
Estruturas de arquivos: estrutarq

Release 0.1

Jander Moreira

31 mar. 2022

Contents:

1	Módulo campo	1
1.1	Cadeias de caracteres	2
1.2	Inteiros	3
1.3	Reais	4
1.4	Tempo (datas e horas)	4
	Índice de Módulos Python	7
	Índice	9

class estrutarq.campo.campo_comum.CampoBasico(*tipo: str*)

Estruturação básica do campo como menor unidade de informação.

Parâmetros **tipo** (*str*) – cadeia de caracteres com o nome do tipo

abstract bytes_para_valor(*dado: bytes*)

Conversão de uma sequência de bytes para armazenamento no valor do campo, de acordo com a representação de dados :param dado: sequência de bytes :return: o valor do campo de acordo com seu tipo

comprimento()

Obtém o comprimento atual do campo

Retorna o comprimento do campo

comprimento_fixo()

Retorna se o comprimento é ou não fixo

Retorna *True* para comprimento fixo ou *False* para variável

copy()

Cópia “rasa” deste campo :return: outra instância com os mesmos valores

escreva(*arquivo: BinaryIO*)

Conversão do valor para sequência de bytes e armazenamento no arquivo

Parâmetros **arquivo** – arquivo binário aberto com permissão de escrita

leia(*arquivo: BinaryIO*)

Conversão dos dados lidos para o valor do campo, obedecendo à organização e formato de representação

Parâmetros **arquivo** – arquivo binário aberto com permissão de leitura

abstract property valor

Recuperação, com as devidas conversões, do atributo `__valor` :return: o valor de `__valor`

abstract `valor_para_bytes()` → `bytes`

Conversão do valor do campo para sequência de bytes de acordo com a representação de dados :return:

class `estrutarq.campo.campo_comum.CampoBruto(valor="")`

Implementação das funções de um campo bruto, ou seja, sem organização de campo. O valor é sempre armazenado como cadeia de caracteres.

bytes_para_valor(*dado: bytes*)

Conversão de sequência de bytes para valor o campo, considerando uma cadeia de caracteres simples :param dado: a sequência de bytes

property `valor`

Recuperação, com as devidas conversões, do atributo `__valor` :return: o valor de `__valor`

valor_para_bytes() → `bytes`

Conversão do valor do campo (cadeia de caracteres) para uma sequência de bytes, usando codificação UTF-8 :return: a sequência de bytes

1.1 Cadeias de caracteres

1.1.1 campo.campo_cadeia

Campos para armazenamento de cadeias de caracteres.

Este arquivo provê classes para uso de campos cujo conteúdo é uma cadeia de caracteres. Internamente, o tipo `str` é usado para armazenamento e a transformação para sequência de bytes usa a codificação UTF-8.

Uma classe básica `CampoCadeiaBasico` define uma classe abstrata (ABC) com as propriedades e métodos gerais. Dela são derivadas campos:

- Com terminadores
- Prefixada pelo comprimento
- De comprimento fixo predefinido

Jander Moreira, 2021-2022

Licença:

class `estrutarq.campo.campo_cadeia.CampoCadeiaBasico(tipo: str, valor: str = "")`

Classe básica para cadeias de caracteres.

Parâmetros

- **tipo** (`str`) – nome do tipo (definido nas classes derivadas)
- **valor** (`str`, *opcional*) – o valor a ser armazenado no campo (padrão: `""`)

bytes_para_valor(*dado: bytes*)

Armazenamento da sequência de bytes de dado como valor do campo.

Parâmetros **dado** (`bytes`) – sequência de bytes com codificação UTF-8

property `valor`

Recuperação, com as devidas conversões, do atributo `__valor` :return: o valor de `__valor`

valor_para_bytes() → bytes

Retorno do valor do campo convertido para sequência de bytes usando codificação UTF-8.

