



MÓDULO 3. UNIDADES 1, 2, 3 e 4

Exercícios propostos de Classes e
Objetos



DIRETRIZES GERAIS

- Guardar o documento de soluções com o seguinte formato para entrega:
M2_01_nome_apelido1_apelido2
- Software recomendado: **Anaconda** – Jupyter. Junto a este documento está um padrão de Jupyter com os enunciados
- Vamos fazer os exercícios com as ferramentas vistas nas sessões. Ainda não chegámos a condicionais nem ciclos
- **Comentar o código**
- Utilizar nomes de variáveis apropriados. Se vamos guardar uma nota, chamamos a essa variável nota, não n ou x.



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 1

Criação

- Criar uma classe chamada Aluno que tenha os atributos nome e nota.
- Fazer o construtor da classe. Adicionar no construtor um print para informar que o aluno foi criado com sucesso.
- Adicionar o método string (`str`), para que ao imprimir por ecrã, se veja a sua informação.
- Criar um método chamado **qualificacao** que imprima por ecrã se o aluno aprovou ou suspendeu a sua nota na base

Experimentação

- Criar alguns alunos
- Experimentar mostrar os objetos aluno e o método de qualificação



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 2

Criação

Sobre o exercício anterior fazer a seguinte modificação.

No exercício, fazer um `print(alumno1.nota)`, vai devolver a nota do aluno. Pode chegar a ser um problema de segurança. Vamos encapsular o código para que não se possa aceder diretamente às variáveis e o acesso tenha de ser através de métodos.

- Fazer com que as variáveis da classe sejam **privadas**. Lembrete: adicionar duplo traço à frente do nome da variável
- Criar métodos **getter** e **setter** para poder aceder e modificar as variáveis da classe

Experimentação

- Criar alguns alunos
- Comprovar que apenas podemos aceder às variáveis da classe através dos seus métodos `get` e `set`



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 3

Criação

- Fazer uma classe chamada **Produto** que tenha os atributos **codigo**, **nome**, **preço** e **tipo**.
- Criar o **construtor** da classe. Adicionar no construtor um **print** para informar de que o produto foi criado com sucesso
- Criar métodos **getter** e **setter** para aceder e modificar todos os atributos da classe

Experimentação

- Fazer alguns produtos
- Experimentar estes novos métodos que se criaram (getter e setter)



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 4

Criação

- Criar uma classe chamada **Producto** que tenha os atributos **codigo**, **nome**, **preço** e **tipo**.
- Ampliar o exercício anterior: Copiar todo o código que já tenha e adicionar o seguinte:
- Substituir o método `__str__` (`__str__`), para imprimir um produto para o ecrã e mostre todos os dados desse produto. Exemplo:

CONTINUA NA FOLHA
SEGUINTE

```
print("\nProbando el print:")  
print(p1)|
```

Producto 1 creado
Producto 2 creado
Producto 3 creado

Probando el print:
Producto con código 1:
 Nombre: Macbook Pro
 Precio: 1000
 Tipo: Informatica



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 4

- Criar um método chamado `calcular_total` o qual receba um número de unidades para esse produto e com base nisso, calcule o preço total desse conjunto de produtos. Exemplo: 5 unidades do produto 1
- Experimentar estes novos métodos criados.
- Criar o **construtor** da classe. Adicionar um `print` ao construtor para informar que o produto foi criado com sucesso.
- Desenvolver métodos **getter** e **setter** para aceder e modificar todos os atributos da classe.

Experimentação

- Criar alguns produtos.
- Experimentar estes novos métodos que se criaram (getter e setter).



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 5

Criação

- Continuar a ampliar o exercício anterior: Copiar todo o código que tiver e adicionar o seguinte:
- Desenvolver uma classe chamada **Pedido** que tenha os atributos: uma lista de produtos e uma lista de quantidades.
- Criar o **construtor** da classe. Adicionar um print ao construtor para informar que o pedido foi criado com sucesso.
- Fazer o método **total_pedido** o qual devolva o montante total de um pedido.
- Criar o método **mostrar_pedido** o qual mostre todos os produtos e quantidades que compõem o pedido.

