Домашнее задание.

Маратулы Темирболат

**Замечания по выполнению домашнего задания.**

1. **Вверху этого документа напишите свою Фамилию и Имя.**
2. Исходные коды Ваших программ сохраняйте прямо здесь, в тексте задания, строго под текстом каждого задания.
3. В дополнение к исходным кодам вставляйте в документ скриншот результата выполнения программ.
4. Не надо документ архивировать и вкладывать скриншоты в архив. Скриншоты сразу вставляете в документ, как картинку.
5. **Скриншоты удобно делать программой https://app.prntscr.com/ru/privacy.html**
6. Переменные начинаются с маленькими буквами, слова разделяются символом «\_» нижняя черта.
7. Константы пишутся заглавными буквами, слова разделяются символом «\_» нижняя черта.
8. **Не забывайте КРАСИВО оформлять интерфейс программ, за плохой, ЛЕНИВЫЙ интерфейс я тоже СНИЖАЮ ОЦЕНКУ!**

Создать собственный итерируемый класс Student, при итерации, класс должен возвращать, по очереди, оценки студента.

# Создать собственный итерируемый класс Student, при итерации, класс должен возвращать, по очереди, оценки студента.

ONE\_STEP = 1

class Student:

    def \_\_init\_\_(self,name):

        self.\_\_name = name

        self.\_\_subjects = []

        self.\_\_marks = []

    def get\_marks(self):

        return self.\_\_marks

    def get\_name(self):

        return self.\_\_name

    def get\_subjects(self):

        return self.\_\_subjects

    def \_\_iter\_\_(self):

        self.\_\_index = 0

        return self

    def \_\_next\_\_(self):

        if(self.\_\_index >= len(self.\_\_marks)):

            raise StopIteration

        else:

            current\_subject = self.\_\_subjects[self.\_\_index]

            current\_mark = self.\_\_marks[self.\_\_index]

            self.\_\_index += ONE\_STEP

            return  (current\_subject + ' : ' + str(current\_mark))

    def \_\_len\_\_(self):

        return len(self.\_\_subjects)

    def \_\_getitem\_\_(self,key):

        if(key == 'name'):

            return self.\_\_name

        else:

            if(key in self.\_\_subjects):

                return self.\_\_marks[self.\_\_subjects.index(key)]

            return IndexError

    def \_\_setitem\_\_(self,key,value):

        if(key in self.\_\_subjects):

            key\_index = self.\_\_subjects.index(key)

            self.\_\_marks[key\_index] = value

        else:

            self.\_\_subjects.append(key)

            self.\_\_marks.append(value)

student\_a = Student('Темирболат')

student\_a['Философия'] = 5

student\_a['Физика'] = 4

student\_a['Математика'] = 5

student\_a['Физкультура'] = 3

student\_a['Философия'] = 4

print('Оценки студента "{}":'.format(student\_a['name']))

for mark in student\_a:

    print('\t\t\t',mark)