Retorna sequência de bytes

Tipo de retorno bytes

class estrutarq.campo.campo_cadeia.CampoCadeiaFixo(*comprimento: int, **kwargs*)

Classe para cadeia de caracteres com comprimento_bloco fixo e preenchimento de dados inválidos

class estrutarq.campo.campo_cadeia.CampoCadeiaPrefixado(**args, **kwargs*)

Classe para cadeia de caracteres prefixada pelo comprimento_bloco

class estrutarq.campo.campo_cadeia.CampoCadeiaTerminador(***kwargs*)

Classe para cadeia de caracteres com terminador

Parâmetros **kwargs** (:class:dict) – parâmetros nomeados a serem repassados

1.2 Inteiros

class estrutarq.campo.campo_inteiro.CampoIntBasico(*tipo: str, valor: int = 0*)

Classe básica para campo inteiro

bytes_para_valor(*dado: bytes*)

Conversão de uma sequência de bytes (representação textual) para inteiro :param dado: sequência de bytes

property valor: int

Recuperação, com as devidas conversões, do atributo __valor :return: o valor de __valor

valor_para_bytes() → bytes

Conversão do valor inteiro para sequência de bytes usando representação textual e codificação UTF-8 :return: sequência de bytes

class estrutarq.campo.campo_inteiro.CampoIntBinario(***kwargs*)

Classe para inteiro em formato binário (big endian) com 8 bytes e complemento para 2 para valores negativos

bytes_para_valor(*dado: bytes*)

Conversão de uma sequência de bytes (binária big-endian com sinal) para inteiro :param dado: sequência de bytes

valor_para_bytes() → bytes

Conversão do valor inteiro para sequência de bytes usando representação binária big-endian com sinal :return: sequência de bytes

class estrutarq.campo.campo_inteiro.CampoIntFixo(*comprimento: int, **kwargs*)

Classe para inteiro textual com tamanho fixo

class estrutarq.campo.campo_inteiro.CampoIntPrefixado(***kwargs*)

Classe para inteiro textual com prefixo de comprimento_bloco

class estrutarq.campo.campo_inteiro.CampoIntTerminador(*terminador: bytes = b'\x00', **kwargs*)

Classe para inteiro textual com terminador

1.3 Reais

class estrutarq.campo.campo_real.**CampoRealBasico**(*tipo: str, valor: float = 0*)

Classe básica para campo real

bytes_para_valor(*dado: bytes*)

Conversão de sequência de bytes com valor textual para valor real :param dado: sequência de 8 bytes

property valor: float

Recuperação, com as devidas conversões, do atributo `__valor` :return: o valor de `__valor`

valor_para_bytes() → *bytes*

Conversão do valor do campo para sequência de bytes textual :return: a sequência de bytes no padrão especificado

class estrutarq.campo.campo_real.**CampoRealBinario**(***kwargs*)

Classe para real em formato binário usando IEEE 754 de precisão dupla

bytes_para_valor(*dado: bytes*)

Conversão de sequência de bytes com representação IEEE 754 de precisão dupla para real :param dado: sequência de 8 bytes

valor_para_bytes() → *bytes*

Conversão do valor do campo para sequência de bytes no padrão IEEE 754 de precisão dupla :return: a sequência de bytes no padrão especificado

class estrutarq.campo.campo_real.**CampoRealFixo**(*comprimento: int, **kwargs*)

Classe para campo real com representação textual de comprimento_bloco fixo

class estrutarq.campo.campo_real.**CampoRealPrefixado**(***kwargs*)

Classe para campo real com representação textual de comprimento_bloco fixo

class estrutarq.campo.campo_real.**CampoRealTerminador**(*terminador: bytes = b'\x00', **kwargs*)

Classe para campo real com representação textual de comprimento_bloco fixo

1.4 Tempo (datas e horas)

class estrutarq.campo.campo_tempo.**CampoDataBinario**(***kwargs*)

Classe para armazenamento de data (dia, mês e ano) para armazenamento em formato binário.

class estrutarq.campo.campo_tempo.**CampoDataFixo**(***kwargs*)

Classe para data, em número de segundos desde 1/1/1970, 0h00min00s usando armazenamento em cadeia de caracteres no formato 'formato_data'.

class estrutarq.campo.campo_tempo.**CampoHoraBinario**(***kwargs*)

Classe para horário usando armazenamento em valor inteiro em binário, com sinal, big-endian.

class estrutarq.campo.campo_tempo.**CampoHoraFixo**(***kwargs*)

Classe horário usando armazenamento em cadeia de caracteres no formato 'formato_hora'.

class estrutarq.campo.campo_tempo.**CampoTempoBasico**(*tipo: str, formato: str, apenas_data: bool, valor: str = '', **kwargs*)

Classe básica para campo de tempo (data + horário), armazenado internamente como o número de segundos desde 1/1/1970, 0h00min00s.