Experimentação

- Criar alguns produtos.
- Experimentar estes novos métodos que se criaram.



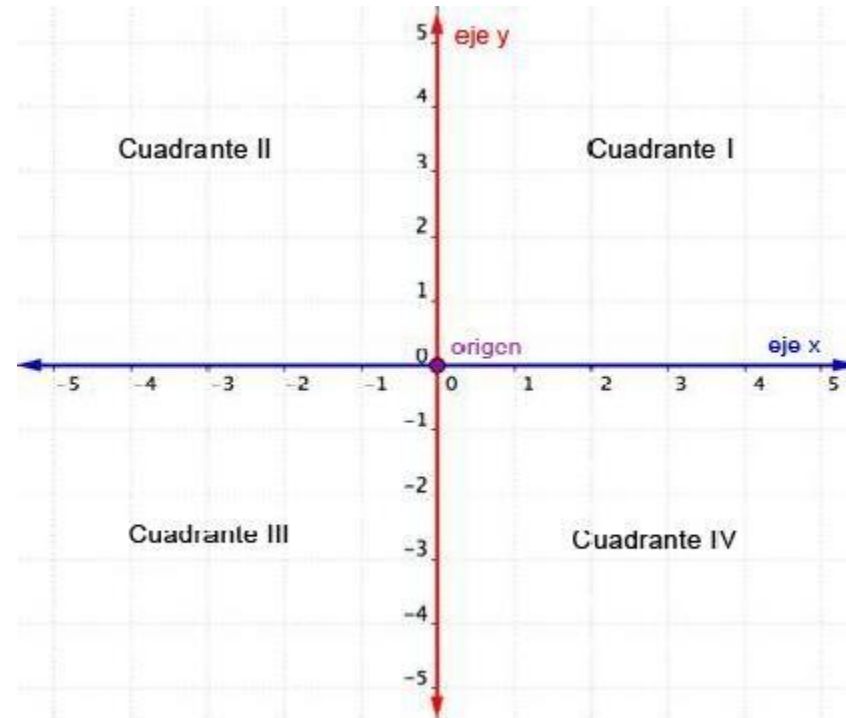
EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6

Exercício de pontos, coordenadas e vetores sobre o plano cartesiano. A seguir os conceitos básicos.

O plano cartesiano

Representa um espaço bidimensional (em 2 dimensões), formado por duas retas perpendiculares, uma horizontal e outra vertical que se cortam num ponto. A reta horizontal denomina-se eixo das abscissas ou **eixo X**, embora que a vertical recebe o nome de eixo das ordenadas ou, simplesmente, **eixo Y**.

Quanto ao ponto onde se cortam, conhece-se como o **ponto de origem O**. O plano divide-se em 4 quadrantes.



