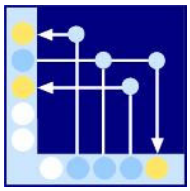
Fakultät Elektronik und Informatik
Studienbereich Informatik



Advanced Programming with MOSTflexiPL

Lecture in Wintersemester 2025/2026
Prof. Dr. habil. Christian Heinlein

9. Task Sheet (January 13, 2026)

Task 15: Strings

Define a type `str` for the representation of strings as well as the following operators:

- For any number of characters `c ...` of type `char`, `str c ...` returns a string consisting of these characters.

  (According to §4.5, this operator is implicitly passed to string literals.)

- For a string `s` of type `str`, `#s` returns its length, i. e., the number of its characters.

  If `s` has neither been created by a string literal (i. e., indirectly by the previously described operator) nor by any of the operators described in the sequel, `#s` returns 0.

  This operator has the same precedence as the predefined change sign operator.

- For a string `s` of type `str` and an `int` value `i`, `s[i]` returns the `i`-th character of `s` if `i` is between 1 and `#s` (both inclusively). Otherwise, nil is returned.

  This operator has the same precedence as the predefined change sign operator, too.

- For a string `s` of type `str`, `for name in s do ... end` successively evaluates the loop body `...` for every character of `s`, where the current character is available as a `char` constant with the name `name` inside the loop body.

  The result value is the number of iterations, i. e., the length of `s`.

- For a string `s` of type `str`, `print s` or `print only s` prints the string `s` with or without a terminating line separator, respectively.

- For an `int` value `i`, `str i` returns the string representation of `i` as a decimal number, which starts with a minus sign for a negative value.

  For an unnatural value `i`, an empty string is returned.

- For strings `s1` and `s2`, `s1 + s2` returns the concatenation of `s1` and `s2` as a new string.

  For a string `s` with type `str` and a value `x` of basically any type `X`, `s + x` (or `x + s`) returns the concatenation of `s` with the string representation of `x` (or vice versa, respectively), which is obtained by applying an implicitly passed operator `str (X) -> (str)` to `x`.

  These operators have the same binding properties as the predefined addition.