Appendix 6: The Jack OS API

The Jack language is supported by eight standard classes that provide basic OS services like memory allocation, mathematical functions, input capturing, and output rendering. This appendix documents the API of these classes.

Math

This class provides commonly needed 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 function. Thus the Jack expressions x * y and the function call 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 function. Thus the Jack expressions x / y and the function call 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

This class represents strings of char values and provides commonly needed string processing services.

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

method void dispose(): Disposes this string.

method int length(): Returns the number of characters in this string.

method char charAt(int i): Returns the character at the i-th location of this string.

method void setCharAt(int i, char c): Sets the character at the i-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 character 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

In the Jack language, arrays are implemented as instances of the OS class Array. Once declared, the array elements can be accessed using the syntax arr[i]. Jack arrays are not typed: each array element can hold a primitive data type or an object type, and different elements in the same array can have different types.

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

method void dispose(): Disposes this array.

Output

This class provides functions for displaying characters. It assumes a character-oriented screen consisting of 23 rows (indexed 0…22, top to bottom) of 64 characters each (indexed 0…63, left to right). The top-left character location on the screen is indexed (0,0). Each character is displayed by rendering on the screen a rectangular image 11 pixels high and 8 pixels wide (which includes margins for character spacing and line spacing). If needed, the bitmap images ("font") of all the characters can be found by inspecting the given code of the Output class. A visible 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 overrides the character displayed there.

function void printChar(char c): Displays the character at the cursor location and advances the cursor one column forward.

function void printString(String s): Displays the string starting at the cursor location and advances the cursor appropriately.

function void printInt(int i): Displays the 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

This class provides functions for displaying graphical shapes on the screen. The Hack physical screen consists of 256 rows (indexed 0…255, top to bottom) of 512 pixels each (indexed 0…511, 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. This color will be used in all the subsequent draw*Xxx* function calls. 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 ≤ 181 around (x,y) using the current color.

Keyboard

This class provides functions for 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 the values in the Hack character set (see appendix 5). These include the characters newLine (128, return value of String.newLine()), backSpace (129, return value of String.backSpace ()), leftArrow (130), upArrow (131), rightArrow (132), downArrow (133), home (134), end (135), pageUp (136), pageDown (137), insert (138), delete (139), esc (140), and f1–f12 (141–152).

function char readChar(): Waits until a keyboard key is pressed and released, then displays the corresponding character on the screen and returns the character.

function String readLine(String message): Displays the message, reads from the keyboard the entered string of characters until a newLine character is detected, displays the string, and returns the string. Also handles user backspaces.

function int readInt(String message): Displays the message, reads from the keyboard the entered string of characters until a newLine character is detected, displays the string on the screen, and returns its integer value until the first non-digit character in the entered string is

detected. Also handles user backspaces.

Memory

This class provides memory management services. The Hack RAM consists of 32,768 words, each holding a 16-bit binary number.

function int peek(int address): Returns the value of RAM[address].

function void poke(int address, int value): Sets RAM[address] to the given value.

function Array alloc(int size): Finds an available RAM block of the given size and returns its base address.

function void deAlloc(Array o): Deallocates the given object, which is cast as an array. In other words, makes the RAM block that starts in this address available for future memory allocations.

Sys

This class provides basic program execution services.

function void halt(): Halts the program execution.

function void error(int errorCode): Displays the error code, using the format ERR<errorCode>, and halts the program's execution.

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