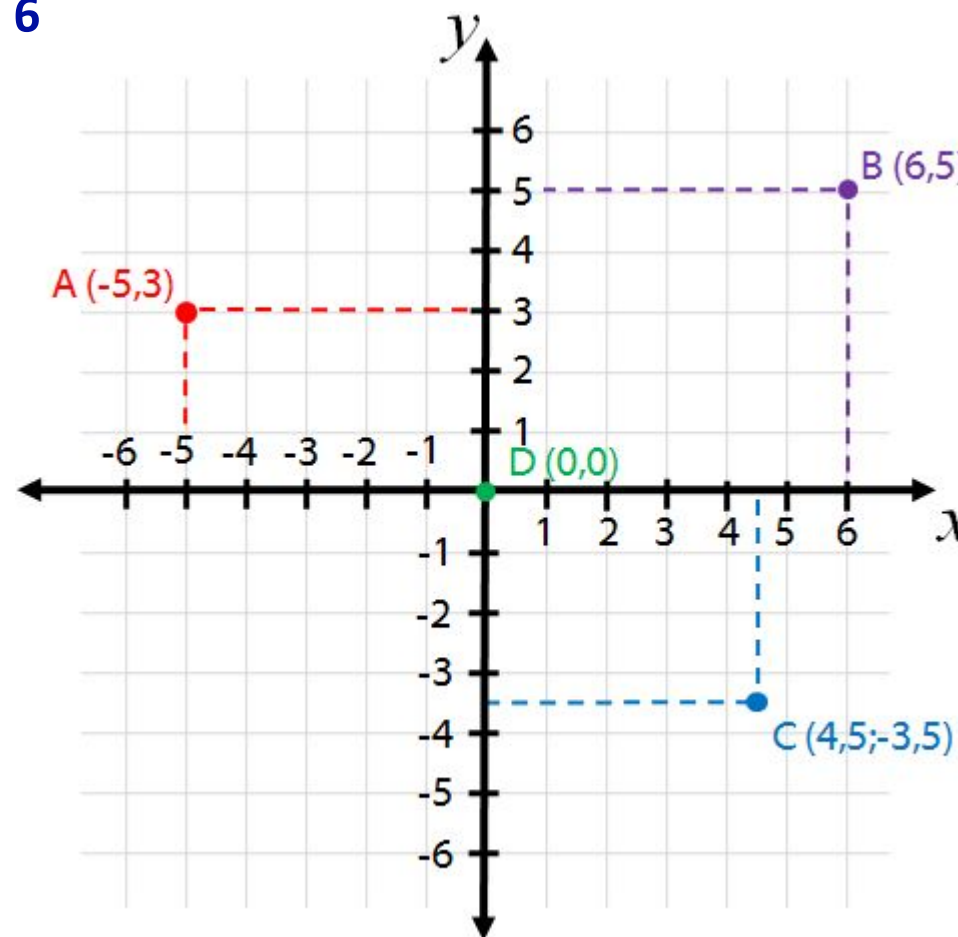


EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6

Pontos e coordenadas

O objetivo é descrever a posição de **pontos** sobre o plano em forma de **coordenadas**, que se formam associando o valor do eixo X (horizontal) com o valor do eixo Y (vertical).

A representação de um ponto é simples: $P(X,Y)$ onde X e Y são a distância horizontal (esquerda ou direita) e vertical (acima ou abaixo) respectivamente, utilizando como referência o ponto de origem (0,0), no centro do plano.



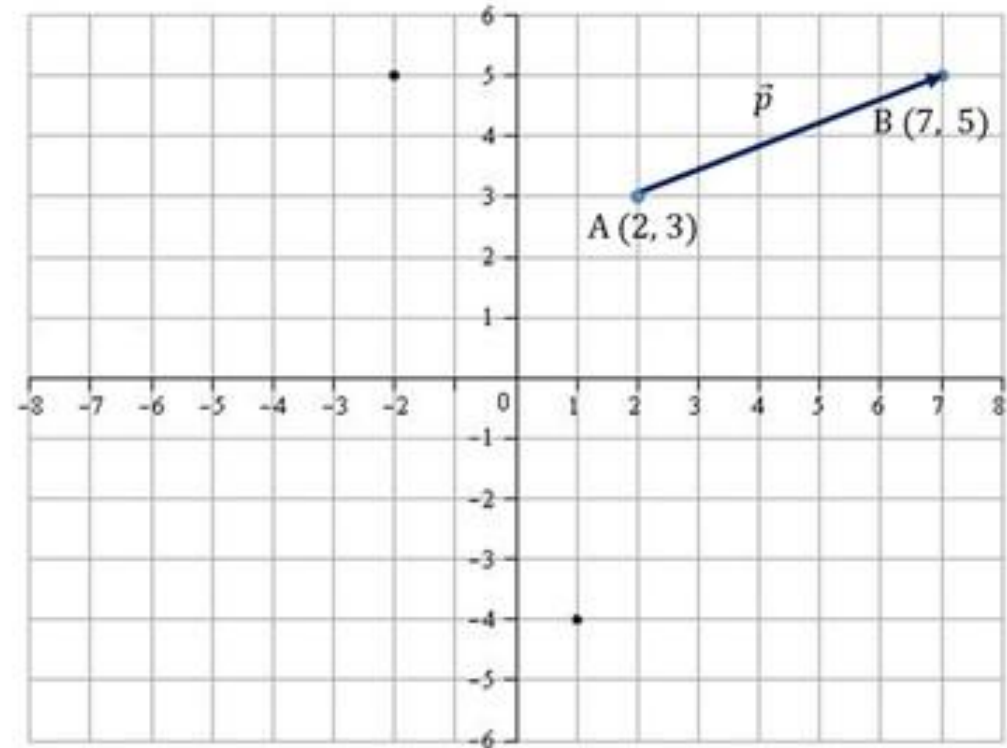


EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6

Vetores no plano

Finalmente, um vetor no plano faz referência a um segmento orientado, gerado a partir de dois pontos distintos.