Quando apenas a data é armazenada, o horário é ajustado para 12h00min00s, para evitar problemas com fuso horário.

property valor: `str`

Recuperação, com as devidas conversões, do atributo `__valor` :return: o valor de `__valor`

class `estrutarq.campo.campo_tempo.CampoTempoBasicoBinario(*args, **kwargs)`

Implementação das conversões tempo-> binário e binário->tempo

bytes_para_valor(*dado: bytes*)

Conversão da representação binária (8 bytes, big-endian, com sinal) para valor inteiro de segundos :param dado: bytes da representação do inteiro em binário

valor_para_bytes() → `bytes`

Conversão do valor do tempo em segundos para representação em inteiro binário (8 bytes, big-endian, com sinal) :return: a sequência de bytes

class `estrutarq.campo.campo_tempo.CampoTempoBasicoFixo(*args, **kwargs)`

Implementação das conversões tempo-> binário e binário->tempo

bytes_para_valor(*dado: bytes*)

Conversão da representação binária (8 bytes, big-endian, com sinal) para valor inteiro de segundos :param dado: bytes da representação do inteiro em binário

valor_para_bytes() → `bytes`

Conversão do valor do tempo em segundos para representação em inteiro binário (8 bytes, big-endian, com sinal) :return: a sequência de bytes

class `estrutarq.campo.campo_tempo.CampoTempoBinario(**kwargs)`

Classe para tempo (data + horário), em número de segundos desde 1/1/1970, 0h00min00s usando armazenamento em valor inteiro em binário, com sinal, big-endian.

class `estrutarq.campo.campo_tempo.CampoTempoFixo(**kwargs)`

Classe para tempo (data + horário), em número de segundos desde 1/1/1970, 0h00min00s usando armazenamento em cadeia de caracteres no formato 'formato_tempo'.

e

`estrutarq.campo.campo_cadeia`, 2
`estrutarq.campo.campo_comum`, 1
`estrutarq.campo.campo_inteiro`, 3
`estrutarq.campo.campo_real`, 4
`estrutarq.campo.campo_tempo`, 4

B

bytes_para_valor()	(método	estru-	tarq.campo.campo_cadeia.CampoCadeiaBasico),	2	CampoCadeiaTerminador	(classe	em	estru-	tarq.campo.campo_cadeia), 3
bytes_para_valor()	(método	estru-	tarq.campo.campo_comum.CampoBasico),	1	CampoDataBinario	(classe	em	estru-	tarq.campo.campo_tempo), 4
bytes_para_valor()	(método	estru-	tarq.campo.campo_comum.CampoBruto),	2	CampoDataFixo	(classe	em	estru-	tarq.campo.campo_tempo), 4
bytes_para_valor()	(método	estru-	tarq.campo.campo_inteiro.CampoIntBasico),	3	CampoHoraBinario	(classe	em	estru-	tarq.campo.campo_tempo), 4
bytes_para_valor()	(método	estru-	tarq.campo.campo_inteiro.CampoIntBinario),	3	CampoHoraFixo	(classe	em	estru-	tarq.campo.campo_tempo), 4
bytes_para_valor()	(método	estru-	tarq.campo.campo_real.CampoRealBasico),	4	CampoIntBasico	(classe	em	estru-	tarq.campo.campo_inteiro), 3
bytes_para_valor()	(método	estru-	tarq.campo.campo_real.CampoRealBinario),	4	CampoIntBinario	(classe	em	estru-	tarq.campo.campo_inteiro), 3
bytes_para_valor()	(método	estru-	tarq.campo.campo_tempo.CampoTempoBasicoBinario),	5	CampoIntFixo	(classe	em	estru-	tarq.campo.campo_inteiro), 3
bytes_para_valor()	(método	estru-	tarq.campo.campo_tempo.CampoTempoBasicoFixo),	5	CampoIntPrefixado	(classe	em	estru-	tarq.campo.campo_inteiro), 3
					CampoIntTerminador	(classe	em	estru-	tarq.campo.campo_inteiro), 3
					CampoRealBasico	(classe	em	estru-	tarq.campo.campo_real), 4
					CampoRealBinario	(classe	em	estru-	tarq.campo.campo_real), 4
					CampoRealFixo	(classe	em	estru-	tarq.campo.campo_real), 4
					CampoRealPrefixado	(classe	em	estru-	tarq.campo.campo_real), 4
					CampoRealTerminador	(classe	em	estru-	tarq.campo.campo_real), 4
					CampoTempoBasico	(classe	em	estru-	tarq.campo.campo_tempo), 4
					CampoTempoBasicoBinario	(classe	em	estru-	tarq.campo.campo_tempo), 5
					CampoTempoBasicoFixo	(classe	em	estru-	tarq.campo.campo_tempo), 5
					CampoTempoBinario	(classe	em	estru-	tarq.campo.campo_tempo), 5
					CampoTempoFixo	(classe	em	estru-	

