|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2021\_1\_C++\_04\_개념활용 | 학번 : | 20185309 | 이름 : | 황명원 |

* 개념 활용 프로그래밍

1. 다음 main() 함수가 잘 작동하도록 Member 클래스를 작성하고 프로그램을 완성하라.

* Member 클래스

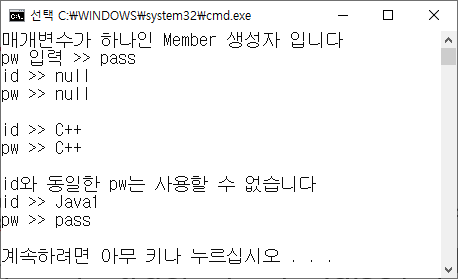
- id와 pw의 두 멤버 변수와 3 개의 생성자, 그리고 한 개의 소멸자로 구성

* 생성자 매개변수로 받은 값으로 멤버 변수를 초기화 한다

- 멤버함수

isCheck() : id와 pw를 비교하여 동일하면 true, 다르면 false반환

disPlay() : 객체 내용 출력, 반환값 없음

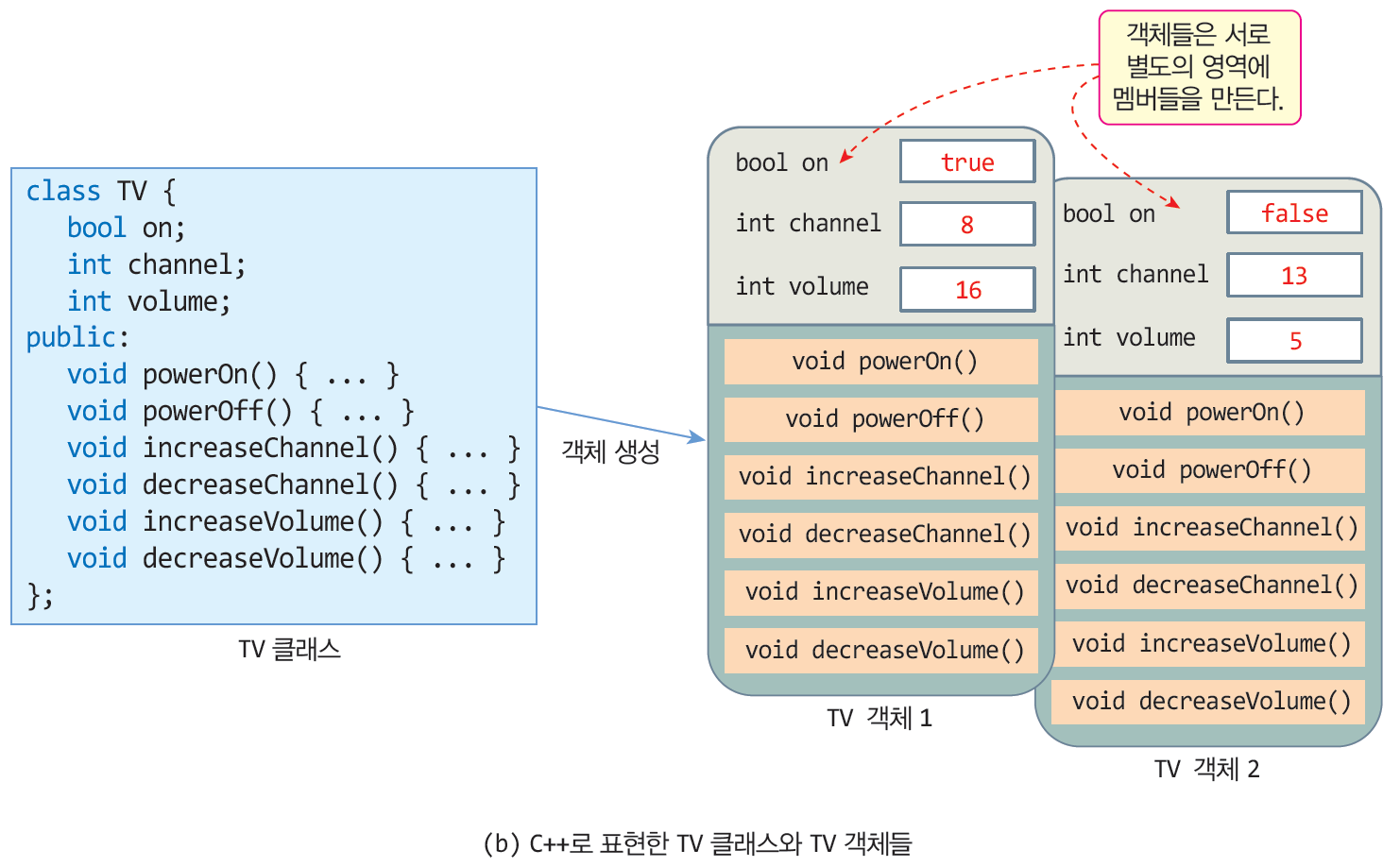


|  |
| --- |
| int main() {  Member mem;  Member mem1("C++", "C++");  Member mem2("Java1");  mem.disPlay();  mem1.disPlay();  if (mem1.isCheck())  cout << "id와 동일한 pw는 사용할 수 없습니다" << endl;  else  cout << "사용 가능한 pw 입니다" << endl;  mem2.disPlay();  return 0;  } |

|  |
| --- |
| [프로그램 소스]  #include <iostream> #include <string> using namespace std;  class Member{   string id;  string pw;  public:  Member();  Member(string s1,string s2);  Member(string s3);  ~Member();  void disPlay();  bool isCheck(); }; Member::Member() {  id="null";  pw="null"; } Member::Member(string s1,string s2) {  id=s1;  pw=s2; } Member::Member(string s3) {  id=s3;  string s4;  cout<<"매개변수가 하나인 Member 생성자 입니다."