A efeitos práticos não deixa de ser uma linha feita desde um ponto inicial em direção a outro ponto final, pelo que se entende que um vetor tenha comprimento e direção/sentido.



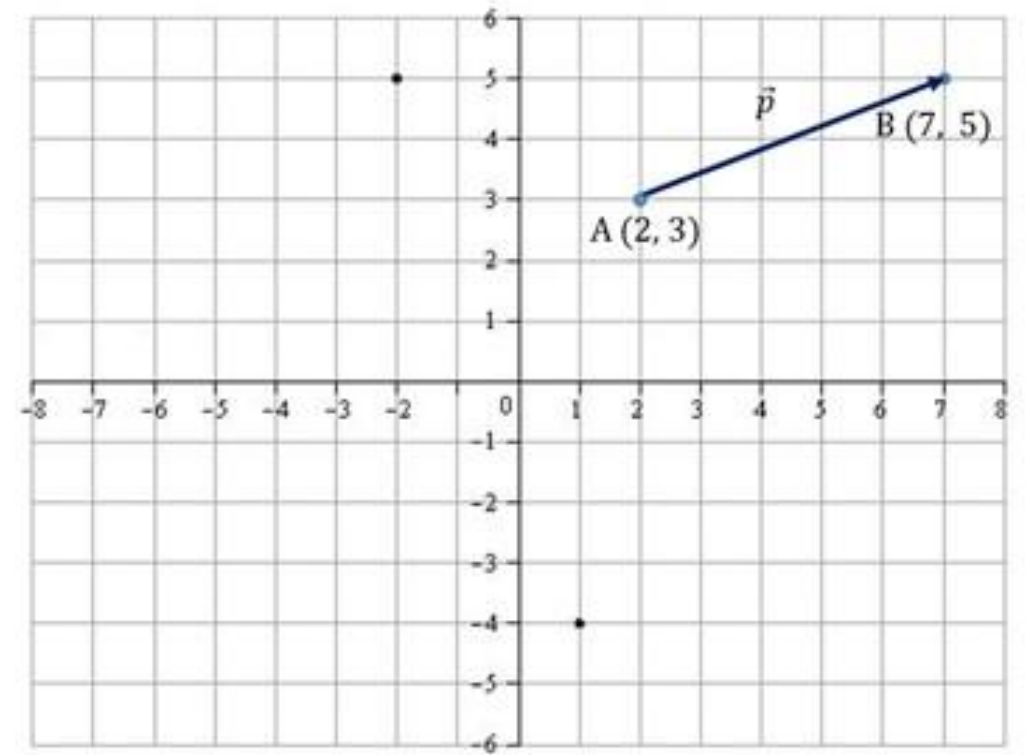
CURSO: PROGRAMAÇÃO PYTHON

EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6

Vetores no plano

Finalmente, um vetor no plano faz referência a um segmento orientado, gerado a partir de dois pontos distintos.

A efeitos práticos não deixa de ser uma linha feita desde um ponto inicial em direção a outro ponto final, pelo que se entende que um vetor tem comprimento e direção/sentido.



Nesta figura, podemos observar dois pontos A e B que poderíamos definir da seguinte forma:

- **$A(x_1, y_1) \Rightarrow A(2, 3)$**
- **$B(x_2, y_2) \Rightarrow B(7, 5)$**

E o vetor seria representado como a diferença entre as coordenadas do segundo ponto relativamente ao primeiro (o segundo menos o primeiro):

- **$AB = (x_2 - x_1, y_2 - y_1) \Rightarrow (7 - 2, 5 - 3) \Rightarrow (5, 2)$**

O que em definitiva não deixa de ser: 5 à direita e 2 acima.



