Programação

2024/2025, 1º semestre

Trabalho Individual

> **Versão**: 1.0, 01/09/2024
>
> **Prazo de entrega**: Sábado, 14/12/2024, às 23:59

ParkGest

A ParkGest é uma empresa responsável pela gestão de múltiplos parques de estacionamento, quer públicos, quer privados. De forma a aumentar a sua eficiência, a empresa pretende desenvolver um sistema de apoio à gestão desses parques de estacionamento. Este sistema deverá ser capaz de armazenar informações sobre o estado atual dos parques e os veículos neles estacionados. Numa reunião com a empresa, foi definido um conjunto de requisitos que devem ser satisfeitos pelo sistema. Com base nesses requisitos, foi definido um conjunto de tarefas, descritas abaixo, que devem ser realizadas de forma a desenvolver um protótipo inicial do sistema.

Observação: Os estudantes podem partilhar e/ou trocar ideias entre si, sobre os trabalhos e/ou resolução dos mesmos. No entanto, o trabalho entregue deve corresponder ao esforço individual de cada estudante. *O projeto é individual, e em nenhum caso deve ser copiado código que será entregue*. A deteção de código copiado será realizada por software especializado bastante sofisticado. Casos de plágio óbvio serão penalizados com a anulação do projeto, o que implica a reprovação à Unidade Curricular (UC). Adicionalmente, a situação será reportada à Comissão Pedagógica da ISTA e ao Conselho Pedagógico do ISCTE, de acordo com a *Política em Caso de Fraude descrita no final deste documento*.

Serão penalizados da mesma forma tanto os estudantes que fornecem código como os que copiam código de outros.

T1: Classe *Veículo* (`Vehicle`)

Deve ser desenvolvida uma classe de objetos que representa o conceito de veículo.

Um veículo tem uma matrícula (cadeia de caracteres), um tipo (*ligeiro*, *pesado* ou *motociclo*), uma marca (cadeia de caracteres), um modelo (cadeia de caracteres), um proprietário (cadeia de caracteres) e um ano de registo (valor inteiro).

Toda a informação referida anteriormente deve ser fornecida durante a criação dum objeto deste tipo. Após a criação, não deverá ser possível alterar a informação do veículo.

Esta classe deverá disponibilizar as seguintes funcionalidades:

- obter o valor de cada um dos atributos;
- converter um veículo numa cadeia de caracteres, permitindo assim mostrar os seus detalhes com base na função `print()`;
- comparar veículos usando o operador `==`, considerando que dois veículos são iguais se tiverem a mesma matrícula.

A representação textual de um veículo deve seguir o seguinte formato:

```
matrícula,tipo,marca,modelo,proprietário,ano_registo
```

Por exemplo:

```
ZZ-99-ZZ,ligeiro,Seat,Ibiza,Zé Carlos,2020
```

T2: Leitura do registo de veículos

O registo de veículos deve ser carregado de um ficheiro e deverá ser armazenado numa estrutura de dados, tal como uma lista ou um dicionário (recomendado). O formato do ficheiro consiste num veículo por linha, usando a representação textual descrita em T1.

Exemplo:

```
ZZ-99-ZZ,ligeiro,Seat,Ibiza,Zé Carlos,2020
VV-13-VV,pesado,Scania,Super,Maria Silva,2018
1,ligeiro,Red Bull Racing,RB20,Max Verstappen,2024
AA-01-01,motociclo,Casal,Boss,Ti Manel,1975
```

Note que será necessário criar o ficheiro manualmente (por exemplo, usando o exemplo anterior).

T3: Classe *Parque* (`Park`)

Deve ser desenvolvida uma classe de objetos que representa o conceito de parque de estacionamento.

Um parque tem um nome (cadeia de caracteres), uma localização (tuplo latitude, longitude, em que a latitude é um número em vírgula flutuante no intervalo [-90, 90] e a longitude é um número em vírgula flutuante no intervalo [-180, 180]) e uma lotação (número inteiro). Para além disso, um parque pode ser público ou privado. Caso seja privado, deve manter uma listagem das matrículas com permissão para aceder ao parque.