<<endl;  cout<<"pw 입력>>";  cin>>s4;  pw=s4; } Member::~Member() {  } void Member::disPlay() {  cout<<"id >> "<<id<<endl;  cout<<"pw >> "<<pw<<endl; } bool Member::isCheck() {  if(id==pw)  return true;  else  return false; }     int main() {  Member mem;  Member mem1("C++", "C++");  Member mem2("Java1");  mem.disPlay();  mem1.disPlay();  if (mem1.isCheck())  cout << "id와 동일한 pw는 사용할 수 없습니다" << endl;  else  cout << "사용 가능한 pw 입니다" << endl;  mem2.disPlay();  return 0; } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 다음 main() 함수가 잘 작동하도록 TV클래스를 작성하시오

* 단, 클래스 선언부와 구현부를 분리하여 작성
* 생성자와 소멸자 추가
* 모든 멤버에 대한 상수화 된 접근자 함수 추가
* 모든 멤버에 대한 설정자 함수 추가

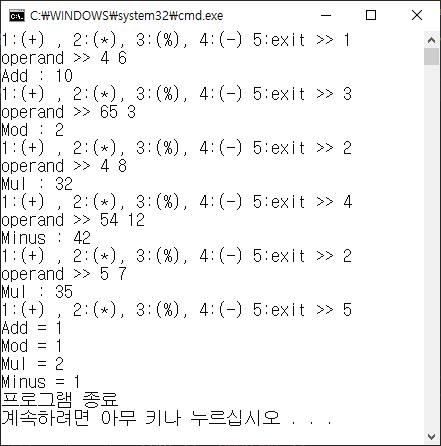


|  |
| --- |
| int main() {  TV tv;  tv.powerOn();  tv.state(); //on, volume, channel 값 출력  tv.increaseChannel();  cout << "채널 "<< CON::INC << " 증가" << endl;  tv.state();  return 0;  } |

