

# Notas de aula de programação em Python

## Curso avançado

### Topico 1 - Classes

Prof. Louis Augusto

`louis.augusto@ifsc.edu.br`



**INSTITUTO FEDERAL  
SANTA CATARINA**

Instituto Federal de Santa Catarina  
São José

- 1 Introdução à programação orientada a objetos
  - Classes como modelo de objetos
- 2 Metendo a mão na massa
  - Arquivo de pastas suspensas
  - Caixa acoplada de vaso sanitário

- 1 Introdução à programação orientada a objetos
  - Classes como modelo de objetos
- 2 Metendo a mão na massa
  - Arquivo de pastas suspensas
  - Caixa acoplada de vaso sanitário

# Classes como modelo de objetos

Um objeto na vida real costuma guardar duas características (já traduzido para a linguagem usada para classes):

**Atributos:** são as distinções de um objeto em relação a outro, como cor, forma, valor, comprimento etc.

**Métodos:** são as ações que o objeto pode realizar ou mudança de estado, como estar ligado ou desligado, produzindo algo ou não, mudando algo de lugar, ou qualquer alteração em outro objeto ou em si próprio.

Pode-se imaginar um carro como um grande objeto, que é formado por vários objetos menores que podem trocar informações entre si ou não. Por exemplo, o motor do carro, o limpador de pára-brisa, as portas (que podem estar abertas ou fechadas), etc.

Dentro de uma linguagem computacional um objeto é modelado por uma classe.

# Classes como modelo de objetos

Um objeto na vida real costuma guardar duas características (já traduzido para a linguagem usada para classes):

**Atributos:** são as distinções de um objeto em relação a outro, como cor, forma, valor, comprimento etc.

**Métodos:** são as ações que o objeto pode realizar ou mudança de estado, como estar ligado ou desligado, produzindo algo ou não, mudando algo de lugar, ou qualquer alteração em outro objeto ou em si próprio.

Pode-se imaginar um carro como um grande objeto, que é formado por vários objetos menores que podem trocar informações entre si ou não. Por exemplo, o motor do carro, o limpador de pára-brisa, as portas (que podem estar abertas ou fechadas), etc.

Dentro de uma linguagem computacional um objeto é modelado por uma classe.

# Classes como modelo de objetos

Um objeto na vida real costuma guardar duas características (já traduzido para a linguagem usada para classes):

- Atributos:** são as distinções de um objeto em relação a outro, como cor, forma, valor, comprimento etc.
- Métodos:** são as ações que o objeto pode realizar ou mudança de estado, como estar ligado ou desligado, produzindo algo ou não, mudando algo de lugar, ou qualquer alteração em outro objeto ou em si próprio.

Pode-se imaginar um carro como um grande objeto, que é formado por vários objetos menores que podem trocar informações entre si ou não. Por exemplo, o motor do carro, o limpador de pára-brisa, as portas (que podem estar abertas ou fechadas), etc.

Dentro de uma linguagem computacional um objeto é modelado por uma classe.

# Classes como modelo de objetos

Um objeto na vida real costuma guardar duas características (já traduzido para a linguagem usada para classes):

- Atributos:** são as distinções de um objeto em relação a outro, como cor, forma, valor, comprimento etc.
- Métodos:** são as ações que o objeto pode realizar ou mudança de estado, como estar ligado ou desligado, produzindo algo ou não, mudando algo de lugar, ou qualquer alteração em outro objeto ou em si próprio.

Pode-se imaginar um carro como um grande objeto, que é formado por vários objetos menores que podem trocar informações entre si ou não. Por exemplo, o motor do carro, o limpador de pára-brisa, as portas (que podem estar abertas ou fechadas), etc.

Dentro de uma linguagem computacional um objeto é modelado por uma classe.

# Classes como modelo de objetos

Um objeto na vida real costuma guardar duas características (já traduzido para a linguagem usada para classes):

**Atributos:** são as distinções de um objeto em relação a outro, como cor, forma, valor, comprimento etc.

**Métodos:** são as ações que o objeto pode realizar ou mudança de estado, como estar ligado ou desligado, produzindo algo ou não, mudando algo de lugar, ou qualquer alteração em outro objeto ou em si próprio.