Ao criar um objeto deste tipo deve ser fornecido o nome do parque, a sua localização, a sua lotação e uma indicação se o parque é privado. Após a criação, não deverá ser possível alterar esta informação. No caso de o parque ser privado, a listagem de permissões deve estar inicialmente vazia.

Esta classe deverá disponibilizar as seguintes funcionalidades:

- obter o valor de cada um dos atributos;
- registar entradas e saídas de veículos do parque;
- saber se um determinado veículo está no parque;
- listar os veículos estacionados no parque;
- saber o número de lugares ocupados e livres;
- converter um parque numa cadeia de caracteres, permitindo assim mostrar os seus detalhes com base na função `print()`;

No caso de o parque ser privado, deve também disponibilizar as seguintes funcionalidades:

- permitir acesso ao parque
- revogar acesso ao parque

A representação textual de um parque deve seguir o seguinte formato:

```
nome (latitude, longitude) ocupação/lotação
```

Por exemplo:

```
Parque do ISCTE (38.7478, -9.1534) 17/380
```

T4: Gestão de parques

A gestão de parques de estacionamento é feita a partir do seguinte menu de interação com o utilizador:

1. Listar parques
2. Gerir parque
3. Criar parque
4. Remover parque
5. Estatísticas e informações
0. Sair

O menu encontra-se em ciclo, permitindo realizar as várias operações múltiplas vezes, até que o utilizador escolha a opção 0, que permite sair do programa.

1. Listar parques

Esta opção lista os parques existentes, um por linha, usando a sua representação textual, descrita em T3.

2. Gerir parque

Esta opção permite aceder ao menu de gestão de um parque descrito em T5. Para isso, o nome do parque deve ser pedido ao utilizador, apresentando a mensagem `Nome:`. Caso não exista um parque com esse nome, a operação deve ser interrompida e apresentada a mensagem `Não existe um parque com esse nome.`

3. Criar parque

Esta opção permite criar um objeto do tipo *Parque*. Para isso, a informação necessária para a sua criação deve ser pedida ao utilizador pela ordem e usando as mensagens descritas na tabela abaixo. Em cada passo, a informação introduzida pelo utilizador deve ser validada e, caso existam problemas, a operação deve ser interrompida e apresentada a mensagem correspondente ao problema.

Informação	Mensagem de Pedido	Mensagem de Erro
Nome	`Nome:`	`Já existe um parque com esse nome.`
Localização	`Localização:`	`Coordenadas inválidas.`
Capacidade	`Capacidade:`	`Capacidade inválida.`
Privacidade	`Privado (s/n):`	`Opção inválida.`

4. Remover parque

Esta opção permite remover um parque. Para isso, o nome do parque deve ser pedido ao utilizador, apresentando a mensagem `Nome:`. Caso não exista um parque com esse nome, a operação deve ser interrompida e apresentada a mensagem `Não existe um parque com esse nome.`

5. Estatísticas e informações

Esta opção permite aceder ao menu de estatísticas e informações descrito em T6.

T5: Gestão de um parque

A gestão de um parque é feita através do seguinte menu de interação com o utilizador (este menu é acedido a partir do menu principal descrito em T4):

1. Registar entrada
2. Registar saída
3. Listar veículos
4. Exportar estado
5. Listar permissões
6. Permitir acesso
7. Revogar acesso
0. Voltar

Caso o parque seja público, as opções 5, 6 e 7 não devem surgir no menu.

Note que a maioria das opções mapeiam diretamente nas funcionalidades da classe *Parque* descrita em T3.

1. Registar entrada

Esta opção permite registar a entrada de um veículo no parque. Para isso, a matrícula do veículo deve ser pedida ao utilizador, apresentando a mensagem `Matrícula:`. Existe um conjunto de restrições, descritas na tabela abaixo, que podem impedir a entrada de um veículo no parque. Estas devem ser verificadas pela ordem apresentada na tabela. Caso alguma restrição se aplique, a operação deve ser interrompida e apresentada a mensagem de erro correspondente.

Restrição	Mensagem de Erro
O veículo tem de estar no registo de veículos	`O veículo não existe.`
O veículo não pode estar estacionado num parque	`O veículo já está num parque.`
Caso o parque seja privado, a matrícula tem de estar registada na lista de permissões	`O veículo não tem permissão para entrar no parque.`
Não são permitidos veículos pesados nos parques	`Não são permitidos veículos pesados.`
O parque tem de ter lugares livres	`O parque está cheio.`

2. Registar saída

Esta opção permite registar a saída de um veículo do parque. Para isso, a matrícula do veículo deve ser pedida ao utilizador, apresentando a mensagem `Matrícula:`.

