

Class Design Strategy InPython

Watch Class

```
1
2
3
4 """
5     @Author:Kunal Narkhede
6     @Date:21/12/2023
7     @Goal:To implement class Watch
8     Capture Real Life Product on Amazon
9     http://surl.li/omrxz
10 """
11
12 import sys
13 class ProductDimension:
14     """
15     this class implement the Dimension of watch
16     @__init__(self, length: float, width: float, height: float, weight: float):
17     Constructor
18     @get_length(self)
19     getter of attribute length
20     @get_width(self)
21     getter of attribute width
22     @get_height(self)
23     getter of attribute height
24     @get_weight(self)
25     getter of attribute weight
26     -----
27     @set_length(self):
28     setter of attribute length
29     @set_width(self):
30     setter of attribute width
31     @set_height(self):
32     setter of attribute height
33     @set_weight(self):
34     setter of attribute weight
```

```
35  """
36
37  def __init__(
38      self,
39      length:float,
40      width:float,
41      height:float,
42      weight:float
43  ):
44
45  """
46      Constructor of ProductDimension class:
47      @__init__(self, length: float, width: float, height: float, weight: float):
48
49      @self:newly created class object of ProductDimension
50      @length:Client specified value for attribute length
51      @width:Client specified value for attribute width
52      @height:Client specified value for attribute height
53      @weight:Client specified value for attribute weight
54
55  """
56  if type(length)!=float:
57      raise TypeError("Bad type:length")
58  if type(width)!=float:
59      raise TypeError("Bad type:width")
60  if type(height)!=float:
61      raise TypeError("Bad type:height")
62  if type(weight)!=float:
63      raise TypeError("Bad type:weight")
64  if length<=0.0:
65      raise ValueError("Length must be positive")
66  if width<=0.0:
67      raise ValueError("Width must be positive")
68  if height<=0.0:
69      raise ValueError("Height must be positive")
```

```
70     if weight<=0.0:
71         raise ValueError("Weight must be positive")
72
73     self.length=length
74     self.width=width
75     self.height=height
76     self.weight=weight
77
78     #getter method
79
80     def get_length(self) -> float:
81         """
82         Returns the length attribute of the calling object
83         """
84         return self.length
85
86     def get_width(self) -> float:
87         """
88         Returns the width attribute of the calling object
89         """
90         return self.width
91
92     def get_height(self) -> float:
93         """
94         Returns the height attribute of the calling object
95         """
96         return self.height
97
98     def get_weight(self) -> float:
99         """
100         Returns the weight attribute of the calling object
101         """
102         return self.height
103
104     #setter method
```

```
105
106 def set_length(self,length:float):
107     """
108     Sets the length attribute of the calling object to @new_length
109     Before setting, TypeCheck and ValueCheck is performed.
110     """
111     if type(length)!=float:
112         raise TypeError("new_length must be an float")
113     if length <= 0.0:
114         raise TypeError("new_length must be positive")
115     self.length=length
116
117 def set_width(self,width:float):
118     """
119     Sets the width attribute of the calling object to @new_width
120     Before setting, TypeCheck and ValueCheck is performed.
121     """
122     if type(width)!=float:
123         raise TypeError("new_width must be an float")
124     if width <= 0.0:
125         raise ValueError("new_width must be positive")
126     self.width=width
127
128 def set_height(self,height:float):
129     """
130     Sets the height attribute of the calling object to @new_height
131     Before setting, TypeCheck and ValueCheck is performed.
132     """
133     if type(height)!=float:
134         raise TypeError("new_height must be an float")
135     if height <= 0.0 :
136         raise ValueError("new_height must be positive")
137     self.height=height
138
139 def set_weight(self,weight:float):
```

```

140     """
141     Sets the weight attribute of the calling object to @new_weight
142     Before setting, TypeCheck and ValueCheck is performed.
143     """
144     if type(weight)!=float:
145         raise TypeError("new_weight must be an float")
146     if weight <= 0.0:
147         raise ValueError("new_weight must be positive")
148
149 class Watch:
150     """
151     This class is implement the characterstics of Watch
152     __init__(self,
153             wtch_brand:str,
154             wtch_manifaturer:str,
155             wtch_model:str,
156             wtch_prod_dimensions:ProductDimension,
157             wtch_nr_batteries:int,
158             wtch_nr_model:str,
159             wtch_features:[str],
160             wtch_dsp_type:str,
161             wtch_are_batteries_included:bool,
162             wtch_are_batteries_required:bool,
163             wtch_country_of_origin:str
164     Constructor
165     @get_wtch_brand(self)
166     getter of attribute wtch_brand
167     @get_wtch_manifaturer(self)
168     getter of attribute wtch_manifaturer
169     @get_wtch_model(self)
170     getter of attribute wtch_model
171     @get_wtch_prod_dimensions(self)
172     getter of attribute wtch_prod_dimensions
173     @get_wtch_nr_batteries(self)
174     getter of attribute wtch_nr_batteries
175     @get_wtch_nr_model(self)