C

CampoBasico	(classe	em	estru-	tarq.campo.campo_comum), 1
CampoBruto	(classe	em	estru-	tarq.campo.campo_comum), 2
CampoCadeiaBasico	(classe	em	estru-	tarq.campo.campo_cadeia), 2
CampoCadeiaFixo	(classe	em	estru-	tarq.campo.campo_cadeia), 3
CampoCadeiaPrefixado	(classe	em	estru-	

`tarq.campo.campo_tempo), 5`
`comprimento()` (método `estru-`
`tarq.campo.campo_comum.CampoBasico`),
`1`
`comprimento_fixo()` (método `estru-`
`tarq.campo.campo_comum.CampoBasico`),
`1`
`copy()` (método `estru-`
`tarq.campo.campo_comum.CampoBasico`),
`1`

E

`escreva()` (método `estru-`
`tarq.campo.campo_comum.CampoBasico`),
`1`
`estrutarq.campo.campo_cadeia`
`módulo, 2`
`estrutarq.campo.campo_comum`
`módulo, 1`
`estrutarq.campo.campo_inteiro`
`módulo, 3`
`estrutarq.campo.campo_real`
`módulo, 4`
`estrutarq.campo.campo_tempo`
`módulo, 4`

L

`leia()` (método `estru-`
`tarq.campo.campo_comum.CampoBasico`),
`1`

M

`módulo`
`estrutarq.campo.campo_cadeia, 2`
`estrutarq.campo.campo_comum, 1`
`estrutarq.campo.campo_inteiro, 3`
`estrutarq.campo.campo_real, 4`
`estrutarq.campo.campo_tempo, 4`

V

`valor` (propriedade `estru-`
`tarq.campo.campo_cadeia.CampoCadeiaBasico`
) , `2`
`valor` (propriedade `estru-`
`tarq.campo.campo_comum.CampoBasico`
) , `1`
`valor` (propriedade `estru-`
`tarq.campo.campo_comum.CampoBruto`),
`2`
`valor` (propriedade `estru-`
`tarq.campo.campo_inteiro.CampoIntBasico`),
`3`

`valor` (propriedade `estru-`
`tarq.campo.campo_real.CampoRealBasico`
) , `4`
`valor` (propriedade `estru-`
`tarq.campo.campo_tempo.CampoTempoBasico`
) , `5`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_cadeia.CampoCadeiaBasico`),
`2`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_comum.CampoBasico`),
`1`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_comum.CampoBruto`),
`2`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_inteiro.CampoIntBasico`),
`3`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_inteiro.CampoIntBinario`),
`3`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_real.CampoRealBasico`),
`4`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_real.CampoRealBinario`),
`4`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_tempo.CampoTempoBasicoBinario`),
`5`
`valor_para_bytes()` (método `estru-`
`tarq.campo.campo_tempo.CampoTempoBasicoFixo`),
`5`