Caso o veículo não exista no registo de veículos, a operação deve ser interrompida e apresentada a mensagem `O veículo não existe.`. Caso o veículo exista, mas não esteja no parque, deve ser apresentada a mensagem `O veículo não está no parque.`.

3. Listar veículos

Esta opção lista os veículos estacionados no parque, um por linha, usando a sua representação textual, descrita em T1.

4. Exportar estado

Esta opção permite exportar o estado atual do parque para um ficheiro. Para isso, o nome do ficheiro deve ser pedido ao utilizador, apresentando a mensagem `Ficheiro:`. A primeira linha escrita no ficheiro deve corresponder à representação textual do parque, descrita em T3. Em seguida, deve ser escrita a listagem dos veículos estacionados no parque, tal como descrito para a opção anterior.

5. Listar permissões

Esta opção, disponível apenas para os parques privados, lista as matrículas dos veículos com permissão para entrar no parque, uma por linha.

6. Permitir acesso

Esta opção, disponível apenas para os parques privados, permite adicionar uma matrícula à lista de permissões do parque. Para isso, a matrícula do veículo deve ser pedida ao utilizador, apresentando a mensagem `Matrícula:`.

7. Revogar acesso

Esta opção, disponível apenas para os parques privados, permite remover uma matrícula da lista de permissões do parque. Para isso, a matrícula do veículo deve ser pedida ao utilizador, apresentando a mensagem `Matrícula:`.

T6: Estatísticas e informações

Tal como o nome indica, este menu, acedido a partir do menu principal descrito em T4, permite obter algumas informações e estatísticas sobre os parques e os veículos neles estacionados:

1. Número total de lugares livres
2. Taxa de ocupação média
3. Percentagem de parques privados
4. Histograma por tipo de veículo
5. Visualizar mapa de parques

0. Voltar

O menu encontra-se em ciclo, permitindo realizar as várias operações múltiplas vezes, até que o utilizador escolha a opção 0, que permite voltar ao menu principal.

1. Número total de lugares livres

Esta opção mostra a soma do número atual de lugares livres de todos os parques registados no sistema.

2. Taxa de ocupação média

Esta opção mostra a taxa de ocupação média dos parques registados no sistema.

3. Percentagem de parques privados

Esta opção mostra a percentagem de parques registados no sistema que são privados. O valor da percentagem deve ser apresentado no intervalo [0, 1].

4. Histograma por tipo de veículo

Esta opção mostra um gráfico com o histograma dos veículos estacionados nos parques por tipo de veículo. Para implementar esta opção, pode ser útil consultar a secção Representação com gráficos.

5. Visualizar mapa de parques

Esta opção mostra um gráfico com o mapa dos parques. Os eixos do gráfico devem corresponder à latitude e longitude, considerando os respetivos limites. Cada parque deve ser representado por um círculo com centro na sua localização. O raio e a cor do círculo são definidos pela capacidade e a taxa de ocupação do parque, respetivamente. O raio deve ser 2 caso o parque tenha menos de 100 lugares, 5 caso tenha menos de 1000 e 10 nos restantes casos. A cor deve ser verde caso a taxa de ocupação seja inferior a 75%, amarela caso seja inferior a 100% e vermelha caso o parque esteja lotado. Para implementar esta opção, pode ser útil consultar a secção Representação com gráficos.