|  |
| --- |
| [프로그램 소스]  ㅡ헤더파일  // // Created by 황명원 on 2021/03/29. //  #ifndef INC\_0329\_BASIC\_APLI\_TEST2\_H #define INC\_0329\_BASIC\_APLI\_TEST2\_H   #include <iostream> using namespace std;  class TV{  bool on;  int channel;  int volume; public:  TV();  ~TV();  void setOn(bool bl);  void setChannel(int ch);  void setVolume(int vol); //On,Channel,Volum 함수는 set 한  bool getOn() const;   int getChannel() const;  int getVolume() const; // On,Channel,Volume 함수를get하고 상수화   void powerOn();  void powerOff();  void increaseChannel();  void decreaseChannel();  void increaseVolume();  void decreaseVolume();   void state(); };  enum CON{*INC*=1,*DNC*}; //열거형 생성 #endif //INC\_0329\_BASIC\_APLI\_TEST2\_H  ㅡ소스파일  // // Created by 황명원 on 2021/03/29. // #include "test2.h"   TV::TV(){ //tv를 꺼놓고 채널,볼륨 0으로 생성  on=false;  channel=0;  volume=0; } TV::~TV() {  } void TV::setOn(bool bl) {  on=bl; } void TV::setChannel(int ch) {  channel=ch; } void TV::setVolume(int vol) { //각각 입력받은 변수로 On,Channel,Volume 설정  volume=vol; } bool TV::getOn() const {  return on; } int TV::getChannel() const{  return channel; } int TV::getVolume() const {//On,Channel,Volume를 각각 리턴받고 상수화  return volume; }  void TV::powerOn() {//티비를 켠다  on=true; } void TV::powerOff() {//티비를 끈다  on=false; } void TV::increaseChannel() { //채널 증가  channel++; } void TV::decreaseChannel() {//채널 감소  channel--; } void TV::increaseVolume() {//볼륨 증가  volume++; } void TV::decreaseVolume() {//볼륨 감소  volume--; } void TV::state() { //채널 상태 나타내기  cout<<"전원 : "<<on<<endl;  cout<<"채널 : "<<channel<<endl;  cout<<"볼륨 : "<<volume<<endl; }  ㅡ메인 소스  #include "test2.h"   int main() {  TV tv;  tv.powerOn();  tv.state(); //on, volume, channel 값 출력  tv.increaseChannel();   int n;  int num;  cout<<"증가(1),감소(2)>>";  cin>>num;  cout<<"증가 또는 감소할 수를 입력하세요>>";  cin>>n;  switch(num){  case CON::*INC*:  for(int i=0 ; i<n ; i++)  tv.increaseChannel();  break;   case CON::*DNC*:  for(int i=0 ; i<n ; i++)  tv.decreaseChannel();  break;  }  cout << "채널 " << CON::*INC* << " 증가" << endl;  tv.state();  return 0; } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

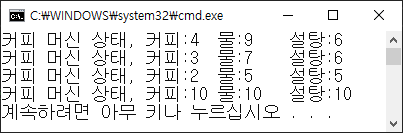
1. 제시된 결과처럼 메뉴에서 선택한 연산을 실행하는 프로그램을 작성하시오. 프로그램 종료 시 각 연산을 수행한 횟수를 출력하도록 한다. +, \*, %, - 연산을 하는 4개의 클래스를 각각 만들고, 각 클래스에는 다음과 같은 멤버를 갖는다. 제시된 조건 외에 필요에 따라 본인이 추가

* int 타입 변수 a, b;
* void setValue(int x, int y) 함수 : 매개변수를 멤버 변수에 복사
* int calculate() 함수 : 연산 결과 반환



