

Задание по программированию: Создание адаптера для класса

✓ Зачет - 1/1 баллов

Срок сдачи Сдайте это задание до 18 окт. г., 9:59 MSK

ИнструкцииМоя работаОбсуждения

Вы продолжаете писать игру, и настало время разобраться с расчетом освещенности на карте. Так как это не совсем тривиальная задача, вы хотели бы использовать готовое решение, а не писать свое собственное. Вам удалось найти готовый класс, который решает задачу, однако интерфейс этого класса не совместим с вашей игрой.

Вам нужно написать адаптер, который позволил бы использовать найденный вами класс совместно с вашей системой.

Интерфейс класса выглядит следующим образом:

```
1 class Light:
2     def __init__(self, dim):
3         self.dim = dim
4         self.grid = [[0 for i in range(dim[0])] for _ in range(dim[1])]
5         self.lights = []
6         self.obstacles = []
7
8     def set_dim(self, dim):
9         self.dim = dim
10        self.grid = [[0 for i in range(dim[0])] for _ in range(dim[1])]
11
12    def set_lights(self, lights):
13        self.lights = lights
14        self.generate_lights()
15
16    def set_obstacles(self, obstacles):
17        self.obstacles = obstacles
18        self.generate_lights()
19
20    def generate_lights(self):
21        return self.grid.copy()
22
```

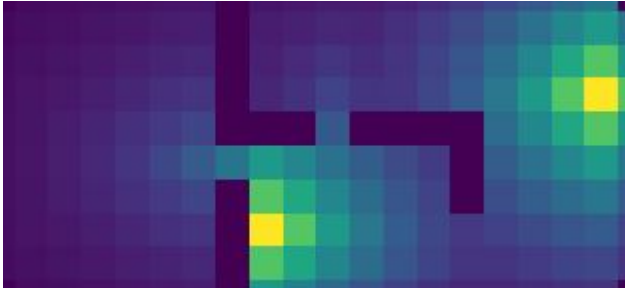
Интерфейс системы выглядит следующим образом:

```
1 class System:
2     def __init__(self):
3         self.map = self.grid = [[0 for i in range(30)] for _ in range(20)]
4         self.map[5][7] = 1 # Источники света
5         self.map[5][2] = -1 # Стена
6
7     def get_lightening(self, light_mapper):
8         self.lightmap = light_mapper.lighten(self.map)
9
10    ...

```

Класс Light создает в методе __init__ поле заданного размера. За размер поля отвечает параметр, представляющий из себя кортеж из 2 чисел. Элемент dim[1] отвечает за высоту карты, dim[0] за ее ширину. Метод set_lights устанавливает массив источников света с заданными координатами и просчитывает освещение. Метод set_obstacles устанавливает препятствия аналогичным образом. Положение элементов задается списком кортежей. В каждом элементе кортежа хранятся 2 значения: elem[0] – координата по ширине карты и elem[1] – координата по высоте соответственно. Метод generate_lights рассчитывает освещенность с учетом источников и препятствий.

Пример карты освещенности, полученной этим методом изображен на следующем рисунке:



В системе в конструкторе создается двумерная, карта, на которой источники света обозначены как 1, а препятствия как -1. Метод get_lightening принимает в качестве аргумента объект, который должен посчитать освещение. У объекта вызывается метод lighten, который принимает карту объектов и источников света и возвращает карту освещенности.

Вам необходимо написать адаптер MappingAdapter. Прототип класса вам дан в качестве исходного кода.

```
1 class MappingAdapter:
2     def __init__(self, adaptee):
3         pass
4
5     def lighten(self, grid):
6         pas

```

Список частых ошибок, совершаемых студентами, при решении данного задания.

1. Решение содержит дополнительный или отладочный код. Решение должно содержать только реализацию класса **MappingAdapter**.

2. Использование сторонних библиотек (например, **numpy**) приводит к ошибкам импорта. В задании можно использовать только модули стандартной библиотеки **Python**.

3. Изменение сигнатуры метода **lighten** класса **MappingAdapter**. Из описания задания -

"У объекта вызывается метод lighten, который принимает карту объектов и источников света и ВОЗВРАЩАЕТ карту освещенности".

Некоторые студенты пишут реализацию класса **MappingAdapter** таким образом, что вызов метода **lighten** изменяет значения атрибутов, определенных внутри класса.

4. Игнорирование различий в способах хранения данных в системе и классе Light. Из описания задания -

"Класс Light создает в методе __init__ поле заданного размера. За размер поля отвечает параметр, представляющий из себя кортеж из 2 чисел. Элемент dim[1] отвечает за высоту карты, dim[0] за ее ширину. Положение элементов задается списком кортежей. В каждом элементе кортежа хранятся 2 значения: elem[0] – координата по ширине карты и elem[1] – координата по высоте соответственно...."

Естественное представление двумерного массива в виде вложенных списков, как правило такое, что первый индекс отвечает за высоту, а второй за ширину. То есть при обращении к массиву **arr[5][3]** - вернется третий элемент из 5-го вложенного списка. В классе **Light**, исходя из условия задания, индексация в массиве получается "перевернутой". Поэтому это нужно учитывать при создании dim и списков с координатами источников света и препятствий. Не учет этого момента приводит к ошибке трейдэка: **Тест 3. При попытке рассчитать освещение выбрасывается исключение IndexError**.

5. Использование «map» в качестве имени переменной для хранения объекта карты. В стандартной библиотеке **Python** есть функция с таким же названием. Называя переменную так, вы «затираете» встроенную функцию. В данной задаче эта ошибка возможно не приведет к side-эффектам, но такая привычка может дорого обойтись в реальном проекте.