно Содержимое текстового файла пустое Рис.7

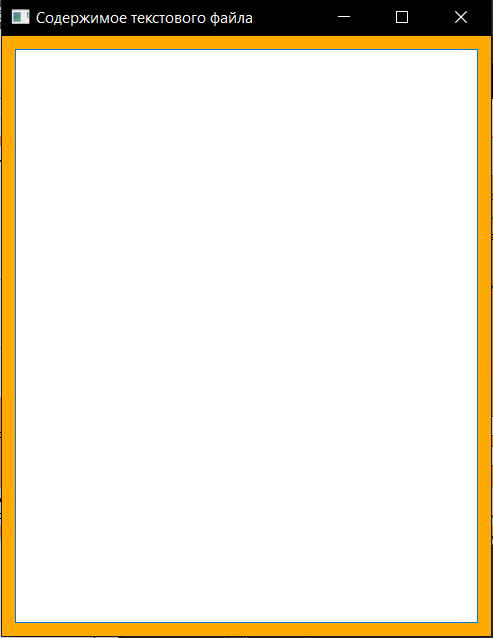


Рис 7 – Окно Содержимое текстового файла

После нажатия на кнопку «Очистить заполненные поля» все заполненные поля в меню ранее, очищаются Рис 8.

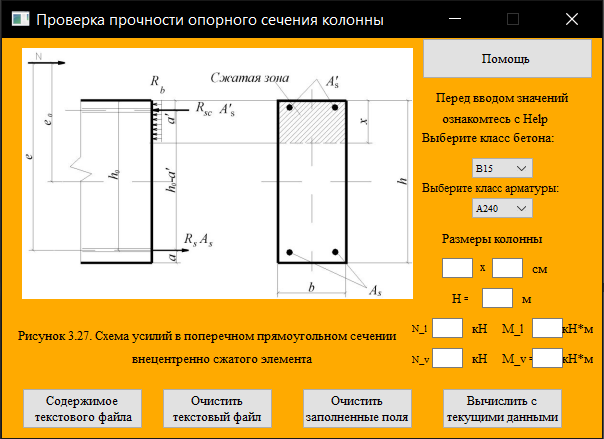


Рис 8 – Окно Проверка прочности опорного сечения колонны

После нажатия на кнопку «Таблица A\_s» Рис.9



Рис 12 – Окно Таблица значений

# ЗАКЛЮЧЕНИЕ

В курсовой работе решена поставленная задача, а именно: создание программы, которая рассчитает железобетонные конструкции без предварительно напряженной арматуры. Программа была составлена и отлажена. С ее помощью были рассчитаны железобетонные конструкции без предварительно напряженной арматуры(прямоугольные сечения с симметричной арматурой). Тестирование программы, показало, что поставленная задача успешно решена.

В ходе курсовой работы были приобретены теоретические и практические навыки программирования на языке Python.QMainWindow и QWidget использовалось для создания главного класса. Весь графический дизайн был создан с помощью программы QtDisigner, дизайн был импортирован с помощью uiс файлов. Были использованы такие виджеты как QLabel, QPushButton, QLineEdit, QTextEdit, comboBox и т.д. Были созданы множество пользовательских функций, был освоен явный тип передачи в функции, создавали «примитивные» классы и работали в них. Также работали со строками: записывали строку в текстовый файл, читали в строку информацию из текстового файла. В ходе расчета добавляли в нее строки с вычислениями, переводили в тип float. Были использованы базы данных с помощью библиотеки sqllite 3 в которых хранились классы бетона.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 1280 с.
2. Методическое пособие к СП 63.13330. Расчет железобетонных конструкций без предварительно напряженной арматуры / Министерство строительства и жилищно-коммунального хозяйства Российской Федерации. –2015. – 294 с.
3. QT [Электронный ресурс]: Официальный сайт- Режим доступа: https://www.qt.io/
4. Электронный фонд правовых и нормативно-технических документов[ Электронный ресурс] : бетонные и железобетонные конструкции– Режим доступа: https://docs.cntd.ru/document/554403082
5. Stepik [Электронный ресурс]: Курс по Python- Режим доступа: https://stepik.org/course/83053/syllabus

