The Jack OS API

The Jack language comes with a collection of eight built-in classes that extend the language's capabilities. This standard library can be viewed as a basic operating system. This document gives the OS API.

Math

A library of commonly used mathematical functions.

`function int multiply(int x, int y)`: returns the product of `x` and `y`. When a Jack compiler detects the multiplication operator '`*`' in the program's code, it handles it by invoking this method. In other words, the Jack expressions `x*y` and `Math.multiply(x,y)` return the same value.

`function int divide(int x, int y)`: returns the integer part of `x/y`. When a Jack compiler detects the division operator '`/`' in the program's code, it handles it by invoking this method. In other words, the Jack expressions `x/y` and `Math.divide(x,y)` return the same value.

`function int min(int x, int y)`: returns the minimum of `x` and `y`.

`function int max(int x, int y)`: returns the maximum of `x` and `y`.

`function int sqrt(int x)`: returns the integer part of the square root of `x`.

String

The String class represents character strings. In addition for constructing and disposing strings, the class features methods for getting and setting individual characters of the string, for erasing the string's last character, for appending a character to the string's end, and more typical string-oriented operations.

`constructor String new(int maxLength)`: constructs a new empty string with a maximum length of `maxLength` and initial length of 0.

`method int dispose()`: disposes this string.

`method int length()`: returns the current length of this string.

`method char charAt(int j)`: returns the character at the `j`-th location of this string.

`method void setCharAt(int j, char c)`: sets the character at the `j`-th location of this string to `c`.

`method String appendChar(char c)`: appends `c` to this string's end and returns this string.

`method void eraseLastChar()`: erases the last character from this string.

`method int intValue()`: returns the integer value of this string, until a non-digit char is detected.

`method void setInt(int val)`: sets this string to hold a representation of the given value.

`function char backSpace()`: returns the backspace character.

`function char doubleQuote()`: returns the double quote (`"`) character.

`function char newLine()`: returns the newline character.

Array

Represents an array. In the Jack language, arrays are instances of the `Array` class. Once declared, the array entries can be accessed using the usual syntax `arr[i]`. Each array entry can hold a primitive data type or any object type. Different array entries can have different data types.

`function Array new(int size)`: constructs a new array of the given size.

`method void dispose()`: disposes this array.

Output

A library of functions for displaying text on the screen.

The Hack physical screen consists of 512 rows of 256 pixels each. The library uses a fixed font, in which each character is displayed within a frame which is 11 pixels high (including 1 pixel for inter-line spacing) and 8 pixels wide (including 2 pixels for inter-character spacing). The resulting grid accommodates 23 rows (indexed 0..22, top to bottom) of 64 characters each (indexed 0..63, left to right). The top left character position on the screen is indexed (0,0). A cursor, implemented as a small filled square, indicates where the next character will be displayed.

`function void moveCursor(int i, int j):` moves the cursor to the `j`-th column of the `i`-th row, and erases the character displayed there.

`function void printChar(char c):` displays the given character at the cursor location, and advances the cursor one column forward.

`function void printString(String s)`: displays the given string starting at the cursor location, and advances the cursor appropriately.

`function void printInt(int i)`: displays the given integer starting at the cursor location, and advances the cursor appropriately.

`function void println():` advances the cursor to the beginning of the next line.

`function void backSpace()`: moves the cursor one column back.

Screen

A library of functions for displaying graphics on the screen. The Hack physical screen consists of 512 rows (indexed 0..511, top to bottom) of 256 pixels each (indexed 0..255, left to right). The top left pixel on the screen is indexed (0,0).

`function void clearScreen()`: erases the entire screen.

`function void setColor(boolean b)`: sets the current color, to be used for all subsequent `drawXXX` commands. Black is represented by `true`, white by `false`.

`function void drawPixel(int x, int y):` draws the (`x`,`y`) pixel, using the current color.

`function void drawLine(int x1, int y1, int x2, int y2)`: draws a line from pixel (`x1`,`y1`) to pixel (`x2`,`y2`), using the current color.

`function void drawRectangle(int x1, int y1, int x2, int y2)`: draws a filled rectangle whose top left corner is (`x1`,`y1`) and bottom right corner is (`x2`,`y2`), using the current color.

`function void drawCircle(int x, int y, int r)`: draws a filled circle of radius $r \leq 181$ around (`x`,`y`), using the current color.

Keyboard

This class allows reading inputs from a standard keyboard.

`function char keyPressed()`: returns the character of the currently pressed key on the keyboard; if no key is currently pressed, returns 0. Recognizes all ASCII characters, as well as the following keys: `newline` (128=`String.newline()`), `backspace` (129=`String.backspace()`), `left arrow` (130), `up arrow` (131), `right arrow` (132), `down arrow` (133), `home` (134), `end` (135), `page up` (136), `page down` (137), `insert` (138), `delete` (139), `ESC` (140), `F1-F12` (141-152).

`function char readChar()`: waits until a key is pressed on the keyboard and released, then echoes the key to the screen and returns the character of the pressed key.

`function String readLine(String message)`: displays the message on the screen, reads from the keyboard the entered text until a `newline` character is detected, echoes the text to the screen, and returns its value. Also handles user backspaces.

`function int readInt(String message):` displays the message on the screen, reads from the keyboard the entered text until a `newline` character is detected, echoes the text to the screen, and returns its integer value (until the first non-digit character in the entered text is detected). Also handles user backspaces.

Memory

This library provides two services: direct access to the computer's main memory (`RAM`), and allocation and recycling of memory blocks. The Hack `RAM` consists of 32,768 words, each holding a 16-bit binary number.

`function int peek(int address):` returns the `RAM` value at the given address.

`function void poke(int address, int value):` sets the `RAM` value at the given address to the given value.

`function Array alloc(int size)`: finds an available `RAM` block of the given size and returns a reference to its base address.

`function void deAlloc(Array o)`: de-allocates the given object (cast as an array) by making it available for future allocations.

Sys

A library that supports various program execution services.

`function void halt()`: halts the program execution.

`function void error(int errorCode)`: displays the given error code in the form "`ERR<errorCode>`", and halts the program's execution.

`function void wait(int duration)`: waits approximately `duration` milliseconds and returns.