

Estrutura de Dados

Prof^a. Dr^a. Aline Marques
prof1658@iesp.edu.br

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

vetores



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.

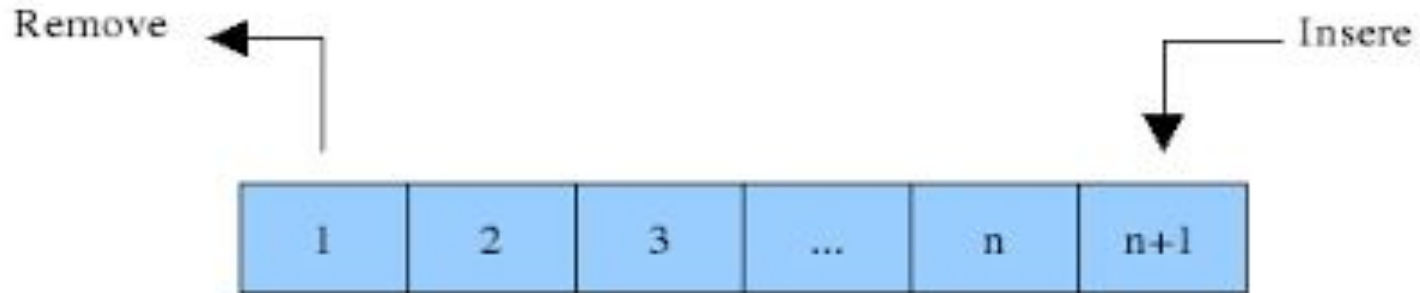
vetores



O que são filas?



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 = [ ]
```

são 2 underlines de cada lado!



Criar fila

```
class Fila():
```

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

Função construtora



Criar fila

```
class Fila():
```

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

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

Função construtora

Variável local da classe construtora

Criar fila

```
class Fila():
```

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

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

Função construtora

Função (método) de construção

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.

Por que preciso usar uma classe para definir uma Fila?

Para acessar diversas vezes em vários códigos Python.

Para manter o comportamento da estrutura consistente.

Criar fila

filaArq.py

```
class Fila():  
  
    def __init__(self):  
  
        self.dados = [ ]  
  
  
fila1 = Fila()
```

Criar fila

filaArq.py

```
class Fila():  
  
    def __init__(self):  
  
        self.dados = [ ]  
  
  
fila1 = Fila()
```

programa.py

```
from filaArq import Fila  
  
fila1 = Fila()
```

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

filaArq.py

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


Mostrar valores da fila

filaArq.py

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

programa.py

```
from filaArq import Fila  
  
  
filaTeste = Fila()  
  
print(filaTeste.getFila())
```

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

filaArq.py

```
class Fila():  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
def inserirDado(self, novoValor):  
    self.dados.append(novoValor)
```

Inserir valores na fila

filaArq.py

```
class Fila():  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
def inserirDado(self, novoValor):  
    self.dados.append(novoValor)
```

programa.py

```
from filaArq import Fila  
  
filaTeste = Fila()  
print(filaTeste.getFila())  
  
filaTeste.inserirDado(2)  
print(filaTeste.getFila())
```

Exercício C

Crie uma classe chamada `infos`. Em seguida, insira os seguintes valores na fila: “UNIESP”, “ED”, “2021.1”, “João Pessoa”, “Estágio 2”. 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

filaArq.py

```
class Fila():  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
    def removerDado(self):  
        self.dados.pop(0)  
  
    ...
```

Remover valores da fila

filaArq.py

```
class Fila():  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
    def removerDado(self):  
        self.dados.pop(0)  
  
    ...
```

programa.py

```
from filaArq import Fila  
  
filaTeste = Fila()  
filaTeste.inserirDado(0)  
filaTeste.inserirDado(1)  
filaTeste.inserirDado(2)  
print(filaTeste.getFila())  
  
filaTeste.removerDado()  
print(filaTeste.getFila())
```

Exercício D

Utilize a fila de infos criada no exemplo anterior (“UNIESP”, “ED”, “2021.1”, “João Pessoa”, “Estágio 2”) 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



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

filaArq.py

```
class Fila():  
  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
    def removeDado2(self, valor):  
        pos = self.dados.index(valor)  
        for i in range(0, pos+1):  
            self.dados.pop(0)
```

...

Remover valores

filaArq.py

```
class Fila():  
  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
    def removeDado2(self, valor):  
        pos = self.dados.index(valor)  
        for i in range(0, pos+1):  
            self.dados.pop(0)  
  
    ...
```

programa.py

```
from filaArq import Fila  
  
filaTeste = Fila()  
filaTeste.inserirDado(0)  
filaTeste.inserirDado(1)  
filaTeste.inserirDado(2)  
print(filaTeste.getFila())  
  
filaTeste.removeDado2(1)  
print(filaTeste.getFila())
```

Exercício E

Utilize a Fila (“UNIESP”, “ED”, “2021.1 “João Pessoa”, “Estágio 2”)
e remova o elemento “2021.1” da fila.

OBS: para que 2021.1 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

filaArq.py

```
class Fila():  
  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
    def tamanhoFila(self):  
        return len(self.dados)
```

...

Definir tamanho da fila

filaArq.py

```
class Fila():  
  
    def __init__(self):  
        self.dados = [ ]  
  
    def getFila(self):  
        return self.dados  
  
    def tamanhoFila(self):  
        return len(self.dados)
```

...

programa.py

```
from filaArq import Fila  
  
filaTeste = Fila()  
print(filaTeste.getFila())  
  
filaTeste.inserirDado(1)  
filaTeste.inserirDado(2)  
filaTeste.inserirDado(3)  
  
print(filaTeste.getFila())  
print(filaTeste.tamanhoFila())
```


Exercício F

Informe o tamanho da fila (“UNIESP”, “ED”, “2021.1”, “João Pessoa”, “Estágio 2”)

Nota 1

Manoel percebeu que seu estacionamento com uma entrada era um fiasco. Ele resolveu vender o seu terreno e comprar um novo que possui uma entrada e uma saída no fundo do terreno. Quando chega um novo carro, este é estacionado no terreno de Manoel, um atrás do outro. Quando um carro precisa sair, os carros do terreno são retirados pela saída, dão uma volta na quadra e são colocados no final da fila pela entrada do estacionamento. Faça um sistema que inclua carros no estacionamento informando o número da placa e retire carros usando o identificador (placa). Depois de ter informado a placa, cada vez que é pressionada a tecla S deve ser mostrado o estado do estacionamento.

Dúvidas?



prof1658@iesp.edu.br