|  |  |  |
| --- | --- | --- |
|  |  | Данилов Вадим 11В |

**Документация по проекту**

**«Межпредметный калькулятор»**

**Содержание**

[Для разработчиков 2](#_Toc71468013)

[Руководство пользователя 6](#_Toc71468014)

Для разработчиков

**Язык реализации**: Python 3

**Для ОС**: MS Windows

**Функции:**

* Возможность выбор предмета, по которому будут производиться расчёты
* Возможность выбора калькулятора
* Возможность ввода данных для расчёта и их обработка
* Вывод иллюстрации данных необходимых для ввода (для калькулятора Геометрии)
* Вывод результата
* Экспорт данных

Целью этого проекта является, как можно понять названию, калькуляторы и расчёты в них. В ней можно производить расчёты по интересующим вас предметам с возможностью дальнейшего экспортирования проделанных вами расчётов.

**Что из себя представляет программа:**

* Сама программа. С оконно-графическим интерфейсом.
* Внутри программы находится 32 встроенных калькулятора. Из которых:

по физике – 7, по химии – 1, по математике – 24

(более подробную информацию можно найти в сл. пункте)

* Файл экспорта и логов

При запуске каждый калькулятор запоминает расчёты, которые в нём производятся и добавляет их в файл экспорта, а также добавляет в логи все попытки перевода в нём (удачные и неудачные)

**Список калькуляторов и что в них включено:**

* 1. Список доступных предметов:
     1. Геометрия
     2. Физика
     3. Химия
  2. Список калькуляторов по геометрии:
     1. Расчет объема и площади поверхности куба, пирамиды, конуса,

цилиндра, шара, призмы

* + 1. Расчет площади квадрата, прямоугольника, параллелограмма, треугольника,

трапеции, ромба, круга

* + 1. Расчет периметра квадрата, прямоугольника, треугольника, круга
    2. Расстояние между двумя точками (по координатам) в 1-мерной,

2-мерной и 3-мерной

* 1. Список калькуляторов по физике:
     1. Расчет первой космической скорости
     2. Расчёт второй космической скорости
     3. Расчёт закона всемирного тяготения Ньютона (сила тяготения)
     4. Расчёт закона Ома для участка цепи (сила тока)
     5. Расчёт силы, массы, ускорения (1 закон ньютона)
     6. Расчет центростремительного ускорения при движении по окружности
  2. Список калькуляторов по химии:
     1. Расчет массы вещества в растворе

**Интерфейс состоит из:**

* Название программы
* Переключатель предметов
* Переключатель калькуляторов
* Поле для ввода параметров
* Отображение иллюстрации данных необходимых для ввода (для Геометрии)
* Отображение результата

**Основной принцип работы:**

Главное окно сделано по принципу:

**win = Tk()**

**win.geometry('ширина x высота') #задаёт размеры окна**

**win.title('Заголовок') #заголовок приложения**

**win.resizable(height = False, width = False) #запрещает**

**#растягивать окно**

**...**

**win.mainloop() #запускает бесконечный цикл, пока не**

**#закроют приложение**

Большинство калькуляторов сделано по принципу:

**def G1cub():**

**wing\_g = Toplevel()) #открывает новое окно**

**win\_g.title('Заголовок') #заголовок этого окна**

**win\_g.geometry('450x350') #задаёт размеры окна**

**win\_g.resizable(height = False, width = False) #запрещает**

**#растягивать окно**

**def G1cub1():**

**#Обработка полей ввода**

**#Если поля пустые, то попросить ввести данные**

**if (ent\_gcub1.get() == ''):**

**lab\_gcub3.configure(text='Пожалуй, стоит ввести данные', fg = 'red')**

**#Если поля содержат буквы, то попросить изменить данные**

**elif (ent\_gcub1.get().isdigit()) == False:**

**lab\_gcub3.configure(text='Введите корректные данные', fg = 'red')**

**#Если данные введены корректно, то производится расчёт**

**#Ниже есть строка, которая пока равна “, как только будет**

**#произведён расчёт, то строка примет значение ответа**

**else:**

**lab\_gcub3.configure(text=str(int(ent\_gcub1.get()) \*\* 3), fg = 'black')**

**#Процедура очистки полей при нажатии кнопки «Очистить».**

**def Del():**

**lab\_gcub3.configure(text='')**

**ent\_gcub1.delete(0, END)**

**#При закрытии калькулятора, с радиокнопки сбрасывается значение.**

**def closing():**

**radio\_var2.set(-1)**

**win\_g.destroy()**

**#Дальше идёт интерфейс**

**#Текст перед полями ввода**

**lab\_gcub1 = Label(win\_g, text='Введите сторону куба:', padx = 3, pady = 2)**

**lab\_gcub2 = Label(win\_g, text='Ответ:', padx = 3, pady = 2)**

**#Строка, которая хранит в себе ответ**

**lab\_gcub3 = Label(win\_g, text='')**

**#Его размещение на экране**

**lab\_gcub1.grid(column=0, row=0, sticky=W)**

**lab\_gcub2.grid(column=0, row=1, sticky=W)**

**lab\_gcub3.grid(column=1, row=1, sticky=W)**

**#Поля ввода**

**ent\_gcub1 = Entry(win\_g, width=10)**

**ent\_gcub1.grid(column=1, row=0, sticky=W)**

**#При открытии калькулятора ставит курсор на поле ввода**

**ent\_gcub1.focus\_set()**

**global photo**

**#Присваивание в переменную иллюстрации**

**photo = tk.PhotoImage(file = './img/volume\_cube.png')**

**#Вывод иллюстрации и её расположение**

**image\_label2 = tk.Label(win\_g,image = photo)**

**image\_label2.place(relx=0.30, rely=0.30)**

**#Кнопки «Расчитать» и «Очистить».**

**#Кнопка «Расчитать» запускает функцию для расчёта значений.**

**but\_gcub1 = Button(win\_g, text='Расчитать', command=G1cub1, cursor = 'hand1')**

**#Кнопка «Очистить» запускает функцию для очистки полей ввода.**

**but\_gcub2 = Button(win\_g, text='Очистить', command=Del, cursor = 'hand2')**

**but\_gcub1.grid(column=0, row=5)**

**but\_gcub2.grid(column=1, row=5, sticky=W)**

**#Обработка закрытия калькулятора. Запускает функцию для сброса**

**#значения радиокнопки.**

**win\_g.protocol("WM\_DELETE\_WINDOW", closing)**

Экспорт:

**exporting = open("output.txt", "w")**

**#Создание файла экспорта и открытия его для записи**

**...**

**#Данные записываются в формате:**

**#Номер) Название калькулятора Данные1 Данные 2 --> Ответ**

**#Данные1 и Данные2 это данные взятые с полей ввода**

**exporting.write(str(count) + ') ' + 'Название ' + '\n' + 'Данные1 = '+ ent\_f1.get() + ' (у.е.)' + ' ' + 'Данные2 = ' + ent\_f2.get() + ' (у.е.)' + ' ' + '--> ' + str(формула) + ' у.е' + '\n')**

**...**

**exporting.close() #Закрывает файл экспорта**

Логи:

**masf = [] #Список в котором сохраняются все действия**

**...**

**now = datetime.datetime.now() #текущее время**

**#Строка, которая в последствии добавляется в список**

**#Строка содержит в себе дату в формат День:Месяц:Год Час:Мин:Сек**

**s = str(now.strftime("%d-%m-%Y %H:%M:%S")) + ' | Действие1'**

**masf.append(s) #добавляется в список**

**...**

**#Когда закрывается окно, происходит запись в конец файла всех**

**#действий из списка**

**for x in masf:**

**exporting.write(x + '\n')**

Руководство пользователя

**Язык реализации**: Python 3

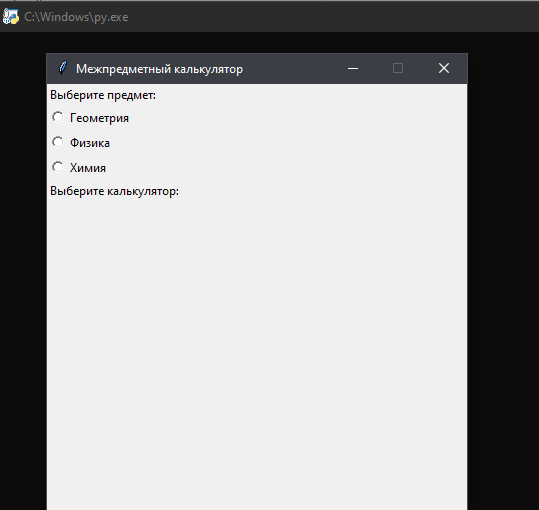
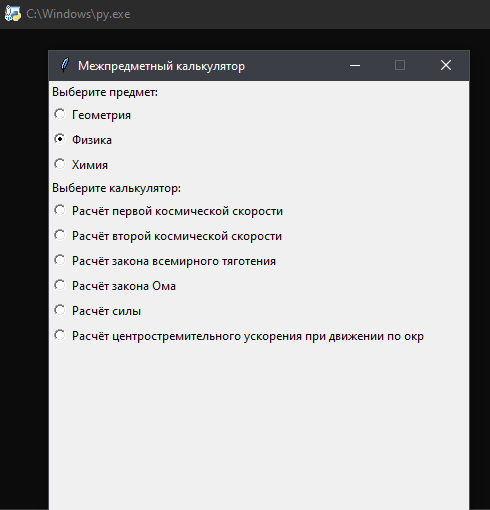
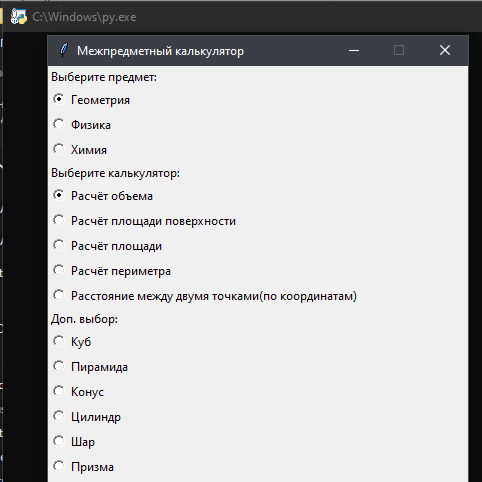
**Для ОС**: MS Windows

