

Estrutura de Dados

Filas

(Introdução a OO)

Prof^a. Dr^a. Alana Moraes
alanamm.prof@gmail.com

Noções Iniciais de Orientação a Objetos

Classe

Atributos

Métodos

Construtor

Objeto

Noções Iniciais de Orientação a Objetos

Classe

Atributos

Métodos

Construtor

Objeto

Noções Iniciais de Orientação a Objetos

Classe

Atributos - variáveis

Métodos

Construtor

Objeto

Noções Iniciais de Orientação a Objetos

Classe

Atributos

Métodos - funções

Construtor

Objeto

Noções Iniciais de Orientação a Objetos

Classe

Atributos

Métodos

Construtor - uma função especial

Objeto

Noções Iniciais de Orientação a Objetos

Classe

Atributos

Métodos

Construtor

Objeto - criado pelo construtor (instanciado pelo construtor)

Objetivo da aula

Entender o comportamento das filas

Recapitulando - Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Dados estruturados:

- Estáticos:
 - arrays;
 - registros;
 - arquivos;
 - conjuntos;
 - cadeias.
- Dinâmicos:
 - filas;
 - pilhas;
 - listas encadeadas;
 - árvores;
 - grafos.

Recapitulando - Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Dados estruturados:

- Estáticos:

- arrays;
- registros;
- arquivos;
- conjuntos;
- cadeias.

listas ou vetores

- Dinâmicos:

- filas;
- pilhas;
- listas encadeadas;
- árvores;
- grafos.

Recapitulando - Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Dados estruturados:

- Estáticos:

- arrays;
- registros;
- arquivos;
- conjuntos;
- cadeias.

listas ou vetores

- Dinâmicos:

- filas;
- pilhas;
- listas encadeadas;
- árvores;
- grafos.

Recapitulando - Estrutura de dados

Dados simples:

- padrão:
 - inteiro (int);
 - real (float);
 - caracter (str);
 - lógico (boolean).

Dados estruturados:

- Estáticos:

- arrays;
- registros;
- arquivos;
- conjuntos;
- cadeias.

listas ou vetores

- Dinâmicos:

- filas;
- pilhas;
- listas encadeadas;
- árvores;
- grafos.

O que são filas?



O que vem na minha mente ...



ingresso.com



OBRIGADO PELO SEU INTERESSE EM COMPRAR O ROCK IN RIO CARD 2015!

O Rock in Rio Card é limitado e está sujeito à disponibilidade. Devido a grande demanda, você está em uma fila virtual. Os primeiros da fila têm a prioridade na compra! Porém, tanto o lugar na fila como o acesso a página de compras não garantem a aquisição do Rock in Rio Card, já que os mesmos são limitados. Contamos com sua compreensão e solicitamos que aguarde até que seja transferido para a página de venda do Rock in Rio Card.

Não é necessário F5 ou atualizar a página!

O que são filas?



O que são filas?



Operações

- Criar fila
- Mostrar valores da fila
- Inserir valores
- Remover valores
- Definir tamanho da fila

Operações

- **Criar fila**
- Mostrar valores da fila
- Inserir valores
- Remover valores
- Definir tamanho da fila

Como criar uma fila em Python?




Criar fila

```
class Fila:  
  
    def __init__(self):  
        self.dados = [ ]
```

Criar fila

```
class Fila:  
  
    def __init__(self):  
        self.dados = [ ]
```



Classe construtora

Criar fila

```
class Fila:
```

```
    def __init__(self):
```

```
        self.dados = [ ]
```

Classe construtora



Variável local da classe construtora

Por que preciso usar uma classe para definir uma Fila?



Por que preciso usar uma classe para definir uma Fila?

Para acessar diversas vezes em vários códigos Python

Criar fila

#fila.py

```
class Fila:
```

```
    def __init__(self):  
        self.dados = [ ]
```

```
fila1 = Fila()
```

Criar fila

#fila.py

```
class Fila:

    def __init__(self):
        self.dados = [ ]

fila1 = Fila() #(modo 1)
```

#programa.py (modo 2)

```
import fila

fila1 = fila.Fila()

#OU

from fila import Fila

fila1 = Fila()
```

Criar fila

#fila.py

```
class Fila:

    def __init__(self):
        self.dados = [ ]

fila1 = Fila() # (modo 1)
```

#programa.py (modo 2)

```
import fila

fila1 = fila.Fila()

#OU

from fila import Fila

fila1 = Fila()
```

Escolha uma
das duas
formas!

Exercício A

Crie uma classe chamada Fila e inicialize essa classe:

a) dentro do arquivo Python da classe e

b) em outro arquivo Python

Operações

- Criar fila
- **Mostrar valores da fila**
- Inserir valores
- Remover valores
- Definir tamanho da fila

Mostrar valores da fila

#fila.py

```
class Fila:
```

```
    def __init__(self):  
        self.dados = [ ]
```

```
    def getFila(self):  
        return self.dados
```

```
//ou
```

```
    def __str__(self):  
        return str(self.dados)
```

Mostrar valores da fila

#fila.py

```
class Fila:

    def __init__(self):
        self.dados = [ ]

    def __str__(self):
        return str(self.dados)
```

Priorizarei utilizar esta notação

#programa.py (modo 2)

```
from fila import Fila

filaTeste = Fila()

print(filaTeste)
```

Exercício B

Crie uma classe chamada Fila com os seguintes valores: -1, -2, -3, -4. Em seguida, mostre esses valores na tela.

