

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/profprobaPy
```

2. Переход в рабочую папку

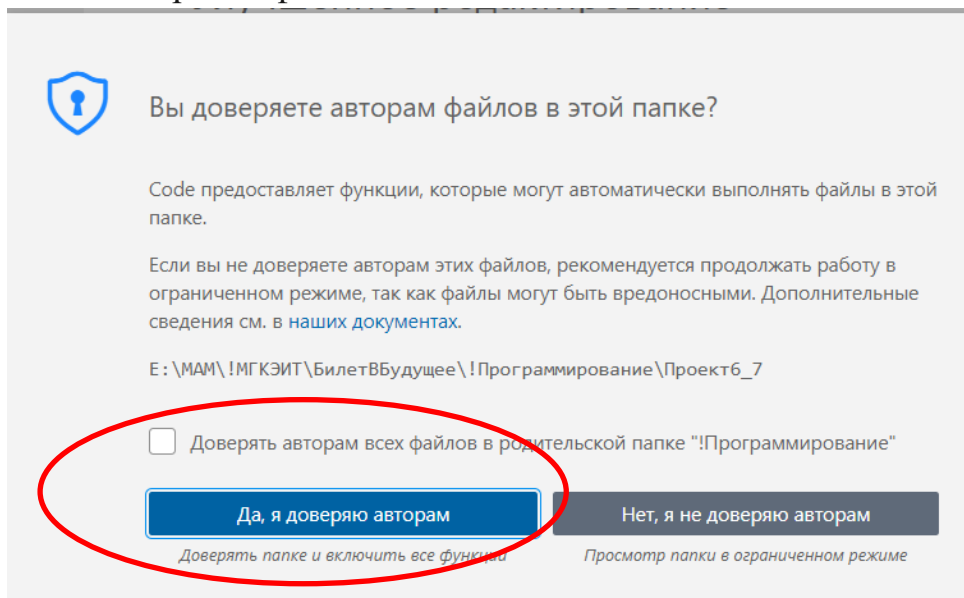
```
cd profprobaPy
```

3. Откройте документ **ИнструкцияМК_90.pdf**

Основная часть

4. На рабочем диске создайте папку для хранения проекта, имя папки **PyProject**.
5. Запустите MS Visual Studio Code
6. Выполните команду:

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



7. Выполните команду **Файл** → **Создать файл** (**выберите тип файла файл Python**)
8. 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))

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

```
File Edit Format Run Options Window Help
1 student = input('Представьтесь, пожалуйста: ')
2 print(f'Хорошо, {student}. Тебе задачка.')
3 |
```

4. Добавим импорт в самый верх файла.

```
1 from random import randint
2 student = input('Представьтесь, пожалуйста: ')
3 print(f'Хорошо, {student}. Тебе задачка.')
4 |
```

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

```
#Минимальное и максимальное число диапазона случайных чисел
min = 1
max = 10

#Первое слагаемое
a = randint(min, max)
#Второе слагаемое
b = randint(min, max)
print(f'{student}, сколько будет {a} + {b}?', end = '')
|
```

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



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

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

7. Найдем правильный ответ — сумму **a** и **b** — и спросим ответ у пользователя.

```
15 correct_answer = a + b
16 student_answer = input()
```

Программа уже знает правильный ответ, он находится в переменной **correct_answer**.

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

```
18 if student_answer == correct_answer:
19     print(f'Ах, ну какой умница, {student}. Это - пятерка!')
20 else:
21     print(f'Два, {student}, можешь садиться! Правильный ответ {correct_answer}')
22
```

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:

```
14
15 correct_answer = a + b
16 student_answer = input()
17
```

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.

10. Перепишем проверку условия — сконвертируем число в строку:

```
17
18 if student_answer == str(correct_answer):
```

11. Запустим программу и убедимся в корректности ее работы.

12. Усложним нашу программу добавив в нее следующие возможности: выбор уровня сложности и возможность вычисления пяти заданий с выставлением итоговой оценки.

13. Для этого создайте в рабочей папке новый файл. Выполните команду **Файл → Создать файл** (выберите тип файла файл Python)

14. 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).

15. 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 = '')

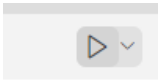
    correct_answ = a + b
    student_answ = input()

    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)')
```

16. Сохраните код программы используя команду **Файл → Сохранить** (или сочетание клавиш **Ctrl + S**)

17. Запустите программу на выполнение при помощи кнопки  сверху справа экрана или набрав в терминале команду **python prog.py**

18. Программа запустится в окне терминала. Проверьте ее работу, отвечая на все сообщения программы.

19. Запустите программу несколько раз, выбирая разный уровень сложности в пределах допустимого. Проверьте работу программы.

20. Проверьте, как будет работать программа, если уровень сложности задать меньше 1 или больше 5.

21. Проверьте, как будет работать программа, если уровень сложности задать не числом, а символами. Например, не 3, а третий.

22. Подумайте и предложите вариант усложнения данной программы и алгоритма