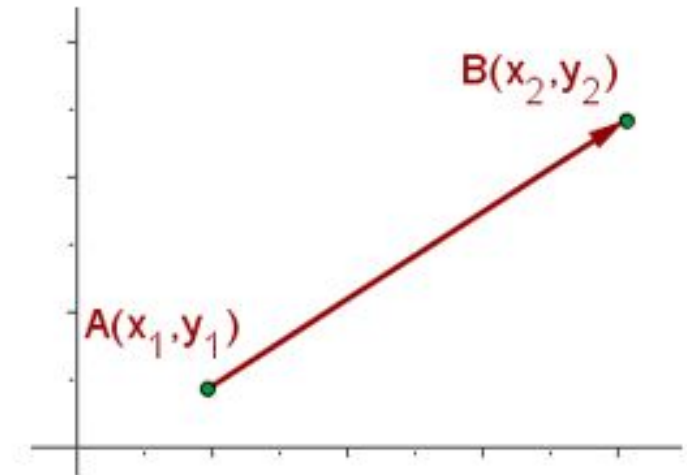
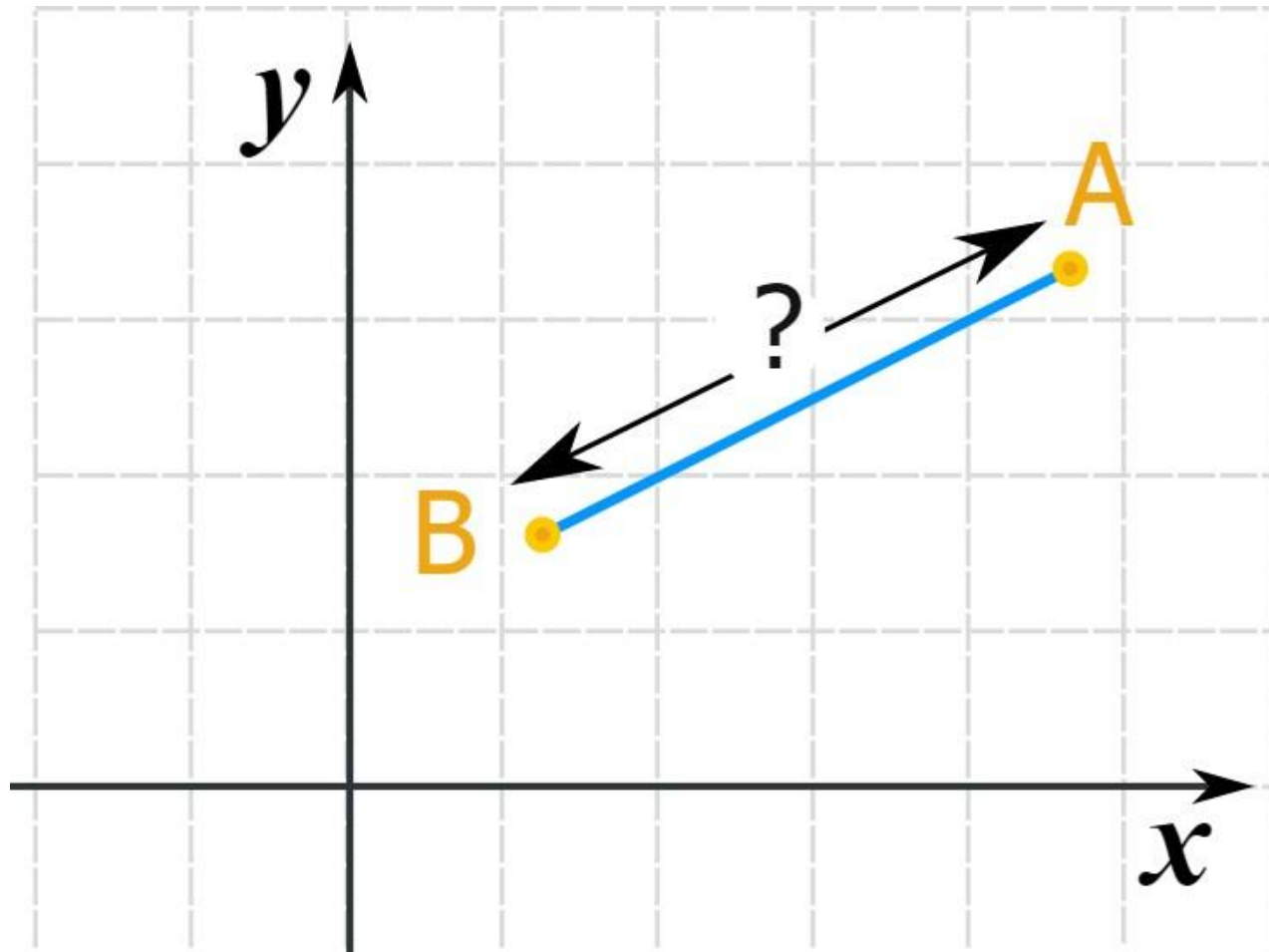
EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6

