**Инструкция по работе на мастер-классе по разработке программы Математический тренажер   
(10 - 11 классы)**

1. **Клонирование репозитория с инструкцией**

**git clone [https://github.com/TimZlat/](https://github.com/TimZlat/%20profprobaPy)****[profprobaPy](https://github.com/TimZlat/%20profprobaPy)**

1. **2. Переход в рабочую папку**

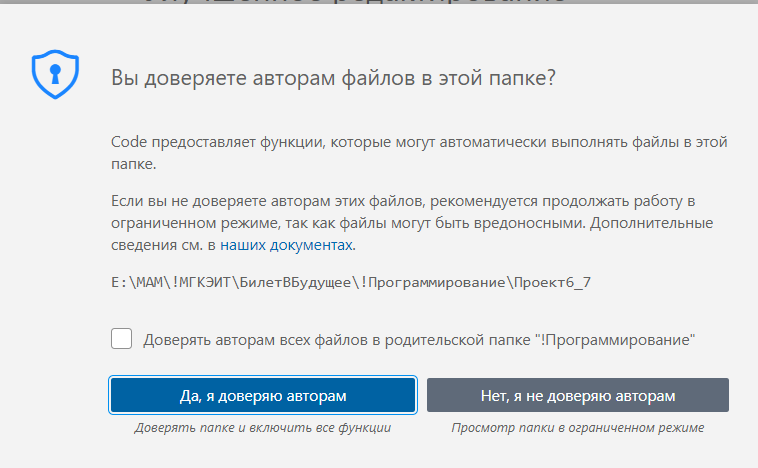
**cd profprobaPy**

1. Откройте документ **ИнструкцияМК\_90.pdf**

Основная часть

1. На рабочем диске создайте папку для хранения проекта, имя папки **PyProject**.
2. Запустите MS Visual Studio Code
3. Выполните команду:

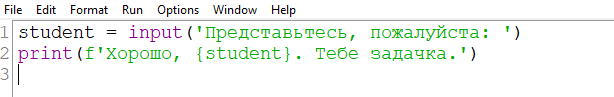
**Файл →Открыть папку** и выберите созданную папку. На запрос о доверенности папке, подтвердите, что вы доверяете авторам и файлам в папке.



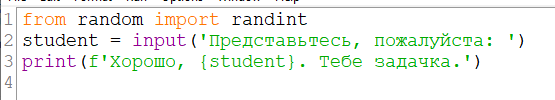
1. Выполните команду **Файл → Создать файл** (***выберите тип файла файл Python***)
2. После этого откроется пустая вкладка редактора для ввода кода программы. Сразу же сохраните пустой файл командой

**Файл → Сохранить как…** (и присвойте файлу имя **pr1011.py** (расширение py можно не ставить, если перед этим вы выбрали тип файла файл Python)

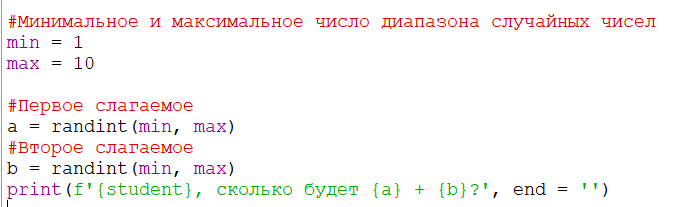
1. Начнем программу с того, что спросим у ученика имя. Каждому пользователю приятно, когда программа помнит, как его зовут.



1. Добавим импорт в самый верх файла.



1. Загадаем случайные слагаемые в интервале от **min** до **mах** и отобразим задание на экране:



1. Сохраним **File Save (Ctrl + S)** и запустим программу 

Теперь программа обращается к пользователю по имени и при каждом запуске загадывает новую задачку.

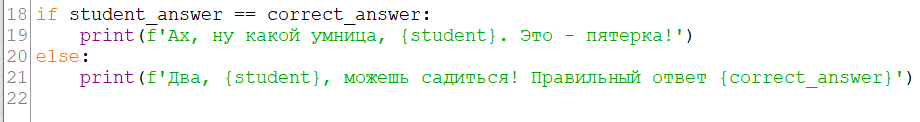
Прежде чем проверить корректность ответа пользователя, программа должна сама узнать правильный ответ. Здесь нет ничего сложного оба слагаемых уже сохранены в переменных а и Ь.