|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  class Add{  int a,b;  int count1=0; public:  Add();  ~Add();  void setValue(int x,int y);  int getCount();  int calculate(); }; Add::Add(){} Add::~Add() {}  class Minus{  int a,b;  int count2=0; public:  Minus();  ~Minus();  void setValue(int x,int y);  int getCount();  int calculate(); }; Minus::Minus() {} Minus::~Minus() {} class Mul{  int a,b;  int count3=0; public:  Mul();  ~Mul();  void setValue(int x,int y);  int getCount();  int calculate(); }; Mul::Mul() {} Mul::~Mul() {}  class Mod{  int a,b;  int count4=0; public:  Mod();  ~Mod();  void setValue(int x,int y);  int getCount();  int calculate(); }; Mod::Mod() {} Mod::~Mod() {}  void Add::setValue(int x, int y) {  a=x;  b=y;  count1++; } void Minus::setValue(int x, int y) {  a=x;  b=y;  count2++; } void Mul::setValue(int x, int y) {  a=x;  b=y;  count3++; } void Mod::setValue(int x, int y) {  a=x;  b=y;  count4++; } int Add::calculate() { //a+b 계산  return a+b; } int Minus::calculate() { //a-b 계산  return a-b; } int Mul::calculate() {//a\*b 계산  return a\*b; } int Mod::calculate() {//a%b 계산  return a%b; } int Add::getCount(){ //add 함수 얼마나썼는지 count출력  return count1; } int Minus::getCount(){//minus함수 얼마나썼는지 count출력  return count2; } int Mul::getCount(){//mul함수 얼마나썼는지 count출력  return count3; } int Mod::getCount(){//mod함수 얼마나썼는지 count출력  return count4; }    int main(){  int num;  int num1,num2;  Add ad;  Minus mi;  Mul mu;  Mod mo;  cout<<"1:(+) , 2:(\*) , 3:(%) , 4:(-) 5:exit >>";  cin>>num; //num 입력   cout<<"operand >> ";  cin>>num1; //숫자 1입력   cin>>num2; //숫자 2입력    while(true) {   if (num == 1) {//num 이 1일때 더하기 함수 실행하기  ad.setValue(num1, num2);  cout << "ADD : " << ad.calculate() << endl;  }  else if (num == 2) { //num 이 2일때 곱하기 함수 실행하기  mu.setValue(num1, num2);  cout << "Mul : " << mu.calculate() << endl;  }  else if (num == 3) {//num 이 3일때 나누기 함수 실행하기  mo.setValue(num1, num2);  cout << "Mod : " << mo.calculate() << endl;  }  else if (num == 4) {//num 이 4일때 빼기 함수 실행하기  mi.setValue(num1, num2);  cout << "Minus : " << mi.calculate() << endl;  }   cout<<"1:(+) , 2:(\*) , 3:(%) , 4:(-) 5:exit >>";  cin>>num;  if(num==5)  break; //num이 5일경우 종료 조건만들기   cout<<"operand >> ";  cin>>num1;  cin>>num2;  };  cout<<"Add = "<<ad.getCount()<<endl;  cout<<"Mod = "<<mo.getCount()<<endl;  cout<<"Mul = "<<mu.getCount()<<endl;  cout<<"Minus = "<<mi.getCount()<<endl;  cout<<"프로그램 종료";   return 0; } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. main()함수와 실행 결과를 참고하여 Coffee 클래스를 작성하시오.



class CoffeeMachine {

int coffee, water, sugar; // 전체 커피, 물량, 설탕 량

public:

CoffeeMachine(int c, int w, int s);

void drinkEspresso(); // 커피 1, 물 1 소비

void drinkAmericano(); // 커피 1, 물 2 소비

void drinkSugarCoffee(); // 커피 1, 물 2, 설탕 1 소비

void fill(); // 커피 10, 물 10, 설탕 10으로 채우기

void show(); // 잔량 출력

};

int main() {

CoffeeMachine java(5, 10, 6); // 커피량:5, 물량:10, 설탕:6으로 초기화

java.drinkEspresso(); // 커피 1, 물 1 소비

java.show(); // 현재 커피 머신의 상태 출력

java.drinkAmericano(); // 커피 1, 물 2 소비

java.show(); // 현재 커피 머신의 상태 출력

java.drinkSugarCoffee(); // 커피 1, 물 2, 설탕 1 소비

java.show(); // 현재 커피 머신의 상태 출력

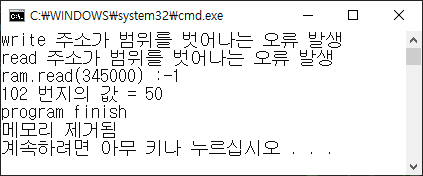
java.fill(); // 커피 10, 물 10, 설탕 10으로 채우기

