

# GABARITO AP2 1º SEMESTRE DE 2017.

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância  
Curso de Tecnologia em Sistemas de Computação  
Disciplina: Programação com Interfaces Gráficas  
Professores: Mario Benevides e Paulo Roma

## 1 Questão

Código:

```
#!/usr/bin/env python
# coding: utf-8

class Intervalo(object):
    """Representa um intervalo de tempo [a:b]"""

    def __init__(self,a,b):
        """Constrói um intervalo [a:b]"""
        if a >= b:
            raise ValueError ('Intervalo Inválido')
        self.a, self.b = a,b

    def contem(self,t):
        """Retorna True se e somente se o instante de tempo
        t está contido no intervalo."""
        return t >= self.a and t <= self.b

    def intersecta(self,outro):
        """Retorna True se este intervalo tem algum
        instante de tempo em comum com o intervalo outro"""
        return max(self.a,outro.a) <= min(self.b,outro.b)

    def __repr__(self):
        """Retorna uma representação string deste objeto"""
        return "[" + str(self.a) + ":" + str(self.b) + "]"

    def __cmp__(self,outro):
        """Retorna < 0 se outro começa depois de self,
        > 0 se outro começa antes de self ou
        0 se self começa junto com outro"""
        return self.a - outro.a

    def uniao(self,outro):
```

```

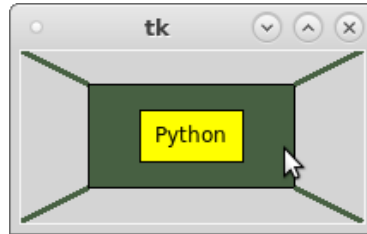
        """Retorna o menor intervalo de tempo que contém
        todos os instantes de self e todos os instantes de outro"""
        return Intervalo(min(self.a,outro.a),max(self.b,outro.b))

    def intersecao(self,outro):
        """Retorna o intervalo de tempo que contem todos os
        instantes comuns a self e a outro. Se não há instantes
        comuns, retorna None. """
        if self.intersecta(outro):
            return Intervalo(max(self.a,outro.a),min(self.b,outro.b))
        return None

i1 = Intervalo(1,3)
i2 = Intervalo(2,3)
i3 = Intervalo(6,7)
print (i1,i2,i3)
print (i1.contem(1.4),i2.contem(4))
print (i1.intersecta(i2),i1.intersecta(i3))
print (i1.uniao(i2),i1.uniao(i3))
print (i1.intersecao(i2), i1.intersecao(i3))

```

## 2 Questão



## 3 Questão

[3, [2, 1, 2], 3]

## 4 Questão