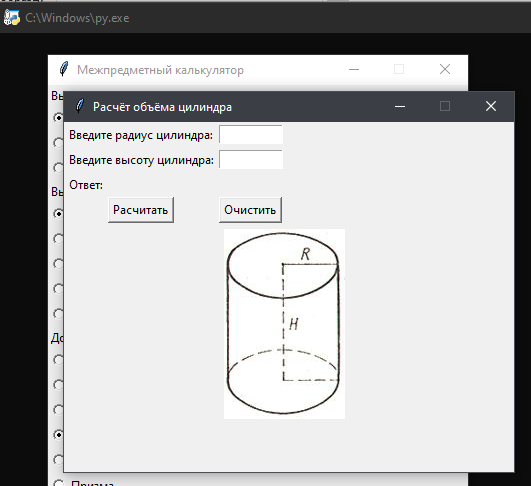
**Функции:**

* Возможность выбор предмета, по которому будут производиться расчёты
* Возможность выбора калькулятора
* Возможность ввода данных для расчёта и их обработка
* Вывод иллюстрации данных необходимых для ввода (для калькулятора Геометрии)
* Вывод результата
* Экспорт данных

Целью этого проекта является, как можно понять названию, калькуляторы и расчёты в них. В ней можно производить расчёты по интересующим вас предметам с возможностью дальнейшего экспортирования проделанных вами расчётов.

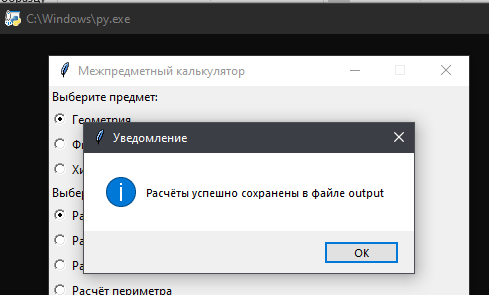
**Как пользоваться программой:**

1. Запустите файл main.py
2. В открывшемся окне выберите интересующий вас предмет (Геометрия, Физика, Химия)
3. Далее выберите нужный калькулятор.
4. Дальше нажимаете на интересующий вас калькулятор (В предмете Геометрия может потребоваться доп.выбор)
5. В открывшемся окне нужно ввести данные для расчёта



|  |
| --- |
| **Возможные проблемы**:   * + Вы ничего не ввели и нажали кнопку «Рассчитать». В результате программа выдаст ошибку.   + Вы заполнили не все поля и нажали кнопку «Рассчитать». В результате программа выдаст ошибку.   + Вы ввели буквы вместо цифр и нажали кнопку «Рассчитать». В результате программа выдаст ошибку.   В любом случае программа уведомит вас, если вы не ввели данные/заполнили не все поля или ввели не цифры, соответствующим сообщением. |

1. После того, как вы использовали нужный калькулятор, чтобы его закрыть достаточно нажать на крестик в правом верхнем углу.
2. После того, как вы сделали всё, что хотели, для того чтобы закрыть программу, нужно проделать аналогичные действия. Нажать на крестик в правом верхнем углу.
3. Потом у вас появится окно, которое уведомит вас, что файл экспорта и логи успешно сохранены.



1. Файл экспорта и логов сохраняется в той же директории, откуда запускается программа

Имейте в виду, что после каждого запуска программы файл перезаписывается.

|  |
| --- |
| **Стоит иметь в виду**:   * + После каждого запуска программы, файл экспорта перезаписывается.   + Файл экспорта нужно смотреть только после того, как вы закрыли программу. В противном случае, если вы откроете файл, то он будет пустой.   + Нельзя вносить изменения в файл экспорта во время работы программы. За последствия от внесенных вами изменений отвечаете вы. |