

Coverage for **madera.py** : 83%

54 statements   45 run   9 missing   0 excluded

```
1 import math
2 import sys
3
4
5 def selector(producto, ancho, largo, alto=None, clase=None, piezas=None):
6     if producto == "raja":
7         raja(ancho, largo, alto)
8     elif producto == "aserrio":
9         aserrio(ancho, alto, largo, clase)
10    elif producto == "troncos":
11        troncos(ancho, largo, piezas)
12    elif producto == "aserrin" or producto == "astilla":
13        celulosas(producto, ancho, largo, alto)
14
15
16 def raja(ancho, largo, altos):
17     """
18     Calcula el volumen de una carga de raja, promediando 3 medidas de la
19     altura y usando un factor de conversion
20
21     :param ancho: ancho de la carga, en metros
22     :param largo: largo de la carga, en metros
23     :param altos: lista con n alturas, en metros
24     :return: tupla con volumen de carga y precio de la misma
25     """
26     alto = 0
27     for a in altos:
28         alto += a
29     alto = alto / len(alto)
30     volumen = ancho * alto * largo
31     vol_str = "{0:.3f}".format(volumen)
32     precio = volumen * 0.65 * 320
33     precio_str = "{0:.2f}".format(precio)
34     print(float(vol_str), float(precio_str))
35     return float(vol_str), float(precio_str)
36
37
38 def aserrio(ancho, alto, largo, clase):
39     """
40     Calcula precio de una pieza de madera aserrada (tabla, polin, viga) de
41     acuerdo a su clasificacion de calidad (primera, segunda o tercera)
42     :param ancho: ancho de la pieza, en pulgadas
43     :param alto: alto de la pieza, en pulgadas
44     :param largo: largo de la pieza, en pies
45     :param clase: calidad de la pieza (1, 2 o 3), determina el precio
46     :return: string con descripcion y precio de la pieza
47     """
48     volumen = ancho * largo * alto / 12
49     if clase == 1:
50         precio = volumen * 15
51     if clase == 2:
52         precio = volumen * 12
53     if clase == 3:
54         precio = volumen * 10
55     nombre = '{0} x {1} x {2} {3}a, ${4}'.format(str(ancho), str(alto),
```

```
56                                     str(largo), str(clase),
57                                     str(precio))
58     print nombre
59     return nombre
60
61
62 def troncos(diametro, longitud, piezas=1):
63     """
64     Calcula el volumen de troncos de madera
65     :param diametro: diametro de la pieza, en centimetros
66     :param longitud: longitud de la pieza, en metros
67     :param piezas: cantidad de piezas que comparten diametro y longitud para
68     calcular sus volúmenes
69     :return: volumen del total de troncos calculados
70     """
71     radio_metros = (float(diametro) / 100) / 2
72     area = math.pi * (radio_metros ** 2)
73     volumen = area * longitud * piezas
74     vol_str = "{0:.3f}".format(volumen)
75     print vol_str
76     return float(vol_str)
77
78
79 def celulosas(producto, ancho, largo, alto):
80     """
81     Calcula volumen y costo de cargas de productos para celulosa,
82     como aserrin o astilla
83     :param producto: tipo de producto [aserrin, astilla] para definir precio
84     :param ancho: ancho de la carga, en metros
85     :param largo: largo de la carga, en metros
86     :param alto: longitud de la carga, en metros
87     :return: tupla que contiene volumen calculado y precio del mismo
88     """
89     volumen = ancho * largo * alto
90     if producto == "astilla":
91         precio = volumen * 240
92     elif producto == "aserrin":
93         precio = volumen * 180
94     else:
95         msj = "Producto incorrecto"
96         print msj
97         return msj
98     vol_str = "{0:.3f}".format(volumen)
99     print(vol_str, precio)
100    return float(vol_str), precio
101
102
103 if __name__ == "__main__":
104     selector(sys.argv[1], sys.argv[2], sys.argv[3], sys.argv[4], sys.argv[5])
```