

The Jack OS API

A linguagem Jack vem com uma coleção de oito classes internas (built-in) que estendem as capacidades da linguagem. Esta biblioteca padrão (standard library), que também pode ser vista como um sistema operacional básico, inclui as seguintes classes, todas implementadas em Jack:

Math

Esta classe disponibiliza várias operações matemáticas.

function int **abs**(int x): retorna o valor absoluto de x.

function int **multiply**(int x, int y): retorna o produto de x e y.

function int **divide**(int x, int y): retorna a parte inteira de x/y.

function int **min**(int x, int y): retorna o menor valor entre x e y.

function int **max**(int x, int y): retorna o maior valor entre x e y.

function int **sqrt**(int x): retorna a parte inteira da raiz quadrada de x.

String

Esta classe implementa o tipo de dado String (usado para armazenar textos) e operações relacionadas.

constructor String **new**(int maxLength): cria String vazia, podendo conter até maxLength caracteres.

method void **dispose**(): libera memória da String.

method int **length**(): retorna o comprimento da String.

method char **charAt**(int j): retorna o caractere na posição j da String.

method void **setCharAt**(int j, char c): Coloca o caractere c na posição j da String.

method String **appendChar**(char c): adiciona o caractere c na String e retorna o ponteiro da String.

method void **eraseLastChar**(): elimina o último caractere da String.

method int **intValue**(): retorna o valor em inteiro do texto da String.

method void **setInt**(int j): faz a String ficar com o texto do valor de j.

function char **backSpace**(): retorna o caractere de backspace.

function char **doubleQuote**(): retorna o caractere de aspas (").

function char **newLine**(): retorna o caractere de nova linha.

Array

Esta classe permite a criação e liberação de vetores.

function Array **new**(int size): constrói um novo vetor do tamanho especificado em size.

method void **dispose**(): libera memória desse vetor.

Output

Esta classe permite escrever textos na tela.

function void **moveCursor**(int i, int j): move o cursos para a coluna j e linha i, e apaga o caractere lá.

function void **printChar**(char c): imprime c na posição do cursos e avança uma casa a frente.

function void **printString**(String s): mostra String s no cursor, avançando o cursor posteriormente.

function void **printInt**(int i): imprime i a partir da posição do cursos avançando da forma apropriada.

function void **println**(): avança o cursos para o início da próxima linha.

function void **backSpace**(): move o cursos uma coluna para traz.

Screen

Esta classe permite desenhos gráficos na tela. Índice de colunas começa no 0 e vão da esquerda para a direita. Índices de linhas começam no 0 e vão de cima para baixo.

function void **clearScreen**(): limpa a tela toda.

function void **setColor**(boolean b): define a cor (branco=false, preto=true) para comandos gráficos.

function void **drawPixel**(int x, int y): desenha um pixel em (x,y).

function void **drawLine**(int x1, int y1, int x2, int y2): desenha uma linha de (x1,y1) até (x2,y2).

function void **drawRectangle**(int x1, int y1, int x2, int y2): desenha retângulo do (x1, y1) até (x2,y2).

function void **drawCircle**(int x, int y, int r): desenha um círculo em (x1, y1) com r de raio.

Keyboard

Esta classe permite ler as entradas do teclado.

function char **keyPressed**(): retorna o caractere da tecla pressionada, ou 0 se nenhuma pressionada.

function char **readChar**(): aguarda que tecla seja pressionada e solta, imprime na tela o caractere.

function String **readLine**(String message): exibe mensagem na tela, lê uma linha de texto do teclado mostrando o valor, e retorna o valor da função. Essa função também trata backspace.

function int **readInt**(String message): imprime a mensagem na tela, lê uma linha de texto do teclado até o primeiro caractere que não seja um dígito, imprimindo também o valor, e retorna o valor da função. Essa função também trata backspace.

Memory

Esta classe permite acessar diretamente a memória do computador.

function int **peek**(int address): retorna o valor da memória no endereço informado.

function void **poke**(int address, int value): coloca o valor value no endereço de memória fornecido em addresss.

function Array **alloc**(int size): encontra e aloca na memória heap, um bloco de memória suficiente para o tamanho pedido em size e retorna uma referência para o endereço base.

function void **deAlloc**(Array o): Libera um objeto e libera seu espaço de memória.

Sys

Esta classe suporta alguns recursos relacionados a execução dos serviços básicos de execução.

function void **halt**(): paralisa o programa sendo executado.

function void **error**(int errorCode): imprime o código de erro na tela.

function void **wait**(int duration): espera pelo número de milissegundos especificado em duration.

Código de Erros do Jack OS

Método/Função	Descrição
1 Sys.wait	Duração deve ser positiva.
2 Array.new	Tamanho do vetor deve ser positiva.
3 Math.divide	Divisão por zero.
4 Math.sqrt	Não calcula raiz quadrada de um número negativo
5 Memory.alloc	Aloca o tamanho de memoria, que deve ser positivo
6 Memory.alloc	Estouro de pilha.
7 Screen.drawPixel	Coordenada do pixel é ilegal.
8 Screen.drawLine	Coordenada do pixel é ilegal.
9 Screen.drawRectangle	Coordenada do pixel é ilegal.
12 Screen.drawCircle	Centro de coordenadas do pixel é ilegal.
13 Screen.drawCircle	Coordenada do círculo é ilegal.
14 String.new	Comprimento máximo deve ser não negativo.
15 String.charAt	Índice da String esta fora dos limites.
16 String.setCharAt	Índice da String esta fora dos limites.
17 String.appendChar	String esta cheia
18 String.eraseLastChar	String esta vazia
19 String.setInt	Insuficiente capacidade de String
20 Output.moveCursor	Posição do cursor ilegal