

# Fundamentos da Programação

## Exemplo dicionários e TADs

### Aula 17

Alberto Abad, Tagus Park, IST, 2021-22

## Exercício Dicionários + TADs: Reserva de salas

Considere que a informação sobre um período de ocupação de uma sala consiste num dia da semana (uma string com os 3 primeiros caracteres do dia), numa hora de início e numa hora de fim (reais), e que esta informação é representada por um dicionário de chaves 'dia', 'inicio' e 'fim'. Por exemplo, o dicionário {'dia': 'seg', 'inicio': 13.5, 'fim': 15} representa o período de segunda-feira das 13h30 às 15h.

Considere ainda que uma sala de aula é representada por um dicionário de chaves 'nome' (string), 'tipo' (1 ou 2), 'capacidade' (inteiro positivo) e 'ocupacao' (tuplo de dicionários representando períodos). Por exemplo,

```
s1 = {'nome' : 'C12', 'tipo': 1, 'capacidade': 50,
      'ocupacao':
      ({'dia': 'seg', 'inicio' : 8, 'fim' : 10},
       {'dia': 'seg', 'inicio' : 13.5, 'fim' : 15},
       {'dia': 'ter', 'inicio' : 8, 'fim' : 10},
       {'dia': 'ter', 'inicio' : 13.5, 'fim' : 15})}
```

```
s2 = {'nome' : 'F8', 'tipo': 1, 'capacidade': 40,
      'ocupacao':
      ({'dia': 'seg', 'inicio' : 10, 'fim' : 12},
       {'dia': 'seg', 'inicio' : 15.5, 'fim' : 17},
       {'dia': 'ter', 'inicio' : 8, 'fim' : 10},
       {'dia': 'ter', 'inicio' : 13.5, 'fim' : 15})}
```

```
In [19]: s1 = {'nome': 'C12', 'tipo': 1, 'capacidade': 50,
              'ocupacao':
              ({'dia': 'seg', 'inicio': 8, 'fim': 10},
               {'dia': 'seg', 'inicio': 13.5, 'fim': 15},
               {'dia': 'ter', 'inicio': 8, 'fim': 10},
               {'dia': 'ter', 'inicio': 13.5, 'fim': 15})}

s2 = {'nome': 'F8', 'tipo': 1, 'capacidade': 40,
      'ocupacao':
      ({'dia': 'seg', 'inicio': 10, 'fim': 12},
       {'dia': 'seg', 'inicio': 15.5, 'fim': 17},
       {'dia': 'ter', 'inicio': 8, 'fim': 10},
       {'dia': 'ter', 'inicio': 13.5, 'fim': 15})}
```

## Exercício Dicionários + TADs: Reserva de salas

- (a) Implemente a função `periodos_compativeis(per1, per2)`, que recebe como argumentos 2 dicionários correspondentes a períodos, e devolve *verdadeiro* apenas se os 2 períodos forem compatíveis, isto é, se não se sobrepuserem. Por exemplo,

```
>>> p1 = {'dia': 'seg', 'inicio': 13.5, 'fim': 15}
>>> p2 = {'dia': 'seg', 'inicio': 14, 'fim': 15}
>>> p3 = {'dia': 'qua', 'inicio': 14, 'fim': 15}
>>> p4 = {'dia': 'qua', 'inicio': 15, 'fim': 16}
>>> periodos_compativeis(p1, p2)
False
>>> periodos_compativeis(p1, p3)
True
>>> periodos_compativeis(p2, p3)
True
>>> periodos_compativeis(p3, p4)
True
```

```
In [29]: p1 = {'dia': 'seg', 'inicio': 13.5, 'fim': 15}
p2 = {'dia': 'seg', 'inicio': 14, 'fim': 15}
p3 = {'dia': 'qua', 'inicio': 14, 'fim': 15}
p4 = {'dia': 'qua', 'inicio': 15, 'fim': 16}

def periodos_compativeis(p1, p2):
    d1, i1, f1 = p1['dia'], p1['inicio'], p1['fim']
    d2, i2, f2 = p2['dia'], p2['inicio'], p2['fim']

    return d1 != d2 or i2 >= f1 or i1 >= f2

periodos_compativeis(p2, p3)
```