```

```
176     getter of attribute wtch_nr_model
177     @get_wtch_feature(self)
178     getter of attribute wtch_feature
179     @get_wtch_dsp_type(self)
180     getter of attribute wtch_dsp_type
181     @get_wtch_are_batteries_included(self)
182     getter of attribute wtch_are_batteries_included
183     @get_wtch_are_batteries_required(self)
184     getter of attribute wtch_are_batteries_required
185     -----
186     @set_wtch_brand(self)
187     setter of attribute wtch_brand
188     @set_wtch_manifaturer(self)
189     setter of attribute wtch_manufacturer
190     @set_wtch_model(self)
191     setter of attribute wtch_model
192     @set_wtch_prod_dimensions(self)
193     setter of attribute wtch_prod_dimensions
194     @set_wtch_nr_batteries(self)
195     setter of attribute wtch_nr_batteries
196     @set_wtch_nr_model(self)
197     setter of attribute wtch_nr_model
198     @set_wtch_feature(self)
199     setter of attribute wtch_feature
200     @set_wtch_dsp_type(self)
201     setter of attribute wtch_dsp_type
202     @set_wtch_are_batteries_included(self)
203     setter of attribute wtch_are_batteries_included
204     @set_wtch_are_batteries_required(self)
205     setter of attribute wtch_are_batteries_required
206
207     """
208     def __init__(self,
209                 wtch_brand:str,
210                 wtch_manufacturer:str,
```

```

211         wtch_model:str,
212         wtch_prod_dimensions:ProductDimension,
213         wtch_nr_batteries:int,
214         wtch_nr_model:str,
215         wtch_features:[str],
216         wtch_dsp_type:str,
217         wtch_are_batteries_included:bool,
218         wtch_are_batteries_required:bool,
219         wtch_country_of_origin:str
220     ):
221
222     """
223     @self:newly created class object of ProductDimension
224     @wtch_brand:Client specified value for attribute wtch_brand
225     @wtch_manufacturer:Client specified value for attribute wtch_manufacturer
226     @wtch_model:Client specified value for attribute wtch_model
227     @wtch_prod_dimensions:Client specified value for attribute wtch_prod_dimensions
228     @wtch_nr_batteries:Client specified value for attribute wtch_nr_batteries
229     @wtch_nr_model:Client specified value for attribute wtch_nr_model
230     @wtch_features:Client specified value for attribute wtch_features
231     @wtch_dsp_type:Client specified value for attribute wtch_dsp_type
232     @wtch_are_batteries_included:Client specified value for attribute wtch_are_batteries_included
233     @wtch_are_batteries_required:Client specified value for attribute wtch_are_batteries_required
234     @wtch_country_of_origin:Client specified value for attribute wtch_country_of_origin
235     """
236
237     if type(wtch_brand)!=str:
238         raise TypeError("Bad type:brand")
239     if type(wtch_manufacturer)!=str:
240         raise TypeError("Bad type:manufacturer")
241     if type(wtch_model)!=str:
242         raise TypeError("Bad type:model")
243     if type(wtch_prod_dimensions)!=ProductDimension:
244         raise TypeError("Bad type:Product Dimension")
245     if type(wtch_nr_batteries)!=int:

```

```
246         raise TypeError("Bad type:number of batteries")
247     if type(wtch_nr_model)!=str:
248         raise TypeError("Bad type:Watch Model")
249     if ' __iter__ ' not in dir(type(wtch_features)):
250         raise TypeError("Bad type:watch feature must iterable")
251     for feature in wtch_features:
252         if type(feature)!=str:
253             raise TypeError("Bad type:feature")
254     if type(wtch_dsp_type)!=str:
255         raise TypeError("Bad type:Display type")
256     if type(wtch_are_batteries_included)!=bool:
257         raise TypeError("Bad type:are_batteries_included")
258     if type(wtch_are_batteries_required)!=bool:
259         raise TypeError("Bad type:are_batteries_required")
260     if type(wtch_country_of_origin)!=str:
261         raise TypeError("Bad type:country of origin")
262     self.wtch_brand=wtch_brand
263     self.wtch_manufacturer=wtch_manufacturer
264     self.wtch_model=wtch_model
265     self.wtch_prod_dimensions=wtch_prod_dimensions
266     self.wtch_nr_batteries=wtch_nr_batteries
267     self.wtch_nr_model=wtch_nr_model
268     self.wtch_feature=wtch_features
269     self.wtch_dsp_type=wtch_dsp_type
270     self.wtch_are_batteries_included=wtch_are_batteries_included
271     self.wtch_are_batteries_required=wtch_are_batteries_required
272     self.wtch_country_of_origin=wtch_country_of_origin
273
274     #getter method
275
276     def get_wtch_brand(self)->str:
277         """
278         Returns the wtch_brand attribute of the calling object
279         """
280         return self.wtch_brand
```



```
281
282 def get_wtch_manufacturer(self)->str:
283     """
284     Returns the wtch_manufacturer attribute of the calling object
285     """
286     return self.wtch_manufacturer
287
288 def get_wtch_model(self)->str:
289     """
290     Returns the wtch_model attribute of the calling object
291     """
292     return self.wtch_model
293
294 def get_wtch_prod_dimensions(self)->ProductDimension:
295     """
296     Returns the wtch_prod_dimensions attribute of the calling object
297     """
298     return self.wtch_prod_dimensions
299
300 def get_wtch_nr_batteries(self)->int:
301     """
302     Returns the wtch_nr_batteries attribute of the calling object
303     """
304     return self.wtch_nr_batteries
305
306 def get_wtch_nr_model(self)->str:
307     """
308     Returns the wtch_nr_model attribute of the calling object
309     """
310     return self.wtch_nr_model
311
312 def get_wtch_feature(self)->[str]:
313     """
314     Returns the wtch_feature attribute of the calling object
315     """
```

```
316     return self.wtch_feature
317
318     def get_wtch_dsp_type(self)->str:
319         """
320         Returns the wtch_dsp_type attribute of the calling object
321         """
322         return self.wtch_dsp_type
323
324     def get_wtch_are_batteries_included(self)->bool:
325         """
326         Returns the wtch_are_batteries_included attribute of the calling object
327         """
328         return self.wtch_are_batteries_included
329
330     def get_wtch_are_batteries_required(self)->bool:
331         """
332         Returns the wtch_are_batteries_required attribute of the calling object
333         """
334         return self.wtch_are_batteries_required
335
336     def get_wtch_country_of_origin(self)->str:
337         """
338         Returns the wtch_country_of_origin attribute of the calling object
339         """
340         return self.wtch_country_of_origin
341
342     #setter method
343
344     def set_wtch_brand(self,new_wtch_brand:str)->None:
345         """
346         Sets the wtch_brand attribute of the calling object to @new_length
347         Before setting, TypeCheck is performed.
348         """
349         if type(new_wtch_brand)!=str:
350             raise TypeError("new_wtch_brand must be str")
```

```

351     self.wtch_brand=new_wtch_brand
352
353 def set_wtch_manufacturer(self,new_wtch_manufacturer:str)->None:
354     """
355     Sets the wtch_manufacturer attribute of the calling object to @new_wtch_manufacturer
356     Before setting, TypeCheck is performed.
357     """
358     if type(new_wtch_manufacturer)!=str:
359         raise TypeError("new_wtch_manufacturer must be str")
360     self.wtch_manufacturer=new_wtch_manufacturer
361
362 def set_wtch_model(self,new_wtch_model:str)->None:
363     """
364     Sets the wtch_model attribute of the calling object to @new_wtch_model
365     Before setting, TypeCheck is performed.
366     """
367     if type(new_wtch_model)!=str:
368         raise TypeError("new_wtch_model must be str")
369     self.wtch_model=new_wtch_model
370
371 def set_wtch_prod_dimensions(self,new_wtch_prod_dimensions:ProductDimension)->None:
372     """
373     Sets the wtch_prod_dimensions attribute of the calling object to @new_wtch_prod_dimensions
374     Before setting, TypeCheck is performed.
375     """
376     if type(new_wtch_prod_dimensions)!=ProductDimension:
377         raise TypeError("new_wtch_prod_dimensions must be ProductDimension")
378     self.wtch_prod_dimensions=new_wtch_prod_dimensions
379
380 def set_wtch_nr_batteries(self,new_wtch_nr_batteries:int)->None:
381     """
382     Sets the wtch_nr_batteries attribute of the calling object to @new_wtch_nr_batteries
383     Before setting, TypeCheck and ValueCheck is performed.
384     """
385     if type(new_wtch_nr_batteries)!=int:

