

Nome: _____ GABARITO _____ Matr.: _____ Turma Teórica: 2

2ª Prova (Valor: 30%)

1. (8%) Complete as lacunas nas seguintes frases de modo que elas se tornem afirmativas verdadeiras:

- a) A estrutura de dados *dicionário* em Python é um tipo mutável, enquanto que a estrutura tupla não permite que um valor do respectivo tipo possa mudar de valor após ser criado.
- b) A estrutura de dados *dicionário* em Python consiste em uma coleção de pares chave e valor.
- c) A estrutura de dados *arquivo* reside normalmente em memória secundária.
- d) Uma *lista* em Python permite repetição de elementos em sua estrutura, mas a estrutura de dados conjunto não permite.
- e) As estruturas de dados *lista* e *tupla* em Python são ordenadas, ao passo que as estruturas dicionário e conjunto não o são.
- f) Uma *classe* em Python é um tipo de dados que pode ser criado (ou definido) pelo usuário.
- g) O método *construtor* em uma classe em Python, denotado por `__init__`, é importante porque ele cria e inicializa um objeto da classe.
- h) As operações típicas de *arquivo* em Python são: abertura, leitura, escrita e fechamento.
- i) O *caminho relativo* de um arquivo no sistema de arquivos do SO em uso é iniciado no(a) diretório raiz e o *caminho absoluto* é iniciado no(a) diretório corrente.
- j) O identificador `self` em uma definição de classe em Python refere-se ao objeto que se estiver usando no momento.

2. (2%) Faça um círculo na(s) letra(s) correspondente(s) ao que for CORRETO:

- a) A abertura de um arquivo nunca destrói o conteúdo do arquivo em questão.
- b) Um arquivo de saída tem que existir de antemão para que a abertura ocorra com sucesso.
- c) No sistema operacional Windows, o separador de diretórios em um caminho é a barra /.
- ☒ d) O operador `read` de arquivos em Python transfere todo o conteúdo de um arquivo para a memória primária do computador.
- e) O operador `readline` de arquivos em Python transfere todo o conteúdo de um arquivo para a memória primária do computador.

3. (2%) Marque a(s) opção(ões) que possibilita(m) a leitura dos dados de um arquivo e, ao mesmo tempo, também permita que se adicione conteúdo ao final do mesmo, preservando o conteúdo original:

- ☒ a) `fp = open(nomeArquivo, 'r+')`
- ☒ b) `fp = open(nomeArquivo, 'a+')`
- c) `fp = open(nomeArquivo, 'rw')`
- d) `fp = open(nomeArquivo, 'w+')`
- e) `fp = open(nomeArquivo, 'w')`

4. (4%) Um arquivo Python chamado `teste.py` é executado a partir da linha de comando, com a seguinte instrução:

```
python3 teste.py seg ter qua qui sex teste.txt
```

Escreva, após cada comando `print` do código abaixo, qual será a saída em tela correspondente à execução do mesmo.

```
import sys
print(len(sys.argv))
7
print(sys.argv[0])
teste.py
print(sys.argv[len(sys.argv)-1])
teste.txt
print(sys.argv[1][0])
s
```