Pode-se imaginar um carro como um grande objeto, que é formado por vários objetos menores que podem trocar informações entre si ou não. Por exemplo, o motor do carro, o limpador de pára-brisa, as portas (que podem estar abertas ou fechadas), etc.

Dentro de uma linguagem computacional um objeto é modelado por uma classe.



# Classes como modelo de objetos

Um objeto na vida real costuma guardar duas características (já traduzido para a linguagem usada para classes):

**Atributos:** são as distinções de um objeto em relação a outro, como cor, forma, valor, comprimento etc.

**Métodos:** são as ações que o objeto pode realizar ou mudança de estado, como estar ligado ou desligado, produzindo algo ou não, mudando algo de lugar, ou qualquer alteração em outro objeto ou em si próprio.

Pode-se imaginar um carro como um grande objeto, que é formado por vários objetos menores que podem trocar informações entre si ou não. Por exemplo, o motor do carro, o limpador de pára-brisa, as portas (que podem estar abertas ou fechadas), etc.

Dentro de uma linguagem computacional um objeto é modelado por uma classe.

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:

○ Pode receber uma pasta nova;

○ Pode remover uma pasta;

○ Pode transferir uma pasta para outro

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:

○ Pode receber uma pasta nova;

○ Pode remover uma pasta;

○ Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).



# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

# Modelagem de um objeto

Vamos modelar uma arquivo de pastas suspensas que guarda informações sobre pessoas.



Este arquivo pode possuir várias características:

- Pode ter uma cor;
- Pode pertencer a um departamento.
- Pode ter um limite máximo de pastas que pode conter.
- Pode realizar algumas ações sobre uma pasta:
  - Pode receber uma pasta nova;
  - Pode remover uma pasta;
  - Pode transferir uma pasta para outro arquivo.

Este é um exemplo de objeto que possui atributos e pode realizar ações.

Modelamos este objeto com uma classe, que é um comando que guarda em seu bojo os atributos (variáveis) e seus métodos (funções).

- 1 Introdução à programação orientada a objetos
  - Classes como modelo de objetos
- 2 Metendo a mão na massa
  - Arquivo de pastas suspensas
  - Caixa acoplada de vaso sanitário

# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos. Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.

# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos. Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.



# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos.

Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.

# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos. Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.

# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos. Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.

# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos. Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.

# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos. Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.

# Arquivo de pastas suspensas

Supondo que no arquivo de pastas suspensas vamos guardar dados sobre pessoas, como uma pessoa também tem atributos e pode realizar ações, vamos definir um objeto pessoa.

## Programa 1

Vamos inicialmente definir uma pessoa somente com atributos, sem métodos. Os atributos devem ser: nome, idade, sexo.

Crie uma quantidade de pessoas, guarde as informações numa lista e depois mostre ao usuário as pessoas.

## Programa 2

Crie agora um arquivo de pastas suspensas que guarde as pessoas, este arquivo pode receber uma pessoa ou excluir uma pessoa, e pode mostrar ao usuário uma pessoa específica ou todas as pessoas no arquivo.

Considere para o objeto Pessoa, que se for uma mulher maior que 30 anos, sua idade será confidencial e não poderá ser mostrada ao usuário.

- 1 Introdução à programação orientada a objetos
  - Classes como modelo de objetos
- 2 Metendo a mão na massa
  - Arquivo de pastas suspensas
  - Caixa acoplada de vaso sanitário

# Caixa acoplada

Vamos codificar o objeto abaixo, um vaso sanitário e uma caixa acoplada.



O processo é simples: acabou o serviço, aperte um botão e a água desce da caixa acoplada levando o que tem ali do vaso sanitário para a rede de esgoto.



# Caixa acoplada

Vamos codificar o objeto abaixo, um vaso sanitário e uma caixa acoplada.



O processo é simples: acabou o serviço, aperte um botão e a água desce da caixa acoplada levando o que tem ali do vaso sanitário para a rede de esgoto.

# Caixa acoplada

Vamos codificar o objeto abaixo, um vaso sanitário e uma caixa acoplada.



O processo é simples: acabou o serviço, aperte um botão e a água desce da caixa acoplada levando o que tem ali do vaso sanitário para a rede de esgoto.