## **ПРИЛОЖЕНИЕ А**

(главный листинг исходного кода программы)

**Файл cyrsovaya.py**

import sys

import sqlite3

from PyQt5 import uic

from PyQt5.QtWidgets import QApplication, QWidget, QMainWindow

conn = sqlite3.connect('class\_beton.db')

cur = conn.cursor()

# cur.execute("""CREATE TABLE IF NOT EXISTS beton(

# class TEXT,

# mean TEXT);

# """)

# conn.commit()

# all\_beton = [('B15', '8.5'), ('B20', '11.5'), ( 'B25', '14.5'),('B30', '17'),

# ('B35', '19.5'),('B40', '22'),

# ('B45', '25'), ('B50', '27.5'), ('B55', '30'), ('B60', '33')]

# cur.executemany("INSERT INTO beton VALUES(?, ?);", all\_beton)

# conn.commit()

cur.execute("SELECT \* FROM beton;")

one\_result = cur.fetchall()

class Calc(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setting\_calc()

def setting\_calc(self):

uic.loadUi("Calculating.ui", self)

self.setWindowTitle('Результат')

self.pushButton.clicked.connect(self.input\_A\_s)

def input\_A\_s(self):

self.take\_A\_s = Asform()

self.take\_A\_s.show()

def returnres(self, result):

self.textedit.setText(result)

text = open('zxc.txt')

history = text.read()

text.close()

with open('zxc.txt', 'w'): pass

text = open('zxc.txt', 'a+')

text.write(result)

text.write(history)

text.close()

class Help(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setting\_help()

def setting\_help(self):

uic.loadUi("help.ui", self)

self.setWindowTitle('Помощь')

self.textEdit.setText("1) Посмотрите допустимые значениями,"

" нажав на кнопку 'Допустимые значения переменных'.\n"

"2) Начинайте вводить переменные."

"выберите класс бетона и арматуры. "

"3) После нажмите на 'Вычислить'.\n"

"4) Для просмотра текствого файла"

" куда записывается история ваших "

"вычислений нажмите на кнопку 'Содержимое текствого файла'.\n"

"5) Для очистки текстового "

"файла используйте кнопку 'Очистить текстовый файл'.\n"

"6) Чтобы стереть заполненные вамиполя, "

"нажмите на кнопку 'Очистить заполненные поля'.")

self.info.clicked.connect(self.info\_about)

self.instuction.clicked.connect(self.instraction\_use)

self.limit.clicked.connect(self.limit\_mean)

def info\_about(self):

string = "H - высота колонны.\n" \

"Продольная сила и изгибающие моменты в опорном сечении " \

"от вертикальных нагрузок: полная N\_v, M\_v," \

" постоянных и длительных: N\_l, M\_l.\n" \

"Тяжёлый бетон: от B15 до B60 включительно.\n" \

"Арматура: от A240 до A500 включительно.\n"

self.textEdit.setText(string)

def limit\_mean(self):

strin = "Значения должны быть от 0 до 10000, включая 0 и 10000"

self.textEdit.setText(strin)

def instraction\_use(self):

self.textEdit.setText("1) Посмотрите допустимые значениями,"

" нажав на кнопку 'Допустимые значения переменных'.\n"

"2) Начинайте вводить переменные."

"выберите класс бетона и арматуры. "

"3) После нажмите на 'Вычислить'.\n"

"4) Для просмотра текствого "

"файла куда записывается история ваших "

"вычислений нажмите на кнопку 'Содержимое текствого файла'.\n"

"5) Для очистки текстового "

"файла используйте кнопку 'Очистить текстовый файл'.\n"

"6) Чтобы стереть заполненные вамиполя, "

"нажмите на кнопку 'Очистить заполненные поля'.")

