Array Class Documentation

Source File: Util.h
Namespace: ds
Class Header: `template <class T> class Array : public Object`

Overview

The *Array* class is a container class for a dynamic generic array.

Constructors

- `Array()` (default constructor)
  - **Purpose:** Allocates and initializes a dynamic generic array to size 30 and the default value of the generic type.
- `Array(const Array<T>& obj)` (copy constructor)
  - **Purpose:** Constructs a deep copy of *obj*.
  - **Parameter(s):**
    - *obj*: Constant *Array* reference object.
- `Array(unsigned long sz)`
  - **Purpose:** Allocates and initializes a dynamic generic array to size *sz* and the default value of the generic type. However, if *sz* is 0, the array is allocated to 30.
  - **Parameter(s):**
    - *sz*: A possible size for the generic dynamic array.
- `Array(const T& value, unsigned long sz)`
  - **Purpose:** Allocates and initializes a dynamic generic array to size *sz* and *value*. However, if *sz* is 0, the array is allocated to 30.
  - **Parameter(s):**
    - *value*: A default value of the elements.
    - *sz*: A possible size for the generic dynamic array.
- `Array(initializer_list<T> lst)`
  - **Purpose:** Allocates and initializes a dynamic generic array to the size and elements of *lst*.
  - **Parameter(s):**
    - *lst*: A list of elements of the generic type.

Destructor

- `~Array()`
  - **Purpose:** Deallocates the generic array.

Assignment Operators

- `operator=(const Array<T>& rhs)`
  - **Purpose:** Constructs a deep copy of *rhs*.
  - **Parameter(s):**
    - *rhs*: Constant *Array* reference object.
  - **Return:** `*this`.
- `operator=(initializer_list<T> lst)`
  - **Purpose:** Allocates and initializes a dynamic generic array to the size and elements of *lst*.
  - **Parameter(s):**
    - *lst*: A list of elements of the generic type.
  - **Return:** `*this`.

Methods

- `Size() const`
`Length() const`
    - **Purpose:** Gets the capacity of the generic dynamic array.
    - **Return:** The capacity of the generic array.
- `operator[](unsigned int idx) const`
`operator[](unsigned int idx)`
    - **Purpose:** Retrieves an element of the generic dynamic array with the index *idx*.
    - **Parameter(s):**
        - *idx*: A possible index of the generic dynamic array.
    - **Return:** A (constant) reference of an element of the generic dynamic array.
    - **Exception(s):**
        - `Out-Of-Bound Error`: Thrown if *idx* exceeds or equals the capacity of the generic dynamic array.
- `ToString() const override`
    - **Purpose:** Provides a string representation of the *Array* object.
    - **Return:** A string representation of the elements of the generic dynamic array all enclosed within square braces.

Example:

```
#include <iostream>
#include "Array.h"

int main()
{
  ds::Array<int> a, b = {2,3,4,5}, c = b, d(10); //constructor calls

  for(int i = 0;i < a.Size();i += 1)
  {
    a[i] = 2 * b[i % b.Length()] + i;
  }
  d = a;
  std::cout << d << "\n";
  return 0;
}
```