5. (6%) Complete as linhas de código abaixo **com operações de conjuntos** para que o programa apresente a saída em tela esperada ao final da questão.

```
disciplinas_outros = set()
disciplinas_dpi = set(['inf100', 'inf101', 'inf103', 'inf280'])

alunos_dpi = set()
alunos_ufv = set()

matriculas = {
    'Roberto': ['inf100', 'mat241', 'fis202'],
    'Francisca': ['adm250', 'cco100', 'arq301'],
    'Geraldo': ['inf101', 'inf280', 'fis203', 'fis225'],
    'Hugo': ['inf100', 'let100', 'mat141'],
    'Joana': ['inf103', 'inf101', 'bio131', 'eam451'],
    'Pedro': ['inf103', 'eam432', 'est220'],
    'Carlos': ['adm250', 'adm308', 'cco100'],
    'Bruna': ['inf280', 'fis201', 'fis224'],
    'Paulo': ['fis201', 'mat141', 'fis224', 'mat271']
}

for aluno in matriculas.keys():
    alunos_ufv.add(aluno)
    for disciplina in matriculas[aluno]:
        if disciplina in disciplinas_dpi:
            alunos_dpi.add(aluno)
        else:
            disciplinas_outros.add(disciplina)

disciplinas_ufv = disciplinas_dpi | disciplinas_outros _____
alunos_nao_vinculados = alunos_ufv - alunos_dpi _____

print('%d alunos vinculados ao DPI: ' % len(alunos_dpi), end=' ')
print(alunos_dpi, '\n')
print('%d alunos nao vinculados ao DPI:' % len(alunos_nao_vinculados), end=' ')
print(alunos_nao_vinculados, '\n')
print('%d alunos da UFV:' % len(alunos_ufv), end=' ')
print(alunos_ufv, '\n')
print('%d disciplinas de outros departamentos:' % len(disciplinas_outros),
end=' ')
```

```

print(disciplinas_outros, '\n')
print('%d disciplinas do DPI:' % len(disciplinas_dpi), end=' ')
print(disciplinas_dpi, '\n')
print('%d disciplinas da Ufv:' % len(disciplinas_ufv), end=' ')
print(disciplinas_ufv, '\n')

```

Saída em tela esperada:

```

6 alunos vinculados ao DPI: {'Joana', 'Geraldo', 'Roberto', 'Pedro', 'Hugo',
'Bruna'}

```

```

3 alunos nao vinculados ao DPI: {'Francisca', 'Paulo', 'Carlos'}

```

```

9 alunos da Ufv: {'Joana', 'Geraldo', 'Paulo', 'Roberto', 'Pedro', 'Carlos',
'Francisca', 'Hugo', 'Bruna'}

```

```

17 disciplinas de outros departamentos: {'fis201', 'bio131', 'adm250',
'fis224', 'fis225', 'fis202', 'let100', 'mat271', 'cco100', 'eam451',
'arq301', 'mat141', 'fis203', 'eam432', 'adm308', 'est220', 'mat241'}

```

```

4 disciplinas do DPI: {'inf280', 'inf103', 'inf100', 'inf101'}

```

```

21 disciplinas da Ufv: {'fis201', 'bio131', 'fis224', 'fis202', 'mat271',
'inf103', 'inf100', 'eam451', 'mat141', 'adm308', 'eam432', 'adm250',
'fis225', 'inf280', 'let100', 'cco100', 'arq301', 'fis203', 'inf101',
'est220', 'mat241'}

```

6. (8%) Assinale V (verdadeiro) ou F (falso) nas afirmações a seguir.

(☐ F) Uma função pertencente a uma classe é chamada de método.

(☒ V) Não é permitida a duplicação de nomes de chaves em um dicionário Python.

(☐ F) O modo 'w', na operação de abertura de um arquivo, indica que novo conteúdo será adicionado ao final do mesmo, preservando seu conteúdo atual.

(☒ V) Construtor é um método que é invocado toda vez que se cria um objeto de uma determinada classe.

(☐ F) Quando se tenta realizar a abertura de um arquivo no modo 'w', a operação resulta em erro, caso não exista um arquivo em disco com o nome informado.

(☒ V) Seja uma lista em Python armazenada na variável `lista1`. A operação `s = set(lista1)` tem, como efeito eliminar os valores duplicados originalmente na lista.

- (V) O uso de caminhos relativos para localizar e/ou identificar um determinado recurso (arquivo ou pasta) em disco é vantajoso quando o recurso está em uma localização próxima à do diretório corrente, pois encurta o *string* que identifica tal caminho.
- (F) É permitida a duplicação de itens dentro de um conjunto em Python (*set*).
- (V) Classe é um tipo de estrutura que agrega dados e funções relacionadas ao dados.
- (V) Um atributo de uma classe é qualquer membro desta, seja ele função ou variável.