**Подвиг 3.** В программе объявлены два класса:

class Student:

def \_\_init\_\_(self, fio, group):

self.\_fio = fio # ФИО студента (строка)

self.\_group = group # группа (строка)

self.\_lect\_marks = [] # оценки за лекции

self.\_house\_marks = [] # оценки за домашние задания

def add\_lect\_marks(self, mark):

self.\_lect\_marks.append(mark)

def add\_house\_marks(self, mark):

self.\_house\_marks.append(mark)

def \_\_str\_\_(self):

return f"Студент {self.\_fio}: оценки на лекциях: {str(self.\_lect\_marks)}; оценки за д/з: {str(self.\_house\_marks)}"

class Mentor:

def \_\_init\_\_(self, fio, subject):

self.\_fio = fio

self.\_subject = subject

Первый класс описывает студентов, а второй - менторов. Вам поручается на основе базового класса Mentor разработать еще два дочерних класса:

**Lector** - для описания лекторов;  
**Reviewer** - для описания экспертов.

Объекты этих классов должны создаваться командами:

lector = Lector(fio, subject)

reviewer = Reviewer(fio, subject)

где fio - ФИО (строка); subject - предмет (строка). Инициализации этих параметров (fio, subject) должна выполняться базовым классом Mentor.

В самих классах Lector и Reviewer необходимо объявить метод:

def set\_mark(self, student, mark): ...

для простановки оценки (mark) студенту (student). Причем, в классе Lector оценки добавляются в список \_lect\_marks объекта класса Student, а в классе Reviewer - в список \_house\_marks. Используйте для этого методы add\_lect\_marks() и add\_house\_marks() класса Student.

Также в классах Lector и Reviewer должен быть переопределен магический метод:

\_\_str\_\_()

для формирования следующей информации об объектах:

- для объектов класса Lector: Лектор <ФИО>: предмет <предмет>  
- для объектов класса Reviewer: Эксперт <ФИО>: предмет <предмет>

Пример использования классов (эти строчки в программе писать не нужно):

lector = Lector("Балакирев С.М.", "Информатика")

reviewer = Reviewer("Гейтс Б.", "Информатика")

students = [Student("Иванов А.Б.", "ЭВМд-11"), Student("Гаврилов С.А.", "ЭВМд-11")]

persons = [lector, reviewer]

lector.set\_mark(students[0], 4)

lector.set\_mark(students[1], 2)

reviewer.set\_mark(students[0], 5)

reviewer.set\_mark(students[1], 3)

for p in persons + students:

    print(p)

# в консоли будет отображено:

# Лектор Балакирев С.М.: предмет Информатика

# Эксперт Гейтс Б.: предмет Информатика

# Студент Иванов А.Б.: оценки на лекциях: [4]; оценки за д/з: [5]

# Студент Гаврилов С.А.: оценки на лекциях: [2]; оценки за д/з: [3]

P.S. В программе требуется объявить только классы. На экран выводить ничего не нужно.  
P.P.S. Подумайте, где в этой программе полиморфизм.

class Student:

def \_\_init\_\_(self, fio, group):

self.\_fio = fio

self.\_group = group

self.\_lect\_marks = [] # оценки за лекции

self.\_house\_marks = [] # оценки за домашние задания

def add\_lect\_marks(self, mark):

self.\_lect\_marks.append(mark)

def add\_house\_marks(self, mark):

self.\_house\_marks.append(mark)

def \_\_str\_\_(self):

return f"Студент {self.\_fio}: оценки на лекциях: {str(self.\_lect\_marks)}; оценки за д/з: {str(self.\_house\_marks)}"

class Mentor:

def \_\_init\_\_(self, fio, subject):

self.\_fio = fio

self.\_subject = subject

# здесь продолжайте программу