Operações

- Criar fila
- Mostrar valores da fila
- **Inserir valores**
- Remover valores
- Definir tamanho da fila

Inserir valores na fila

#fila.py

```
class Fila:
    def __init__(self):
        self.dados = [ ]

    def __str__(self):
        return str(self.dados)

    def inserirDado(self, novoValor):
        self.dados.append(novoValor)
```

Inserir valores na fila

#fila.py

```
class Fila:
    def __init__(self):
        self.dados = [ ]

    def __str__(self):
        return str(self.dados)

    def inserirDado(self,novoValor):
        self.dados.append(novoValor)
```

#programa.py

```
from fila import Fila

filaTeste = Fila()
print(filaTeste)

filaTeste.inserirDado(2)
print(filaTeste)
```

Exercício C

Crie uma classe chamada Fila. Em seguida, insira os seguintes valores na lista: “IFPB”, “ED-POO”, “2020.1”, “João Pessoa”.
Mostre os valores da lista antes e depois da inserção.

Operações

- Criar fila
- Mostrar valores da fila
- Inserir valores
- **Remover valores**
- Definir tamanho da fila

Remover valores da fila

“primeiro”	“segundo”	“terceiro”	“quarta”	“quinta”
------------	-----------	------------	----------	----------

Remover valores da fila



primeiro que
chegou na
fila

Remover valores da fila

fila.py

```
class Fila:
    def __init__(self):
        self.dados = [ ]

    def is_empty(self):
        return len(self.dados) == 0

    ...
```

Primeiro precisamos criar um método `is_empty()`

Remover valores da fila

fila.py

```
class Fila:
    def __init__(self):
        self.dados = [ ]
        ...

    def is_empty(self):
        return len(self.dados) == 0
```

```
    def remover(self):
        if (not self.is_empty()):
            self.dados.pop(0)
```

Sempre remove o primeiro elemento da fila!!!

Remover valores da fila

#fila.py

```
class Fila:
    def __init__(self):
        self.dados = [ ]
    ...

    def is_empty(self):
        return len(self.dados) == 0

    def remover(self):
        if (not self.is_empty()):
            self.dados.pop(0)
```

#programa.py

```
from fila import Fila

filaTeste = Fila()
filaTeste.inserirDado(0)
filaTeste.inserirDado(1)
filaTeste.inserirDado(2)
print(filaTeste)

filaTeste.remover()
print(filaTeste)
```

Exercício D

Utilize a Fila criada no exemplo anterior (“IFPB”, “ED-POO”, “2020.1”, “João Pessoa”) e remova um elemento da fila (aquele que chegou primeiro na fila).

Se quisermos remover um item específico da fila,
como implementar?



Remover valores da fila

“primeiro”	“segundo”	“terceiro”	“quarta”	“quinta”
------------	-----------	------------	----------	----------

Remover valores da fila

“primeiro”	“segundo”	“terceiro”	“quarta”	“quinta”
------------	-----------	------------	----------	----------



**quero remover
esse elemento da
fila**

Remover valores da fila

	“segundo”	“terceiro”	“quarta”	“quinta”
--	-----------	------------	----------	----------



**quero remover
esse elemento da
fila**

Remover valores da fila

		“terceiro”	“quarta”	“quinta”
--	--	------------	----------	----------



**quero remover
esse elemento da
fila**

Remover valores da fila



quero remover
esse elemento da
fila

Remover valores

#fila.py

```
class Fila:

    def __init__(self):
        self.dados = [ ]

    def is_empty(self):
        return len(self.dados) == 0

    def removerDado(self, valor):
        if (not self.is_empty()):
            pos = self.dados.index(valor)
            for i in range(0, pos + 1):
                self.dados.pop(0)
```

Remover valores

#fila.py

```
class Fila:

    def __init__(self):
        self.dados = [ ]

    def is_empty(self):
        return len(self.dados) == 0

    def removerDado(self, valor):
        if (not self.is_empty()):
            pos = self.dados.index(valor)
            for i in range(0, pos + 1):
                self.dados.pop(0)
```

#programa.py

```
from fila import Fila

filaTeste = Fila()
filaTeste.inserirDado(0)
filaTeste.inserirDado(1)
filaTeste.inserirDado(2)
print(filaTeste)

filaTeste.removerDado(1)
print(filaTeste)
```

Exercício E

Utilize a Fila (“IFPB”, “ED”, “2020.1”, “João Pessoa”) e remova o elemento “2020.1” da fila.

OBS: para que 2020.1 seja eliminado todos os valores que chegaram antes desse elemento também precisa ser removido.

Operações

- Criar fila
- Mostrar valores da fila
- Inserir valores
- Remover valores
- **Definir tamanho da fila**

Definir tamanho da fila

#fila.py

```
class Fila:

    def __init__(self):
        self.dados = [ ]

    def __len__(self):
        return len(self.dados)
```

Definir tamanho da fila

#fila.py

```
class Fila:

    def __init__(self):
        self.dados = [ ]

    def __len__(self):
        return len(self.dados)
```

#programa.py

```
from fila import Fila

filaTeste = Fila()
print(filaTeste)

filaTeste.inserirDado(1)
filaTeste.inserirDado(2)
filaTeste.inserirDado(3)

print(filaTeste)
print(filaTeste.__len__())
```

Exercício F

Informe o tamanho da fila (“IFPB”, “ED”, “2020.1”, “João Pessoa”, “Estágio 2”)

Operações Adicionais

Início da fila

```
def inicio(self):  
    if self.is_empty():  
        return ("Fila vazia")  
    else:  
        return self.dados[0]
```

Final da fila

```
def final(self):  
    if self.is_empty():  
        return ("Fila vazia")  
    else:  
        return self.dados[-1]
```

Dúvidas?



alanamm.prof@gmail.com