|  |
| --- |
| **2019년 10월 15일 실습보고서** |
| **학번 : 201810754, 이름 : 조윤직** |
| **실습자료1 : [복사 생성자]-** |
| **소스코드** |
| //Arraydata.h  #pragma once  class ArrayData  {  private:  double\* data;  int capacity;  int used;  static int count;  public:  static int getCount();  ArrayData(const int& capacity);  ArrayData(const ArrayData& arr);  ArrayData();  ~ArrayData();  void addElement(double num);  bool full() const;  int getCapacity() const;  int getUsed() const ;  void emptyArray();  void showData() const;  friend double getArraySum(const ArrayData& arr);  };  //Arraydata.cpp  #include "ArrayData.h"  #include<iostream>  using namespace std;  int ArrayData::getCount()  {  return count;  }  ArrayData::ArrayData(const int& capacity):capacity(capacity), used(0)  {  count++;  this->data = new double[capacity];  }  ArrayData::ArrayData(const ArrayData& arr) : capacity(arr.capacity),used(arr.used)  {  count++;  cout << "복사 생성자\n";  this->data = new double[this->capacity];  for (int i = 0; i < used; i++)  data[i] = arr.data[i];  }  ArrayData::ArrayData() :ArrayData(5)  {  }  int ArrayData::count = 0;  ArrayData::~ArrayData()  {  count--;  delete[] this->data;  data = nullptr;  }  void ArrayData::addElement(double num)  {  if (!full())  this->data[used++] = num;  else  cout << "빈 공간이 없음\n";    }  bool ArrayData::full() const  {  return capacity==used;  }  int ArrayData::getCapacity() const  {  return this->capacity;  }  int ArrayData::getUsed() const  {  return this->used;  }  void ArrayData::emptyArray()  {  this->used = 0;  }  void ArrayData::showData() const  {  double\* p = this->data;  double\* endp = p + this->used;  cout << "배열 출력 :";  while (p < endp) {  cout << \*(p++)<<" ";    }  cout << endl;  }  //main.cpp  #include <iostream>  #include <string>  #include "ArrayData.h"  using namespace std;  void showArray(ArrayData arr) { //여기서도 복사 생성자  arr.showData();  }  double getArraySum(const ArrayData& arr) {  double total = 0;  for (int i = 0; i < arr.getUsed(); i++) {  total += arr.data[i];  }  return total;  }  void lab01() {  ArrayData a1(3);  //ArrayData a2 = a1; //default 복사생성자 로 복사할경우 data가 같은 주소값으로 복사되어서 다른 복사 생성자 재정의 필요  a1.addElement(1.1);  a1.addElement(2.1);  /\*ArrayData a2(a1);  a2.addElement(3.1);  a1.addElement(5.1);\*/  showArray(a1);  }  void lab02() {  ArrayData a1(3);  a1.addElement(1.1);  a1.addElement(2.1);  ArrayData a2(a1);  ArrayData\* a3 = new ArrayData(2);  cout << ArrayData::getCount() << endl;  delete a3;  cout << ArrayData::getCount() << endl;  }  int main()  {  std::cout << "201810754 조윤직\n";  lab02();  } |
| **실행결과** |
|  |
| **실습자료2 : []** |
| **소스코드** |
|  |
| **실행결과** |
|  |
| **실습자료3 : []-** |
| **소스코드** |
|  |
| **실행결과** |
|  |