

```

1 """Урок 7 Задание 2
2 Реализовать проект расчета суммарного расхода ткани
3 на производство одежды. Основная
4 сущность (класс) этого проекта – одежда , которая
5 может иметь определенное название. К
6 типам одежды в этом проекте относятся пальто и костюм
7 . У этих типов одежды существуют
8 параметры: размер (для пальто ) и рост (для костюма
9 ). Это могут быть обычные числа: V и
10 H , соответственно.
11 Для определения расхода ткани по каждому типу одежды
12 использовать формулы: для пальто
13 (V/6.5 + 0.5) , для костюма (2*H + 0.3) . Проверить
14 работу этих методов на реальных данных.
15 Реализовать общий подсчет расхода ткани. Проверить на
16 практике полученные на этом уроке
17 знания: реализовать абстрактные классы для основных
18 классов проекта, проверить на
19 практике работу декоратора @property .
20 """
21
22
23
24 from abc import ABC, abstractmethod
25
26
27 class Outfit(ABC):
28     def __init__(self, value):
29         self.value = value
30
31     @abstractmethod
32     def consumption(self):
33         pass
34
35     def __add__(self, other):
36         return self.consumption + other.consumption
37
38
39 class Coat(Outfit):
40     @property
41     def consumption(self):
42         print(f'Coat.consumption - {round(self.value
43             / 6.5 + 0.5, 2)}')
44         return round(self.value / 6.5 + 0.5, 2)
45
46
47 class Suit(Outfit):
48     @property
49     def consumption(self):

```

```

36         print(f'Coat.consumption - {round(self.value
    * 2 + 0.3, 2)}')
37         return round(self.value * 2 + 0.3, 2)
38
39 a = Coat(int(input('Coat size - ')))
40 b = Suit(int(input('Suit height - ')))
41
42 print(a + b)
43
44
45
46 # with setter method
47 class Clothes(ABC):
48     def __init__(self):
49         pass
50
51     @property
52     @abstractmethod
53     def raschet(self):
54         pass
55
56     def __add__(self, other):
57         return self.raschet + other.raschet
58
59
60 class Coat(Clothes):
61     def __init__(self, size):
62         super().__init__()
63         self.size = size
64
65     @property
66     def size(self):
67         return self.__size
68
69     @size.setter
70     def size(self, size):
71         if size < 40:
72             print('Min size - 40')
73             self.__size = 40
74         elif size > 60:
75             print('Max size - 60')
76             self.__size = 60
77         else:
78             self.__size = size

```

```

79
80     @property
81     def raschet(self):
82         return self.__size / 6.5 + 0.5
83
84
85 class Costume(Clothes):
86     def __init__(self, height):
87         super().__init__()
88         self.height = height
89
90     @property
91     def height(self):
92         return self.__height
93
94     @height.setter
95     def height(self, height):
96         if height < 100:
97             print('Min height - 100')
98             self.__height = 150
99         elif height > 240:
100             print('Max height - 240')
101             self.__height = 240
102         else:
103             self.__height = height
104
105     @property
106     def raschet(self):
107         return 2 * (self.__height / 100) + 0.3
108
109
110 coat_1 = Coat(int(input('Enter coat size - ')))
111 print(f'You need - {coat_1.raschet:.2f}')
112 costume_1 = Costume(int(input('Enter suit size - '
113 )))
114 print(f'You need - {costume_1.raschet:.2f}')
115 print(f'Total consumption - {coat_1 + costume_1:.2f}
116 ')

```