



Problema: Game Score

```
RANK SCORE
                   NAME
           012000
                    AKI
           009000
                    CHI
      2ND
      3RD
           008000
                    SEI
      4TH
           005400
                    NAO
      5 T H
           003200
                    IYO
NINTENDO
                     CREDIT 0
```





Passo 1: Classe ItemPlacar

- * Implemente uma classe para representar um Item do placar.
- * Cada item possui o nome do jogador e o placar obtido
- Faça o método __str__ que retorna a linha do placar formatada

* Utilize encapsulamento!





Passo 1: Classe ItemPlacar

```
class ItemPlacar:
    def __init__(self, nome, placar):
        self._nome = nome
        self._placar = placar
   @property
                                           def str (self):
    def nome(self):
                                               return f"{self. nome:10}\t{self. placar:5}"
        return self. nome
   @nome.setter
    def nome(self, valor):
        self. nome = valor
   @property
    def placar(self):
        return self. placar
   @placar.setter
    def placar(self, valor):
        self. placar = valor
```



Passo 2: Classe Placar

- * Na classe Placar recebemos como parâmetro o tamanho máximo do placar (Utilize como valor padrão 5).
- * No construtor dessa classe, crie uma lista vazia para armazenar os itens de score
- * Defina o método __str__ dessa classe, que retorna todo o placar
- * Pense em como fazer a implementação do método para inserir um item no placar:



def inserir_item(self, item: ItemPlacar):



```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```



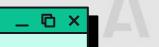


```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

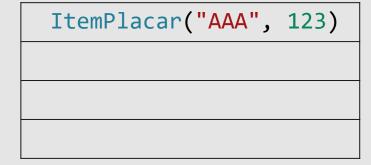
ItemPlacar("AAA", 123)







```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```





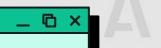


```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("BBB", 234)

ItemPlacar("AAA", 123)





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("BBB", 234)

ItemPlacar("AAA", 123)

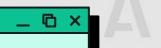




```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("BBB", 234)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("CCC", 200)
```

```
ItemPlacar("BBB", 234)

ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("CCC", 200)

ItemPlacar("BBB", 234)

ItemPlacar("AAA", 123)





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("DDD", 100)
```

```
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
ItemPlacar("DDD", 100)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("EEE", 400)
```

```
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
ItemPlacar("DDD", 100)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("EEE", 400)
```

```
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("EEE", 400)
```

```
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)

ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("EEE", 400)

ItemPlacar("BBB", 234)

ItemPlacar("CCC", 200)

ItemPlacar("AAA", 123)





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("EEE", 400)
```

```
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("FFF", 99)
```

```
ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("GGG", 150)
```

```
ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("GGG", 150)
```

```
ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
```





```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

```
ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("GGG", 150)
```





* Seria mais fácil implementar esse método utilizando as funções de lista!

```
def inserir_item(self, item: ItemPlacar):
          self._scores.append(item)
          self._scores.sort(reverse=True)
          if len(self._scores) > self._max_items:
                self._scores = self._scores[:self._max_items]
```

* Porém, o método sort não sabe como comparar dois ItemPlacar!





- * O método sort utiliza os operadores de comparação (>, <, ==, !=, etc.) para comparar objetos
- * Podemos dizer como esses operadores se comportam na nossa classe sobrescrevendo certas funções mágicas:

a < b	alt(b)
$a \mathrel{<=} b$	ale(b)
a>b	agt(b)
a >= b	age(b)
a == b	aeq(b)
a != b	ane(b)



_ © ×

```
def lt (self, outroItem):
    return self. placar < outroItem. placar</pre>
def gt (self, outroItem):
    return self. placar > outroItem. placar
                                                     p1 = ItemPlacar("AAA", 123)
def eq (self, outroItem):
    return self. placar == outroItem. placar
                                                     p2 = ItemPlacar("ZZZ", 123)
def le (self, outroItem):
                                                     print(p1 == p2)
    return self. placar <= outroItem. placar
                                                     print(p1 > p2)
                                                     print(p1 <= p2)</pre>
def ge (self, outroItem):
    return self. placar >= outroItem. placar
def ne (self, outroItem):
   return self. placar != outroItem. placar
```





Sobrecarga de Operadores

 Podemos fazer nossa classe Placar se comportar como uma lista sobrescrevendo alguns métodos

v in a	acontains(v)
a[k]	agetitem(k)
a[k] = v	asetitem(k,v)
del a[k]	adelitem(k)
len(a)	alen()





Outros Métodos

* Operadores Matemáticos:

a + b	aadd(b);	alternatively bradd(a)
a — b	asub(b);	alternatively brsub(a)
a * b	amul(b);	alternatively brmul(a)
a / b	atruediv(b);	alternatively brtruediv(a)
a // b	afloordiv(b);	alternatively brfloordiv(a)
a % b	amod(b);	alternatively brmod(a)
a ** b	apow(b);	alternatively brpow(a)





Outros Métodos

* Conversões para tipos básicos do Python:

bool(a)	abool()
float(a)	afloat()
int(a)	aint()





Sobrecarga de Operadores

- * A sobrecarga de operadores costuma ser utilizada em três ocasiões especiais:
 - → Para emular tipos numéricos que não existem no Python
 - → Converter um objeto para um tipo básico do Python (str, int, etc.)
 - → Implementar coleções