java.show(); // 현재 커피 머신의 상태 출력

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;   class CoffeeMachine {  int coffee, water, sugar; // 전체 커피, 물량, 설탕 량 public:  CoffeeMachine(int c, int w, int s);  void drinkEspresso(); // 커피 1, 물 1 소비  void drinkAmericano(); // 커피 1, 물 2 소비  void drinkSugarCoffee(); // 커피 1, 물 2, 설탕 1 소비  void fill(); // 커피 10, 물 10, 설탕 10으로 채우기  void show(); // 잔량 출력 }; CoffeeMachine::CoffeeMachine(int c, int w, int s) {  coffee=c;  water=w;  sugar=s; } void CoffeeMachine::drinkEspresso() {  coffee--;  water--; } void CoffeeMachine::drinkAmericano() {  coffee--;  water=water-2; } void CoffeeMachine::drinkSugarCoffee() {  coffee--;  water=water-2;  sugar--; } void CoffeeMachine::fill() {  coffee=10;  water=10;  sugar=10; } void CoffeeMachine::show() {  cout<<"커피머신 상태, 커피:"<<coffee<<" 물:"<<water<<" 설탕:"<<sugar<<endl; } int main() {  CoffeeMachine java(5, 10, 6); // 커피량:5, 물량:10, 설탕:6으로 초기화  java.drinkEspresso(); // 커피 1, 물 1 소비  java.show(); // 현재 커피 머신의 상태 출력  java.drinkAmericano(); // 커피 1, 물 2 소비  java.show(); // 현재 커피 머신의 상태 출력  java.drinkSugarCoffee(); // 커피 1, 물 2, 설탕 1 소비  java.show(); // 현재 커피 머신의 상태 출력  java.fill(); // 커피 10, 물 10, 설탕 10으로 채우기  java.show(); // 현재 커피 머신의 상태 출력 } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |

1. 메인 메모리를 관리하는 Ram클래스를 구현하여 제시된 결과처럼 실행되도록 프로그램을 작성하시오.



class Ram {

char mem[100 \* 1024]; // 100KB 메모리

int size;

public:

Ram(); // mem을 0으로 초기화하고 size를 100\*1024로 초기화

~Ram(); // "메모리 제거됨" 문자열 출력

char read(int address); // address 주소의 메모리를 읽어 리턴

void write(int address, char value); // address 주소에 value 저장

};

int main() {

Ram ram;

ram.write(100, 20); // 100 번지에 20 저장

ram.write(101, 30); // 101 번지에 30 저장

ram.write(400000, 37); //400000 번지에 37 저장

cout<<"ram.read(345000) :" << (int)(ram.read(345000)) <<endl; //주소 범위를 벗어나면 -1 반환

char res = ram.read(100) + ram.read(101); // 20 + 30 = 50

ram.write(102, res); // 102 번지에 50 저장

cout << "102 번지의 값 = " << (int)ram.read(102) << endl; // 102 번지 메모리 값 출력

cout << "program finish" << endl;

}

|  |
| --- |
| [프로그램 소스]  #include <iostream> using namespace std;  class Ram {  char mem[100 \* 1024]; // 100KB 메모리  int size; public:  Ram(); // mem을 0으로 초기화하고 size를 100\*1024로 초기화  ~Ram(); // "메모리 제거됨" 문자열 출력  char read(int address); // address 주소의 메모리를 읽어 리턴  void write(int address, char value); // address 주소에 value 저장 }; Ram::Ram() {  size=100\*1024;  for(int i=0; i<size ; i++)  mem[i]=0;  } Ram::~Ram() {  cout<<"메모리 제거됨"<<endl; } char Ram::read(int address) {  if(address>100\*1024 or address<0){    return -1;  }  else  return mem[address]; } void Ram::write(int address, char value) {  if(address>100\*1024 or address<0)  cout<<"write 주소가 범위를 벗어나는 오류 발생"<<endl;  else  mem[address]=value;   } int main() {  Ram ram;  ram.write(100, 20); // 100 번지에 20 저장  ram.write(101, 30); // 101 번지에 30 저장  ram.write(400000, 37); //400000 번지에 37 저장   cout<<"ram.read(345000) :" << (int)(ram.read(345000)) <<endl; //주소 범위를 벗어나면 -1 반환  char res = ram.read(100) + ram.read(101); // 20 + 30 = 50  ram.write(102, res); // 102 번지에 50 저장  cout << "102 번지의 값 = " << (int)ram.read(102) << endl; // 102 번지 메모리 값 출력  cout << "program finish" << endl;  } |
| [실행결과]  텍스트이(가) 표시된 사진  자동 생성된 설명 |