Criação

- Criar uma classe chamada Ponto com as suas duas coordenadas X e Y.
- Adicionar um método construtor para criar pontos facilmente. Se não se recebe uma coordenada, o seu valor será zero.
- Substituir o método string (`__str__`), para que ao imprimir um ponto por ecrã apareça em formato (X,Y)
- Adicionar um método chamado quadrante que indique a que quadrante pertence o ponto, ou se é a origem.
- Agregar um método chamado vetor, que tome outro ponto e calcule o vetor resultante entre os dois pontos.
- Incorporar um método chamado distância, que tome outro ponto e calcule a distância entre os dois pontos e a mostre por ecrã. A fórmula é a seguinte:



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6



• $A(x_1, y_1)$

$B(x_2, y_2)$

• $d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6

Nota: A função raiz quadrada em Python `sqrt()` deve-se importar do módulo `math` e utilizar da seguinte forma:

```
import math  
math.sqrt(9)
```

- Criar uma classe chamada Retângulo com dois pontos (inicial e final) que fazem a diagonal do retângulo.
- Incorporar um método construtor para criar ambos os pontos facilmente, se não se enviam criam-se dois pontos na origem por defeito.
- Adicionar ao retângulo um método chamado `base` que mostre a base.
- Agregar ao retângulo um método chamado `altura` que mostre a altura.
- Adicionar ao retângulo um método chamado `área` que mostre a área.
- Podemos identificar facilmente estes valores se tentar desenhar o quadrado a partir da sua diagonal. Se andar perdido, experimentar a desenhar num papel, de certeza que se vê muito mais claro! Além do mais podemos utilizar a função `abs()` para saber o valor absolute de um número.



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 6

Experimentação

- Criar os pontos A(2, 3), B(5,5), C(-3, -1) e D(0,0) e imprimir por ecrã.
- Averiguar a que quadrante pertence o ponto A, C e D.
- Consultar os vetores AB e BA.
- Consultar a distância entre os pontos 'A e B' e 'B e A'.
- Determinar qual dos 3 pontos A, B ou C, se encontra mais longe da origem, ponto (0,0).
- Fazer um retângulo utilizando os pontos A e B.
- Consultar a base, altura e área do retângulo.



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 7

Criação

Trabalhar com herança.

Criar uma classe chamada Veículo que tenha os atributos: cor e rodas

Fazer o construtor da classe.

Substituir o método string (`str`), para que ao imprimir um veículo por ecrã se possa ver a sua informação.

Fazer uma classe que se chame Carro que herde de Veículo

Criar o construtor da classe, que inclua tudo proveniente do construtor do veículo , mais a velocidade e o deslocamento.

