```
class String {
  field Array string_array;
  field int string_end, max_length;

  /** Constructs a new empty String with a maximum length of maxLength. */
  constructor String new(int maxLength) {
    if (maxLength < 0) {
      do Sys.error(14); // String.new: Maximum length must be non-negative
    }

    let string_end = 0;
    let max_length = maxLength + 1;
    let string_array = Array.new(max_length);

    return this;
  }

  /** De-allocates the string and frees its space. */
  method void dispose() {
    do string_array.dispose();
    do Memory.deAlloc(this);

    return;
  }

  /** Returns the current length of this String. */
  method int length() {
    return string_end;
  }

  /** Returns the character at location j. */
  method char charAt(int j) {
    if ((j < 0) | (j > (string_end - 1))) {
      do Sys.error(15); // String.charAt: String index out of bounds
    }

    return string_array[j];
  }

  /** Sets the j'th character of this string to be c. */
  method void setCharAt(int j, char c) {
    if ((j < 0) | (j > (string_end - 1))) {
      do Sys.error(16); // String.setCharAt: String index out of bounds
    }

    let string_array[j] = c;

    return;
  }
```

```
  /** Appends the character c to the end of this String.
   *  Returns this string as the return value. */
  method String appendChar(char c) {
    if (string_end = max_length) {
      do Sys.error(17); // String.appendChar: String is full
    }

    let string_array[string_end] = c;
    let string_end = string_end + 1;

    return this;
  }

  /** Erases the last character from this String. */
  method void eraseLastChar() {
    if (string_end = 0) {
      do Sys.error(18); // String.eraseLastChar: String is empty
    }

    let string_end = string_end - 1;
    let string_array[string_end] = 0;

    return;
  }

  /** Returns the integer value of this String until the first non
   *  numeric character. */
  method int intValue() {
    var boolean negative;
    var int number, i, digit;

    let number = 0;
    let i = 0;

    while (i < string_end) {
      let digit = string_array[i] - 48;

      if (string_array[i] = 45) { // -
        let negative = true;
      } else {
        let number = (number * 10) + digit;
      }

      let i = i + 1;
    }

    if (negative) {
      return -number;
    }
```

```
    return number;
  }

  /** Sets this String to hold a representation of the given number. */
  method void setInt(int number) {
    var int first_digit, exp;

    let string_end = 0;

    if (number < 0) {
      let number = -number;

      if (string_end = max_length) {
        do Sys.error(19); // String.setInt: Insufficient string capacity
      }

      do appendChar(45);
    }

    while (number > 0) {
      if (string_end = max_length) {
        do Sys.error(19); // String.setInt: Insufficient string capacity
      }

      let exp = 1;
      let first_digit = number;

      while (first_digit > 9) {
        let first_digit = first_digit / 10;
        let exp = exp * 10;
      }

      let number = number - (exp * first_digit);
      do appendChar(first_digit + 48);
    }

    return;
  }

  /** Returns the new line character. */
  function char newLine() {
    return 128;
  }

  /** Returns the backspace character. */
  function char backSpace() {
    return 129;
  }

```

```
  /** Returns the double quote (") character. */
  function char doubleQuote() {
    return 34;
  }
}

```