# Caixa acoplada

Vamos codificar o objeto abaixo, um vaso sanitário e uma caixa acoplada.



O processo é simples: acabou o serviço, aperte um botão e a água desce da caixa acoplada levando o que tem ali do vaso sanitário para a rede de esgoto.

# Caixa acoplada

Vamos codificar o objeto abaixo, um vaso sanitário e uma caixa acoplada.



O processo é simples: acabou o serviço, aperte um botão e a água desce da caixa acoplada levando o que tem ali do vaso sanitário para a rede de esgoto.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

**Alavanca de acionamento:** abrir a comporta de vedação, permitindo que a água saia da caixa.

**Comporta de vedação:** se fechar depois que toda a água saiu da caixa acoplada para o vaso.

**Válvula de alimentação:** Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

**Alavanca de acionamento:** abrir a comporta de vedação, permitindo que a água saia da caixa.

**Comporta de vedação:** se fechar depois que toda a água saiu da caixa acoplada para o vaso.

**Válvula de alimentação:** Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

Alavanca de acionamento: abrir a comporta de vedação, permitindo que a água saia da caixa.

Comporta de vedação: se fechar depois que toda a água saiu da caixa acoplada para o vaso.

Válvula de alimentação: Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

Alavanca de acionamento: abrir a comporta de vedação, permitindo que a água saia da caixa.

Comporta de vedação: se fechar depois que toda a água saiu da caixa acoplada para o vaso.

Válvula de alimentação: Encher de água a caixa até que a bóia chegue ao nível completo.



# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

Alavanca de acionamento: abrir a comporta de vedação, permitindo que a água saia da caixa.

Comporta de vedação: se fechar depois que toda a água saiu da caixa acoplada para o vaso.

Válvula de alimentação: Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

Alavanca de acionamento: abrir a comporta de vedação, permitindo que a água saia da caixa.

Comporta de vedação: se fechar depois que toda a água saiu da caixa acoplada para o vaso.

Válvula de alimentação: Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

Alavanca de acionamento: abrir a comporta de vedação, permitindo que a água saia da caixa.

Comporta de vedação: se fechar depois que toda a água saiu da caixa acoplada para o vaso.

Válvula de alimentação: Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

Alavanca de acionamento: abrir a comporta de vedação, permitindo que a água saia da caixa.

Comporta de vedação: se fechar depois que toda a água saiu da caixa acoplada para o vaso.

Válvula de alimentação: Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

**Alavanca de acionamento:** abrir a comporta de vedação, permitindo que a água saia da caixa.

**Comporta de vedação:** se fechar depois que toda a água saiu da caixa acoplada para o vaso.

**Válvula de alimentação:** Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

**Alavanca de acionamento:** abrir a comporta de vedação, permitindo que a água saia da caixa.

**Comporta de vedação:** se fechar depois que toda a água saiu da caixa acoplada para o vaso.

**Válvula de alimentação:** Encher de água a caixa até que a bóia chegue ao nível completo.

# Caixa acoplada

Queremos entender o funcionamento da caixa acoplada.

A caixa acoplada é formada de:

- Uma alavanca de acionamento: o botão pressionado no fim do serviço.
- Uma comporta de vedação: uma tampa de borracha cuja função é abrir para a água descer e fechar para encher a caixa de água.
- Uma válvula de alimentação: responsável por encher de água.
- Uma boia: cuja função é fechar a válvula de alimentação assim que o nível da água estiver completo.

Cada objeto tem uma função específica:

**Alavanca de acionamento:** abrir a comporta de vedação, permitindo que a água saia da caixa.

**Comporta de vedação:** se fechar depois que toda a água saiu da caixa acoplada para o vaso.

**Válvula de alimentação:** Encher de água a caixa até que a bóia chegue ao nível completo.

Faça um programa que simule a operação de uma caixa acoplada.

Para isto use os objetos:

**Caixa Acoplada:** Contém os outros objetos, tem como atributo o nível de água, e tem como métodos acionar o botão e acionar a entrada de água, controlando até o nível máximo.