class Asform(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.setting\_asform()

def setting\_asform(self):

uic.loadUi("A\_sForm.ui", self)

self.setWindowTitle('Таблица значений A\_s')

class HistoryCalc(QWidget):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.settingHistoryCalc()

def settingHistoryCalc(self):

uic.loadUi("histirycalc.ui", self)

self.setWindowTitle('Содержимое текстового файла')

def concatStr(self):

text = open('zxc.txt')

a = text.read()

text.close()

result = a

self.textedit1.setText(self.textedit1.toPlainText() + result + '\n')

class BuildingCalc(QMainWindow):

def \_\_init\_\_(self):

super().\_\_init\_\_()

self.string\_result = ''

self.settingBuildCalc()

self.res = ''

self.R\_s = 350

self.R\_sc = 350

self.R\_b = 14.5

def settingBuildCalc(self):

uic.loadUi("Mainwindow.ui", self)

self.setWindowTitle("Проверка прочности опорного сечения колонны")

self.calculating.clicked.connect(self.print\_result)

self.history\_calc.clicked.connect(self.show\_history)

self.clear.clicked.connect(self.clear\_data)

self.btn\_clear\_file.clicked.connect(self.clear\_file)

self.help.clicked.connect(self.help\_form)

for i in one\_result:

self.comboBox.addItem(i[0])

self.comboBox.activated[str].connect(self.selectcombobeton)

self.comboBox\_2.addItem("A240")

self.comboBox\_2.addItem("A400")

self.comboBox\_2.addItem("A500")

self.comboBox\_2.activated[str].connect(self.selectcomboarmatura)

def selectcombobeton(self):

self.meanbox = str(self.comboBox.currentText())

for i in one\_result:

if self.meanbox == i[0]:

self.R\_b = i[1]

def selectcomboarmatura(self):

self.meanbox = str(self.comboBox\_2.currentText())

if self.meanbox == "A240":

self.R\_s = 210

self.R\_sc = 210

elif self.meanbox == "A400":

self.R\_s = 340

self.R\_sc = 340

elif self.meanbox == "A500":

self.R\_s = 435

self.R\_sc = 435

def take(self, result):

self.res1 = result

def help\_form(self):

self.help = Help()

self.help.show()

def show\_history(self):

self.history\_result = HistoryCalc()

self.history\_result.concatStr()

self.history\_result.show()

def clear\_data(self):

self.first\_size.clear()

self.second\_size.clear()

self.five.clear()

self.six.clear()

self.seven.clear()

self.eight.clear()

self.eight\_2.clear()

def clear\_file(self):

with open('zxc.txt', 'w'): pass

def print\_result(self):

self.calc = Calc()

self.a1 = self.first\_size.text()

self.a2 = self.second\_size.text()

self.N\_v = self.five.text()

self.M\_v = self.six.text()

self.N\_l = self.seven.text()

self.M\_l = self.eight.text()

self.h\_size = self.eight\_2.text()

if not self.a1.strip() or not self.a2.strip() \

or not self.N\_v.strip() \

or not self.M\_v.strip() or not self.N\_l.strip() \

or not self.M\_l.strip() or not self.h\_size.strip():

self.string\_result += ("Проверьте все ли поля заполнены!!!\n\n")

self.take(self.string\_result)

elif self.a1.isdigit() == False or self.a2.isdigit() == False \

or self.N\_v.isdigit() == False \

or self.M\_v.isdigit() == False or self.N\_l.isdigit() == False \

or self.M\_l.isdigit() == False or self.h\_size.isdigit() == False:

self.string\_result = ("Не все поля, введённые "

"вами, являются числами!\n")

self.take(self.string\_result)

elif (float(self.a1) < 0 or float(self.a1) > 10000) or \

(float(self.a2) < 0 or float(self.a2) > 10000) or \

(float(self.N\_v) < 0 or float(self.N\_v) > 10000) or \

(float(self.M\_v) < 0 or float(self.M\_v) > 10000) or \

(float(self.N\_l) < 0 or float(self.N\_l) > 10000) or \

(float(self.M\_l) < 0 or float(self.M\_l) > 10000) or \

(float(self.h\_size) < 0 or float(self.h\_size) > 10000):

self.string\_result = ("Вы вышли за диапазон допустимых значений,\n"