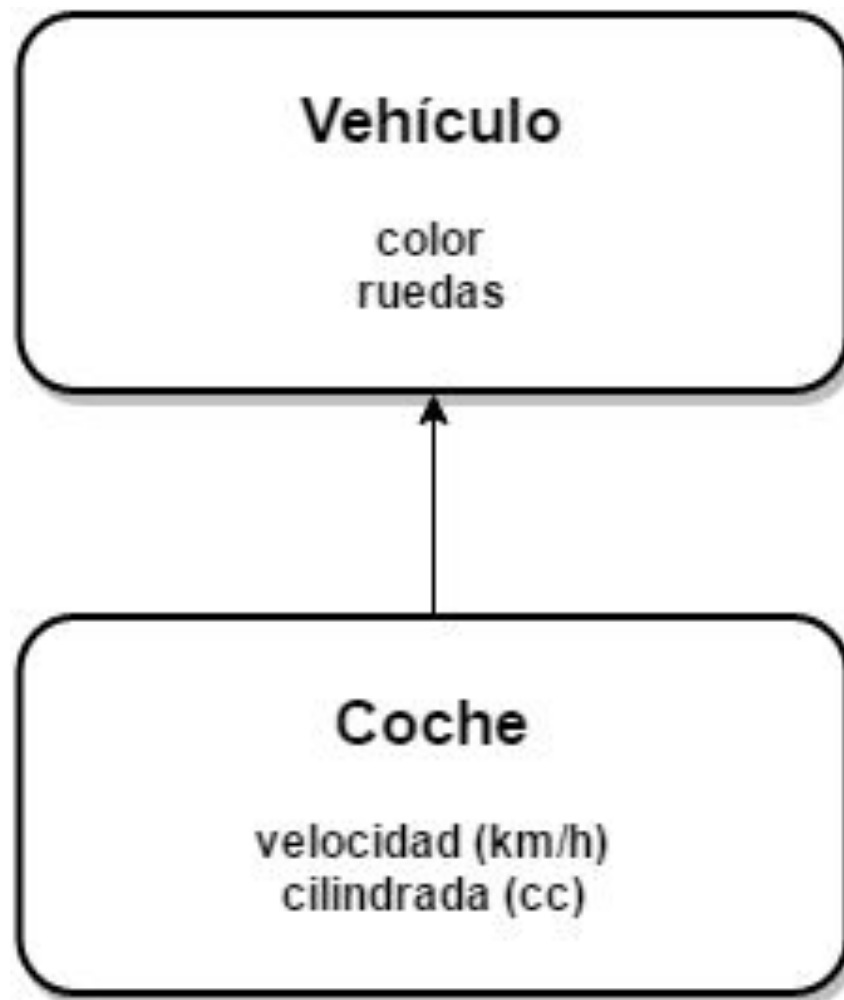
Substituir o método string (`str`), para que ao imprimir um carro para o ecrã se possa ver a sua informação