Representação com gráficos

De seguida mostram-se exemplos de gráficos em Python

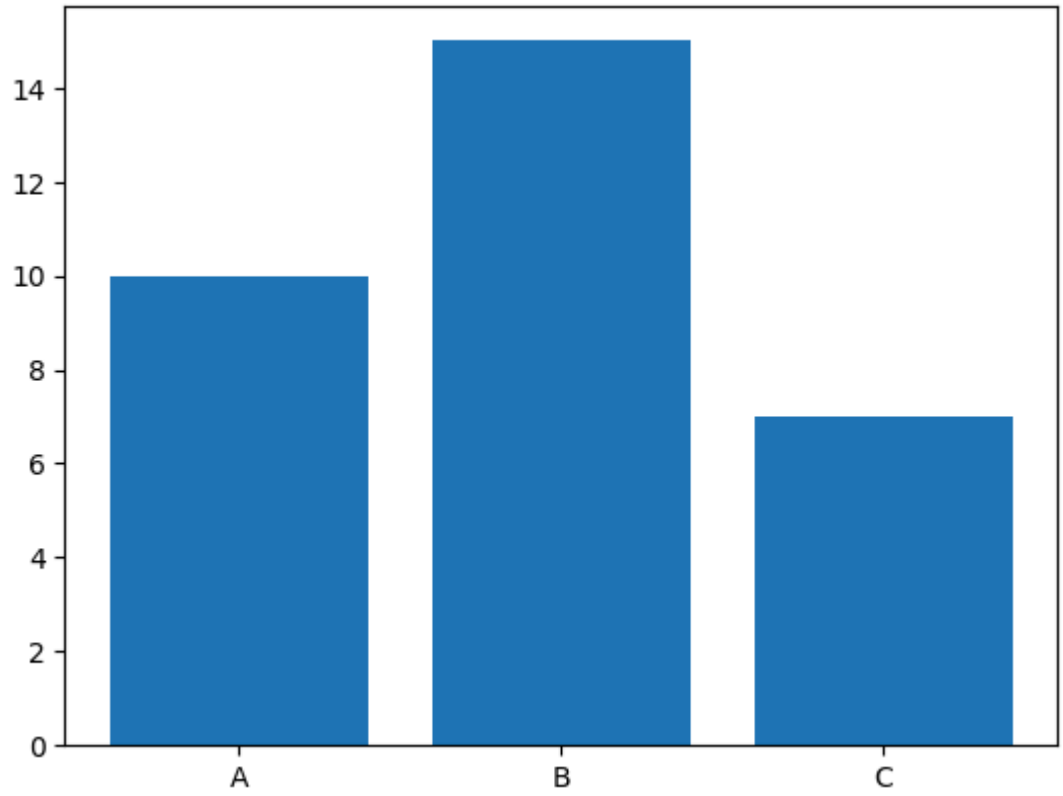
> **Observação:** Pode ser necessário instalar a biblioteca *matplotlib*.

In [1]:
```
import matplotlib.pyplot as plt

hist = {
    'A': 10,
```

```
    'B': 15,
    'C': 7
}

plt.bar(hist.keys(), hist.values())
plt.show(block=False)
```



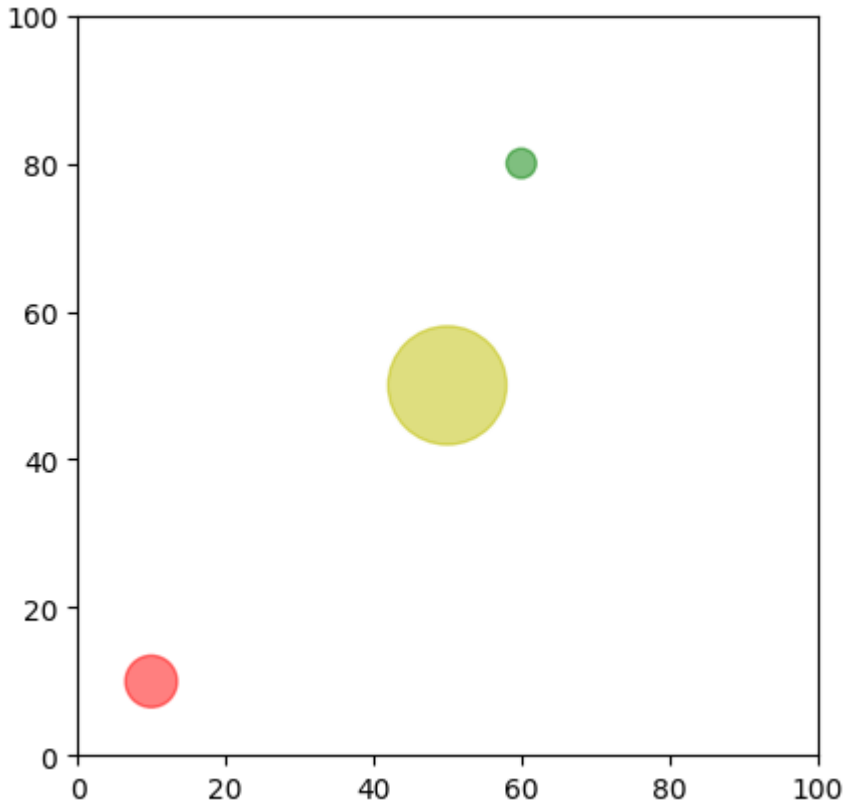
In [2]:
```
import matplotlib.pyplot as plt

fig, ax = plt.subplots()
ax.set_xlim((0, 100))
ax.set_ylim((0, 100))
ax.set_box_aspect(1)

circles = [
    {'position': (10,10), 'radius': 3.5, 'color': 'r'},
    {'position': (50,50), 'radius': 8.0, 'color': 'y'},
    {'position': (60,80), 'radius': 2.0, 'color': 'g'}
]

for c in circles:
    circle = plt.Circle(c['position'], c['radius'], color=c['color'], alp
    ax.add_patch(circle)

plt.show(block=False)
```