```

```

386         raise TypeError("new_wtch_nr_batteries must be int")
387     if new_wtch_nr_batteries<=0:
388         raise ValueError("number of batteries must be positive")
389     self.wtch_nr_batteries=new_wtch_nr_batteries
390
391 def set_wtch_nr_model(self,new_wtch_nr_model:str)->None:
392     """
393     Sets the wtch_nr_model attribute of the calling object to @new_wtch_nr_model
394     Before setting, TypeCheck is performed.
395     """
396     if type(new_wtch_nr_model)!=str:
397         raise TypeError("new_wtch_nr_model must be str")
398     self.wtch_nr_model=new_wtch_nr_model
399
400 def set_wtch_features(self,new_wtch_feature:[str])->None:
401     """
402     Sets the wtch_feature attribute of the calling object to @new_wtch_feature
403     Before setting, TypeCheck is performed.
404     """
405     if '__iter__' not in dir(type(new_wtch_feature)):
406         raise TypeError("new_wtch_feature must itetable")
407     for feature in new_wtch_feature:
408         if type(feature)!=str:
409             raise TypeError("new_wtch_feature must be str")
410     self.wtch_feature=new_wtch_feature
411
412 def set_wtch_dsp_type(self,new_wtch_dsp_type:str)->None:
413     """
414     Sets the wtch_dsp_type attribute of the calling object to @new_wtch_dsp_type
415     Before setting, TypeCheck is performed.
416     """
417     if type(new_wtch_dsp_type)!=str:
418         raise TypeError("new_wtch_dsp_type must be str")
419     self.wtch_dsp_type=new_wtch_dsp_type
420

```

```

421     def set_wtch_are_batteries_included(self,new_wtch_are_batteries_included:bool)->None:
422         """
423         Sets the wtch_are_batteries_included attribute of the calling object to
424         @new_wtch_are_batteries_included
425         Before setting, TypeCheck is performed.
426         """
427         if type(new_wtch_are_batteries_included!=bool):
428             raise TypeError("new_wtch_are_batteries_included must be bool")
429         self.wtch_are_batteries_included=new_wtch_are_batteries_included
430
431     def set_wtch_are_batteries_required(self,new_wtch_are_batteries_required:bool)->None:
432         """
433         Sets the wtch_are_batteries_required attribute of the calling object to
434         @new_wtch_are_batteries_required
435         Before setting, TypeCheck is performed.
436         """
437         if type(new_wtch_are_batteries_required!=bool):
438             raise TypeError("new_wtch_are_batteries_required must be bool")
439         self.wtch_are_batteries_required=new_wtch_are_batteries_required
440
441     def set_wtch_country_of_origin(self,new_wtch_country_of_origin:str)->None:
442         """
443         Sets the wtch_country_of_origin attribute of the calling object to @new_wtch_country_of_origin
444         Before setting, TypeCheck is performed.
445         """
446         if type(new_wtch_country_of_origin!=str):
447             raise TypeError("new_wtch_country_of_origin must be str")
448         self.wtch_country_of_origin=new_wtch_country_of_origin
449
450     def show(self)->None:
451
452         """
453         This function display all the characterstics of Watch class
454         """
455
456         print("Watch Brand:{}".format(self.wtch_brand))

```

```
457     print("Watch Manifaturer:{}".format(self.wtch_manufacturer))
458     print("Watch Model:{}".format(self.wtch_model))
459     print("Watch Product Dimension:{}".format(self.wtch_prod_dimensions))
460     print("Watch Number of batteries:{}".format(self.wtch_nr_batteries))
461     print("Watch Model number:{}".format(self.wtch_nr_model))
462     print("Watch Spacial feature:{}".format(self.wtch_feature))
463     print("Watch Display type:{}".format(self.wtch_dsp_type))
464     print("Watch Batteries included:{}".format(self.wtch_are_batteries_included))
465     print("Watch Batteries required:{}".format(self.wtch_are_batteries_required))
466     print("Watch Country of origin:{}".format(self.wtch_country_of_origin))
467
468
469 def main():
470     Wtch_obj=Watch(
471         "Seiko",
472         "Seiko",
473         "SSK003K1",
474         ProductDimension(30.5,30.5,30.5,250.0),
475         1,
476         "SSK003K1",
477         ["Calculator","Dual Time","Glow in the dark","Water Resistant"],
478         "Analog",
479         False,
480         False,
481         "China"
482     )
483     Wtch_obj.show()
484     #we can also get the attribute using getter method and
485     #set the specific attribute using setter method
486     sys.exit(0)
487 main()
488
```