Experimentação

Criar alguns carros e mostrar a sua informação



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 7





EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 8

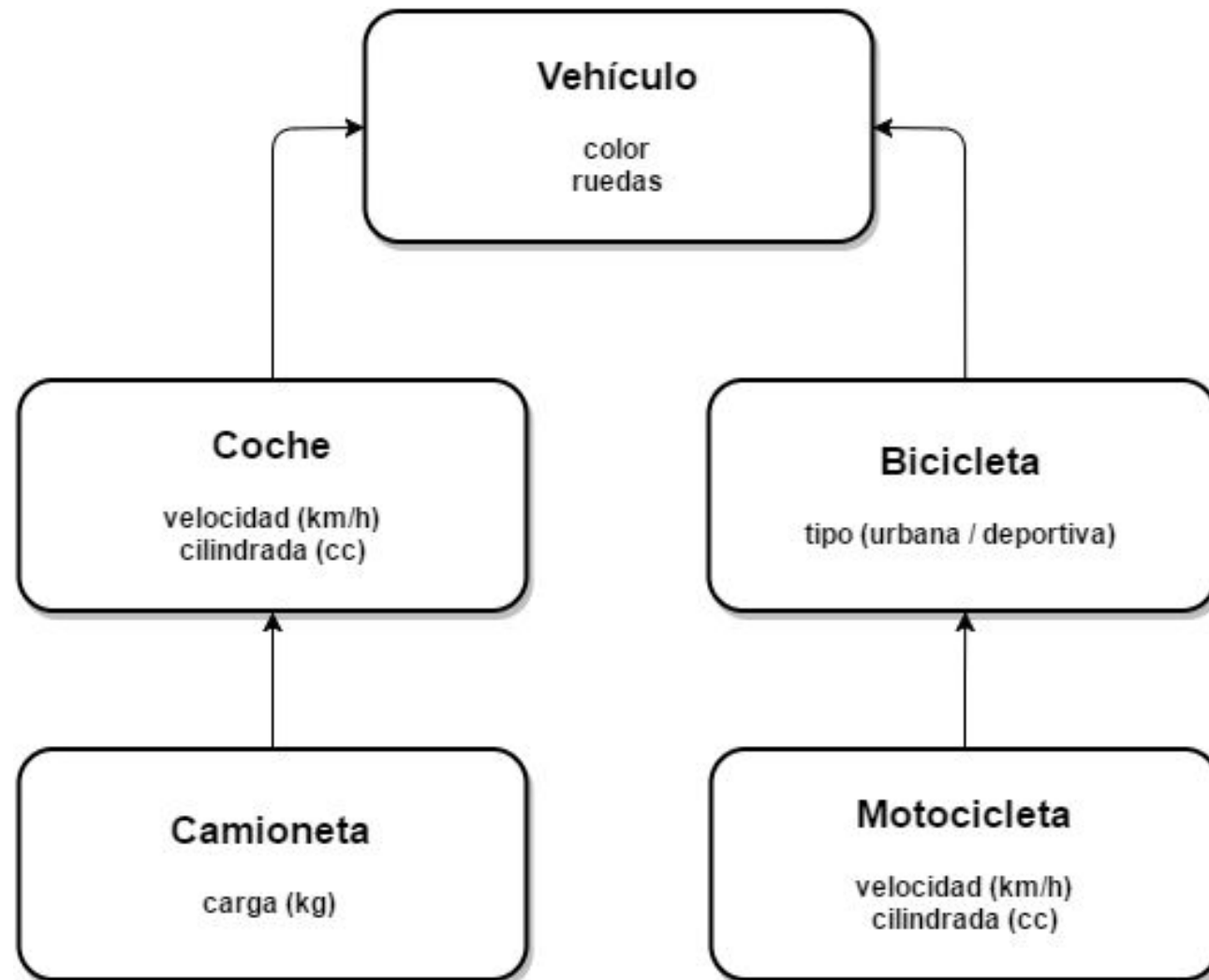
Trabalhar com herança. Continuar a ampliar o exercício anterior. Copiar todo o código que já tem e adicionar o seguinte:

- Criar as classes herdadas de Veículo que aparecem na imagem.
- Fazer os construtores para essas classes.
- Substituir os métodos string (`__str__`), para que ao imprimir por ecrã se veja a informação correspondente.
- Criar uma lista e dentro dela introduzir pelo menos um objeto de cada subclasse de Veículo
- Fazer uma função chamada catalogar() que receba a lista de veículos e os percorra mostrando o nome da sua classe e os seus atributos.
- Modificar a função catalogar() para que receba um argumento opcional de rodas, fazendo com que mostre apenas o número de rodas que concorde com o valor do argumento. Também deve mostrar uma mensagem "Encontrou-se {} veículos com {} rodas:" apenas se enviar o argumento de rodas. Colocar à prova com 0, 2 e 4 rodas como valor.

Pista. Investigar para que serve esta instrução e se nos vai ser útil `type(objeto).__name__`



EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 8





EXERCÍCIOS DE CLASSES E OBJETOS. EXERCÍCIO 9

Criação

Trabalhar com herança. Continuar a ampliar o exercício anterior. Copiar todo o código que já tem e adicionar o seguinte:

Um método que se chame filtrar que receba 2 parâmetros, a lista de veículos e o nome de um dos tipos de veículos, para poder filtrar na lista, ou seja, Carro, Camioneta, etc.. O método deve percorrer a lista, encontrar as coincidências e guardá-las numa lista nova. Esta nova lista deve-se devolver ao programa principal onde recorre e mostra o seu conteúdo. A chamada no programa principal deverá ser assim: `listaCarros = filtrar(lista, "Carro")`

Experimentação

Criar alguns veículos (mais de um por tipo)

Filtrar pelos diferentes tipos de Veículo e mostrar os resultados retornados