**Comporta de vedação:** Tem como ações abrir, fechar e retornar o estado.

**Alavanca de acionamento:** Tem como método acionamento da alavanca e como atributo um contador de utilização.

**Válvula de alimentação:** Tem como método encher de água a caixa e como atributo a vazão.



Faça um programa que simule a operação de uma caixa acoplada.  
Para isto use os objetos:

**Caixa Acoplada:** Contém os outros objetos, tem como atributo o nível de água, e tem como métodos acionar o botão e acionar a entrada de água, controlando até o nível máximo.

**Comporta de vedação:** Tem como ações abrir, fechar e retornar o estado.

**Alavanca de acionamento:** Tem como método acionamento da alavanca e como atributo um contador de utilização.

**Válvula de alimentação:** Tem como método encher de água a caixa e como atributo a vazão.

# Caixa acoplada

Faça um programa que simule a operação de uma caixa acoplada.

Para isto use os objetos:

**Caixa Acoplada:** Contém os outros objetos, tem como atributo o nível de água, e tem como métodos acionar o botão e acionar a entrada de água, controlando até o nível máximo.

**Comporta de vedação:** Tem como ações abrir, fechar e retornar o estado.

**Alavanca de acionamento:** Tem como método acionamento da alavanca e como atributo um contador de utilização.

**Válvula de alimentação:** Tem como método encher de água a caixa e como atributo a vazão.

# Caixa acoplada

Faça um programa que simule a operação de uma caixa acoplada.

Para isto use os objetos:

**Caixa Acoplada:** Contém os outros objetos, tem como atributo o nível de água, e tem como métodos acionar o botão e acionar a entrada de água, controlando até o nível máximo.

**Comporta de vedação:** Tem como ações abrir, fechar e retornar o estado.

**Alavanca de acionamento:** Tem como método acionamento da alavanca e como atributo um contador de utilização.

**Válvula de alimentação:** Tem como método encher de água a caixa e como atributo a vazão.

# Caixa acoplada

Faça um programa que simule a operação de uma caixa acoplada.


Para isto use os objetos:




**Caixa Acoplada:** Contém os outros objetos, tem como atributo o nível de água, e tem como métodos acionar o botão e acionar a entrada de água, controlando até o nível máximo.




**Comporta de vedação:** Tem como ações abrir, fechar e retornar o estado.

**Alavanca de acionamento:** Tem como método acionamento da alavanca e como atributo um contador de utilização.

**Válvula de alimentação:** Tem como método encher de água a caixa e como atributo a vazão.

-  Flávio Ulhoa Coelho e Mary Lilian Lourenço,  
*Um curso de Álgebra Linear*  
*LedUsp - São Paulo.*  
Editora da Universidade de São Paulo.
-  Isabel Cabral, Cecília Perdigão e Carlos Saiago,  
*Álgebra Linear, 2ª edição.*  
*Escolar Editora, Lisboa, 2010.*
-  Seymour Lipschutz e Marc Lipson,  
*Álgebra Linear*  
*Coleção Schaum, 3ª edição.*  
Editora Bookman, Porto Alegre 2004.

-  Flávio Ulhoa Coelho e Mary Lilian Lourenço,  
*Um curso de Álgebra Linear*  
*LedUsp - São Paulo.*  
Editora da Universidade de São Paulo.
-  Isabel Cabral, Cecília Perdigão e Carlos Saiago,  
*Álgebra Linear, 2ª edição.*  
*Escolar Editora, Lisboa, 2010.*
-  Seymour Lipschutz e Marc Lipson,  
*Álgebra Linear*  
*Coleção Schaum, 3ª edição.*  
Editora Bookman, Porto Alegre 2004.

-  Flávio Ulhoa Coelho e Mary Lilian Lourenço,  
*Um curso de Álgebra Linear*  
*LedUsp - São Paulo.*  
Editora da Universidade de São Paulo.
-  Isabel Cabral, Cecília Perdigão e Carlos Saiago,  
*Álgebra Linear, 2ª edição.*  
*Escolar Editora, Lisboa, 2010.*
-  Seymour Lipschutz e Marc Lipson,  
*Álgebra Linear*  
*Coleção Schaum, 3ª edição.*  
Editora Bookman, Porto Alegre 2004.