1. Найдем правильный ответ — сумму а и b — и спросим ответ у пользователя.



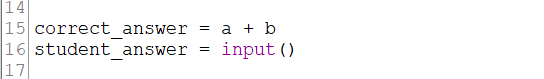
Программа уже знает правильный ответ, он находится в переменной **correct\_answer.**

1. Запишем условие правильности ответа и подставим его в полную форму условного оператора.



Мы записали условие правильности ответа как **student\_answer == correct\_answer**. Проверяем, идентичен ли ответ пользователя правильному ответу, и, если да, ставим пятерку, если нет — двойку.

Программа не сломалась, но работает некорректно. В ней появились «баги». Чтобы разобраться, почему так получилось, посмотрим внимательно на фрагмент, предшествующий условию:



В программе сравниваются два значения двух разных типов, а это некорректно.

Для сравнения значений переменных **correct\_answer** и **student\_answer** их нужно привести к одному типу. Безопаснее привести **correct\_answer** к строке, чем **student\_answer** приводить к целому.

1. Перепишем проверку условия — сконвертируем число в строку:



1. Запустим программу и убедимся в корректности ее работы.
2. Усложним нашу программу добавив в нее следующие возможности: выбор уровня сложности и возможность вычисления пяти заданий с выставлением итоговой оценки.
3. Для этого создайте в рабочей папке новый файл. Выполните команду **Файл → Создать файл** (выберите тип файла файл Python)
4. После этого откроется пустая вкладка редактора для ввода кода программы. Сразу же сохраните пустой файл командой Файл → Сохранить как… (и присвойте файлу имя prog.py.
5. В окне вкладки наберите следующий код (соблюдая все отступы)

from random import randint

student = input('Представьтесь, пожалуйста: ')

#Зададим уровень сложности тренажера

try:

    level =  int(input('Выберите уровень сложности 1-5: '))

except:

    level = 1

    print('Установлен первый уровень сложности \n')

if level < 1 or level > 5:

    level = 1

    print('Установлен первый уровень сложности \n')

print(f'Хорошо, {student}. Тебе задачка')

#Задаем границы диапазона случайных чисел

min = 10 \*\* (level - 1)

max = 10 \*\* level - 1

#организация цикла из 5 серий заданий и подсчета

#количества верных ответов

points = 0

for i in range(5):

    a = randint(min, max)

    b = randint(min, max)

    print(f'{student}, сколько будет {a} + {b}? ', end = '')

    correct\_answ = a + b

    student\_answ = input()

    if student\_answ == str(correct\_answ):

        points += 1

        print(f'Верно!!! ')

    else:

        print(f'Не верно! Правильный ответ {correct\_answ}')

if points == 5:

    print(f'Вы великолепны! {student}, ваша оценка - пять!')

elif points == 4:

    print(f'Молодец, {student}, но можно лучше! Ваша оценка - четыре!')

elif points == 3:

    print(f'Так себе, {student}! Ваша оценка - 3 (удовлетворительно)')

else:

    print(f'Учи арифметику, {student}! Ваша оценка - 2 (неудовлетворительно)')

1. Сохраните код программы используя команду **Файл → Сохранить** (или сочетание клавиш **Ctrl + S**)
2. Запустите программу на выполнение при помощи кнопки  сверху справа экрана или набрав в терминале команду   
   **python prog.py**
3. Программа запустится в окне терминала. Проверьте ее работу, отвечая на все сообщения программы.
4. Запустите программу несколько раз, выбирая разный уровень сложности в пределах допустимого. Проверьте работу программы.
5. Проверьте, как будет работать программа, если уровень сложности задать меньше 1 или больше 5.
6. Проверьте, как будет работать программа, если уровень сложности задать не числом, а символами. Например, не 3, а третий.
7. Подумайте и предложите вариант усложнения данной программы и алгоритма