Memory Class Documentation

Source File: Memory.h
Class Header: `class Memory : public Object`
Namespace: `cal`

Overview

The *Memory* class represents a named computer memory.

Constructors

- `Memory()` (default constructor)
  - **Purpose:** Creates memory cell with the name '`name`', a content of zero, and deactivated.
- `Memory(const Memory& obj)` (copy constructor)
  - **Purpose:** Constructs a deep copy of *obj*.
  - **Parameter(s):**
    - *obj*: Constant *Memory* reference object.

Destructor

- `~Memory()` [virtual]
  - **Purpose:** Does nothing.

Assignment Operators

- `operator=(const Memory& rhs)`
  - **Purpose:** Constructs a deep copy of *rhs*.
  - **Parameter(s):**
    - *rhs*: Constant *Memory* reference object.
  - **Return:** `*this`.

Methods

- `read()`
  - **Purpose:** retrieves the content of the memory and deactivates the memory cell.
  - **Return:** a string.
- `write(string obj)`
  - **Purpose:** assigns *obj* to the content of the memory cell and activates the cell if *obj* is a binary string.
  - **Parameters:**
    - *obj*: A string.
- `active() const`
  - **Purpose:** checks if the memory cell is active.
  - **Return:** a Boolean.
- `change(bool obj)`
  - **Purpose:** changes the status of the memory cell.
  - **Parameters:**
    - *obj*: A Boolean.
- `toString() const` [overridden]
  - **Purpose:** Provides a string representation of the *Memory* object.
  - **Return:** A string in the format: *name* : *content*.

Non-Member Functions

- `operator>>(istream& ins,Memory& obj)`
  - **Purpose:** Overloads the istream (input stream) operator to allow writing into the class object.
  - **Parameters:**
    - *ins*: Reference of an istream object.
    - *obj*: Reference of an *Mask* object.
  - **Return:** *ins*.
- `isBinary(string obj)`
  - **Purpose:** Checks if *obj* is a binary string.
  - **Parameters:**
    - *obj*: A string object.
  - **Return:** A Boolean.
- `isHex(string obj)`
  - **Purpose:** Checks if *obj* is a hexadecimal string.
  - **Parameters:**
    - *obj*: A string object.
  - **Return:** A Boolean.
- `toDecimal(string obj)`
  - **Purpose:** Converts *obj* to a integer if *obj* is a binary string.
  - **Parameters:**
    - *obj*: A string object.
  - **Return:** An integer.
- `toBinary(int obj)`
  - **Purpose:** Converts the absolute value of *obj* to a binary string.
  - **Parameters:**
    - *obj*: An integer.
  - **Return:** A binary string.
- `negate(string obj)`
  - **Purpose:** Retrieves the twos-complement of *obj* if *obj* is a binary string.
  - **Parameters:**
    - *obj*: A string object.
  - **Return:** A string.