```
class Keyboard {
  /** Initializes the keyboard. */
  function void init() {
    return;
  }

  /**
   * Returns the ASCII code (as char) of the currently pressed key,
   * or 0 if no key is currently pressed.
   * Recognizes all ASCII characters, as well as the following extension
   * of action keys:
   * New line = 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 keyPressed() {
    return Memory.peek(24576);
  }

  /**
   * Reads the next character from the keyboard.
   * waits until a key is pressed and then released, then echoes
   * the key to the screen, and returns the value of the pressed key.
   */
  function char readChar() {
    var char c;

    do Output.printChar(0); // cursor

    while (~(Keyboard.keyPressed())) {}
    let c = Keyboard.keyPressed();
    while (Keyboard.keyPressed() = c) {}

    do Output.backSpace(); // remove cursor
    do Output.printChar(c);

    return c;
  }

```

```
  /**
   * Prints the message on the screen, reads the next line
   * (until a newline character) from the keyboard, and returns its value.
   */
  function String readLine(String message) {
    var String s;
     var char c;

     do Output.printString(message);

     let s = String.new(20);

     while (true) {
       let c = Keyboard.readChar();

       if (c = String.newLine()) {
         do Output.println();
         return s;
       } else {
         if (c = String.backSpace()) {
           do s.eraseLastChar();
         } else {
           let s = s.appendChar(c);
         }
       }
     }

     return s;
  }

  /**
   * Prints the message on the screen, reads the next line
   * (until a newline character) from the keyboard, and returns its
   * integer value (until the first non numeric character).
   */
  function int readInt(String message) {
    var String s;

    let s = Keyboard.readLine(message);

    return s.intValue();
  }
}
```