Entrega e Avaliação

O trabalho deverá ser entregue através do Moodle. Cada estudante deverá submeter um ficheiro zip com o código e ficheiros adicionais que possa ter usado no âmbito do trabalho.

TODO: Alterar esta parte. Especialmente as cotações. O projeto será classificado numa escala de 0 a 20 valores, mas a nota atribuída será sempre um múltiplo de 2, por exemplo, 8, 10, 12, 14, 16, 18 ou 20. O peso de cada uma das tarefas na nota é o seguinte: T1: 0v, T2: 0v, T3: 0v, T4: 0v, T5: 0v, T6: 0v

O projeto será inicialmente avaliado em termos funcionais, i.e., se as funções e procedimentos produzem os resultados esperados e os objetos dos tipos das classes a desenvolver têm o comportamento esperado, independentemente da forma como estão implementados. Desta avaliação resultará uma classificação, que poderá sofrer uma penalização em função da qualidade do código. Os estudantes poderão obter *feedback* junto dos professores sobre o progresso do projeto e qualidade do código e deverão seguir as recomendações dadas.

Política em caso de fraude

Os estudantes podem partilhar e/ou trocar ideias entre si sobre os trabalhos e/ou resolução dos mesmos. No entanto, o trabalho entregue deve corresponder ao esforço individual de cada estudante. São consideradas fraudes as seguintes situações:

- trabalho parcialmente copiado;
- facilitar a copia através da partilha de ficheiros;
- utilizar material alheio sem referir a sua fonte.

Em caso de deteção de algum tipo de fraude, os trabalhos em questão não serão avaliados, sendo enviados à Comissão Pedagógica ou ao Conselho Pedagógico, consoante a gravidade da situação, que decidirão a sanção a aplicar aos estudantes envolvidos. Serão utilizadas as ferramentas *Moss* e *SafeAssign* para detecção automática de cópias.

Recorda-se ainda que o Anexo I do Código de Conduta Académica, publicado a a 25 de janeiro de 2016 em Diário da Republica, 2ª Série, nº 16, indica no seu ponto 2 que:

> Quando um trabalho ou outro elemento de avaliação apresentar um nível de coincidência elevado com outros trabalhos (percentagem de coincidência com outras fontes reportada no relatório que o referido software produz), cabe ao docente da UC, orientador ou a qualquer elemento do júri, após a análise qualitativa desse relatório, e em caso de se confirmar a suspeita de plágio, desencadear o respetivo procedimento disciplinar, de acordo com o Regulamento Disciplinar de Discentes do ISCTE - Instituto Universitário de Lisboa, aprovado pela deliberação n.o 2246/2010, de 6 de dezembro.

O ponto 2.1 desse mesmo anexo indica ainda que:

> No âmbito do Regulamento Disciplinar de Discentes do ISCTE-IUL, são definidas as sanções disciplinares aplicáveis e os seus efeitos, podendo estas variar entre a advertência e a interdição da frequência de atividades escolares no ISCTE-IUL até cinco anos.