Out[29]: True

## Exercício Dicionários + TADs: Reserva de salas

- (b) Implemente a função `sala_esta_livre_periodo(sala, per)`, que recebe como argumentos 2 dicionários correspondentes a uma sala e um período, e devolve *verdadeiro* apenas se a sala `sala` estiver livre no período `per`. Por exemplo, considerando as atribuições anteriores,

```
>>> sala_esta_livre_periodo(s1, p2)
False
>>> sala_esta_livre_periodo(s2, p1)
True
```

```
In [31]: def sala_esta_livre_periodo(sala, periodo):

    for outro in sala['ocupacao']:
        if not periodos_compativeis(outro, periodo):
            return False

    return True

sala_esta_livre_periodo(s2, p1)
```

```
Out[31]: True
```

## Exercício Dicionários + TADs: Reserva de salas

- (c) Considere as seguintes operações básicas para o TAD *período*, para representar um período:

- *Construtor*: `cria_periodo`
- *Seletores*: `obter_dia_periodo`, `obter_inicio_periodo`, `obter_fim_periodo`.

Implemente estas operações, usando a representação interna indicada no início do enunciado.

```
In [33]: def cria_periodo(dia, inicio, fim):
    if dia in ('seg', 'ter', 'qua', 'qui', 'sex', 'sab', 'dom') \
        and isinstance(inicio, (int,float)) and 0 <= inicio < 24 \
        and isinstance(fim, (int,float)) and 0 <= fim < 24:
        return {'dia': dia, 'inicio': inicio, 'fim': fim}

    raise ValueError('argumentos invalidos')

def obter_dia_periodo(periodo):
    return periodo['dia']

def obter_inicio_periodo(periodo):
    return periodo['inicio']

def obter_fim_periodo(periodo):
    return periodo['fim']
```

## Exercício Dicionários + TADs: Reserva de salas

(d) Considere as seguintes operações básicas para o TAD *sala*, para representar a informação de uma sala:

- *Construtor*: `cria_sala`
- *Seletores*: `obter_nome_sala`, `obter_tipo_sala`, `obter_capacidade_sala` e `obter_ocupacao_sala`.

Implemente estas operações, usando a representação interna indicada no início do enunciado.

```
In [23]: def cria_sala(nome, tipo, capacidade, ocupacao):
          return {'nome': nome, 'tipo': tipo, 'capacidade': capacidade, 'ocupacao': ocupacao}

          def obter_nome_sala(sala):
              return sala['nome']

          def obter_tipo_sala(sala):
              return sala['tipo']

          def obter_capacidade_sala(sala):
              return sala['capacidade']

          def obter_ocupacao_sala(sala):
              return sala['ocupacao']
```

## Exercício Dicionários + TADs: Reserva de salas

(e) Implemente de novo as funções das alíneas a), b), usando os TADs definidos nas alíneas c) e d).

```
In [34]: # FAN independente da representacao interna
def periodos_compativeis(p1, p2):
    d1, i1, f1 = obter_dia_periodo(p1), obter_inicio_periodo(p1), obter_fim_periodo(p1)
    d2, i2, f2 = obter_dia_periodo(p2), obter_inicio_periodo(p2), obter_fim_periodo(p2)

    return d1 != d2 or i2 >= f1 or i1 >= f2

def sala_esta_livre_periodo(sala, periodo):
    for outro in obter_ocupacao_sala(sala):
        if not periodos_compativeis(outro, periodo):
            return False

    return True
```

## Abstração de dados

## Tarefas próximas aulas

- Estudar matéria Abstração e Tipos Abstratos de Dados:
  - Completar exemplos
- Nas aulas teóricas --> Ficheiros (capítulo 10 do livro)
- Na aula práticas --> TADs