" пожалуйста ознакомьтесь "

"с допустимыми значениями в разделе Помощь ")

else:

self.a1 = float(self.first\_size.text())

self.a2 = float(self.second\_size.text())

self.N\_v = float(self.five.text())

self.M\_v = float(self.six.text())

self.N\_l = float(self.seven.text())

self.M\_l = float(self.eight.text())

self.h\_size = float(self.eight\_2.text())

self.string\_result += f"Исходные переменные:" \

f" размер колонны = {self.a1} x {self.a2} cм, " \

f"R\_b = {self.R\_b} МПа, " \

f"R\_s = R\_sc = {self.R\_s} МПа, A\_s " \

f"N\_v = {self.N\_v} кH, M\_v = {self.M\_v} кH\*м, " \

f"N\_l = {self.N\_l} кH, M\_l = {self.M\_l} кH\*м, " \

f" H = {self.h\_size} м.\n"

e\_0 = round(int(self.M\_v) / int(self.N\_v), 3)

e\_0\_r = e\_0\*1000

l\_0 = self.h\_size\*1000

h = 400

e\_a = round(h/30, 1)

fi\_b1 = 0.83

fi\_b2 = 0.875

self.string\_result += f"Расчет. Поскольку колонна закреплена" \

f" с обоих концов шарнирно опёртыми" \

f"ригелями, принимаем, согласно 3.2.42," \

f" а) расчётную длину колонны равной" \

f" l\_0 = H = {self.h\_size} м. Тогда l\_0/h = {self.h\_size}/0,4 = " \

f" {int(self.h\_size) / 0.4} > 4, " \

f"т.е. учет прогиба колонны обязателен." \

f"Эксцентриситет продольной силы от всех нагрузок равен e\_0 = M\_v/N\_v" \

f" = {self.M\_v} / {self.N\_v} = {e\_0} м = {int(e\_0\_r)} мм." \

f"Поскольку h / 30 = {h} / 30 = {round(h/30, 1)} > l\_0 / 600 = " \

f"{l\_0} / 600 = {round(l\_0/600, 2) }мм, согласно 3.2.46 случайный эксцентриситет" \

f" принимаем равным e\_a= {e\_a} мм > е0. Следовательно, " \

f"расчет колонны производим на" \

f" действие продольной силы с эксцентриситетом" \

f"e\_0 = e\_a cогласно 3.2.45." \

f" Из таблицы 3.5 при длительном действии нагрузки" \

f"fi\_b = {fi\_b1}, при кратковременном действии нагрузки fi\_b = {fi\_b2}." \

f"Из условия (3.99) находим при кратковременном действии нагрузки:"

N1 = self.N\_v\*1000

N2 = self.N\_l\*1000

A\_s\_tot1 = ((N1/fi\_b2)-self.R\_b \* self.a1 \* self.a2) / self.R\_sc

self.string\_result += f"A\_s,tot = ((N/fi)-R\_b\*A)/R\_sc =" \

f" (({N1}/{fi\_b2})-{self.R\_b} \* " \

f"{self.a1} \* {self.a2} / " \

f"{self.R\_sc} = {round(A\_s\_tot1, 0)} мм^2."

A\_s\_tot2 = ((N2 / fi\_b1) - 0.9 \* self.R\_b \*

self.a1 \* self.a2) / self.R\_sc

self.string\_result += f"A\_s,tot = " \

f"((N/fi)-R\_b\*A)/R\_sc =" \

f" (({N1}/{fi\_b2})-{self.R\_b} \* " \

f"{self.a1} \* {self.a2}" \

f" / {self.R\_sc} = {round(A\_s\_tot2, 0)} мм^2."

self.string\_result += "Окончательно принимаем A\_s,tot смотри таблицу."

self.take(self.string\_result)

self.calc.returnres(self.res1)

self.calc.show()

self.string\_result = ''

if \_\_name\_\_ == '\_\_main\_\_':

app = QApplication(sys.argv)

ex = BuildingCalc()

ex.show()

sys.exit(app.exec())