

Assignment: Array class.

Anand Kamble

amk23j@fsu.edu

17th October 2023

The C++ program introduces a dynamic array class named "Array," complemented by a set of test cases designed to validate its functionality.

Code Modules

The code is organized in the following modules:

- `Array.h/.cpp` : Array class definitions and methods, which allow us to manipulate the dynamic array.
- `Main_test.cpp`: The main function which implements the Array class and tests its methods.
- `Test.cpp` : The 'Test' class which is used to test the any values with expected values.

Formatting Conventions:

- a. Casing: Camel-Case
 - b. Indentation: 4 spaces
 - c. Line Break: CRLF
-

Array Class

Constructors and Destructor:

- Default Constructor (`Array()`): Initializes an empty array.
- Parameterized Constructor (`Array(int size)`): Initializes the array with a specified size. [1]
- Copy Constructor (`Array(const Array &)`): Enables the creation of a new object by deep copying an existing one.
- Destructor (`~Array()`): Frees allocated memory upon the object's destruction.

Assignment Operator (`operator=`):

- Overloaded to facilitate proper assignment between Array objects.

Element Manipulation:

- `push_back(int value)`: Adds an element to the end of the array.
- `pop_back()`: Removes the last element from the array.
- `remove(int index)`: Removes the element at the specified index.
- `insert(int value, int index)`: Inserts a value at the specified index.

Accessors:

- `getCapacity() const`: Retrieves the current capacity of the array.
- `size() const`: Retrieves the number of elements in the array.

Utility Functions:

- `clear()`: Resets the array by deallocating memory and setting the number of elements to zero.

Private Members:

- `numberOfElements`: Tracks the current number of elements in the array.
 - `capacity`: Indicates the current capacity of the array.
-

Test Class [2] [3]

Private Members:

- `name`: A string that identifies the test case.
- `passed`: A boolean flag indicating whether the test has passed or failed.
- `outputFloat`: A floating-point value for testing.
- `outputInt`: An integer value for testing.
- `outputString`: A string value for testing.
- `outputChar`: A pointer to a character array for testing.

Utility Functions:

- `result()`: Outputs the test result (passed or failed) along with the test name.

Constructors:

- `Test(string name)`: Initializes a test case with a given name.

Assertion Functions:

- `Expect(float value)`: Sets the expected value for a floating-point test.
 - `ToBe(float valueToBeExpected)`: Compares the actual and expected floating-point values and outputs the result.
 - `Expect(int value)`: Sets the expected value for an integer test.
 - `ToBe(int valueToBeExpected)`: Compares the actual and expected integer values and outputs the result.
 - `Expect(string value)`: Sets the expected value for a string test.
 - `ToBe(string valueToBeExpected)`: Compares the actual and expected string values and outputs the result.
 - `Expect(const char *value)`: Sets the expected value for a character array test.
 - `ToBe(const char *valueToBeExpected)`: Compares the actual and expected character array values and outputs the result.
-

Execution

The Makefile is included with this code. You can run the command 'make' to compile the program. After successful compilation, you can find the executable named 'test.x' inside the bin folder. Run this executable by './bin/test.x'. To clean the generated folders and files, use the command 'make clean'.

References

- [1] OpenAI, "ChatGPT," OpenAI, 17 10 2023. [Online]. Available: <https://chat.openai.com/share/88405562-d86b-4f45-a6d5-f945a062afcb>. [Accessed 17 10 2023].
- [2] Meta Platforms, "Jest JS," Meta Open Source, 2023. [Online]. Available: <https://jestjs.io/docs/getting-started>. [Accessed 17 10 2023].
- [3] OpenAI, "ChatGPT," OpenAI, 17 10 2023. [Online]. Available: <https://chat.openai.com/share/6ec0be78-9a9a-4be1-bcb4-11665f3691bf>. [Accessed 17 20 2023].