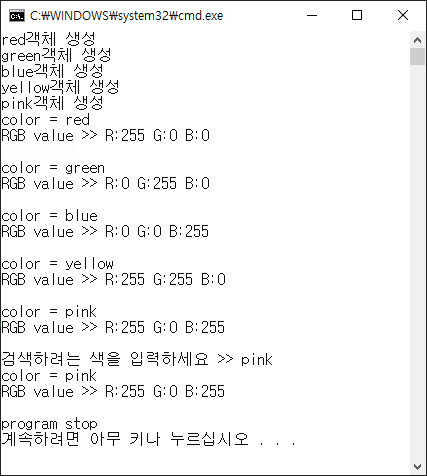
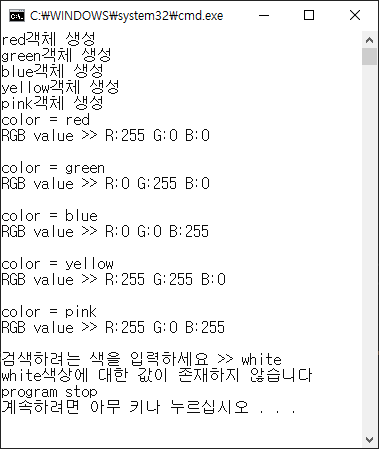
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| cpp\_05\_객체 배열 | 학번 : | 20175260 | 이름 : | 최영빈 |

* **개념 확인 – 객체 배열**

1. 제시된 Color 클래스를 구현한 후 제시된 조건대로 처리하는 프로그램을 작성하시오

* 크기가 5인 객체 배열을 생성하면서 초기화한다. (초기화 값은 본인이 임의로 할 것)
* 입력된 색상에 해당하는 RGB 코드값을 출력한다
* 멤버 변수의 값을 반환하는 getter 함수 추가할 것
* 전체 배열 원소를 출력한다

|  |
| --- |
| class Color {  int red, green, blue;  string color;  public:  Color():Color(0, 0, 0, "black") {}  Color(int r, int g, int b, string c) : red(r), green(g), blue(b), color(c){ }  void setColor(int r, int g, int b, string c); //색상 변경  void show() const; //멤버 변수 출력  //색상을 반환하는 접근자 함수 추가  };  void search(Circle \*p, string color, int size); //매개변수로 전달된 색상에 해당하는 RGB 값 출력, 전역 함수 |

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  using namespace std;  class Color{  int red,green,blue;  string color;  public :  Color():Color(0,0,0,"black"){}  Color(int r,int g,int b,string c) : red(r), green(g),blue(b),color(c){  cout << color <<"객체 생성"<<endl;  }  void setColor(int r,int g,int b,string c);  string getColor();  int getRed();  int getGreen();  int getBlue();  void show() const;  };  void Color::setColor(int r,int g,int b,string c)  {  red = r;  green = g;  blue = b;  color = c;  }  string Color::getColor(){  return color;  }  int Color::getRed(){  return red;  }  int Color::getGreen()  {  return green;  }  int Color::getBlue()  {  return blue;  }  void Color::show() const  {    cout << "color = " << color <<endl;  cout << "RGB balue >> R:" << red <<" G:" <<green <<" B:" <<blue<<endl<<endl;  }  void search(Color \*p, string color, int size);  void search(Color \*p, string color, int size){  for(int i= 0; i < 5; i++){  if(p[i].getColor() == color){  p[i].show();  break;  }  else if(i == 4)  cout << color << "색상에 대한 값이 존재하지 않습니다"<<endl;  }  }  int main()  {  string rColor;  Color \*C = new Color[5]{Color(255, 0, 0, "red"),Color(0, 255, 0, "green"), Color(0, 0, 255, "blue"), Color(127, 127, 127, "gray"), Color(255, 255, 0, "yellow")};  for(auto i = 0; i < 5; i++){  C[i].show();  }  cout << "검색하려는 색을 입력하세요 >>";  cin >> rColor;  search(C,rColor,5);    cout << "program stop"<<endl;  delete []C;  return 0;  } |

* **개념 적용 응용 프로그래밍 – 클래스와 객체**

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

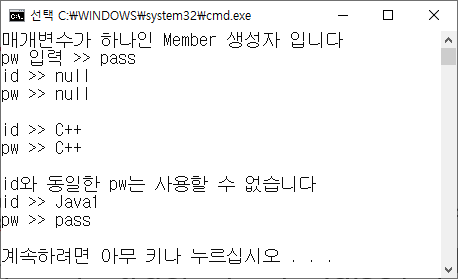
* Member 클래스

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

- 멤버함수(상수화 함수로 구현)

isCheck() : id와 pw를 비교하여 같으면 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 << "사용 가능한 pw 입니다" << endl;  else  cout << "id와 동일한 pw는 사용할 수 없습니다" << endl;  mem2.disPlay();  return 0;  } |

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  using namespace std;  class Member{  string id, pw;  public :  Member():id("null"),pw("null"){};  Member(string i):id(i),pw("null"){  cout << "매개변수가 하나인 Member 생성자 입니다"<<endl;  cout << "pw 입력 >> ";  cin >> pw;  };  Member(string i,string p):id(i),pw(p){};  ~Member(){};  bool isCheck() const;  void display() const;  };  bool Member::isCheck() const{  if(id == pw)  return true;  else  return false;  }  void Member::display() const{  cout << "id >> " <<id<<endl;  cout << "pw >> " <<pw<<endl;  }  int main()  {  Member mem;  Member mem1("C++", "C++");  Member mem2("Java1");  mem.display();  mem1.display();  if (mem1.isCheck())  cout << "사용 가능한 pw 입니다" << endl;  else  cout << "id와 동일한 pw는 사용할 수 없습니다" << endl;  mem2.display();  return 0;  } |
| [실행결과]  이미지 |

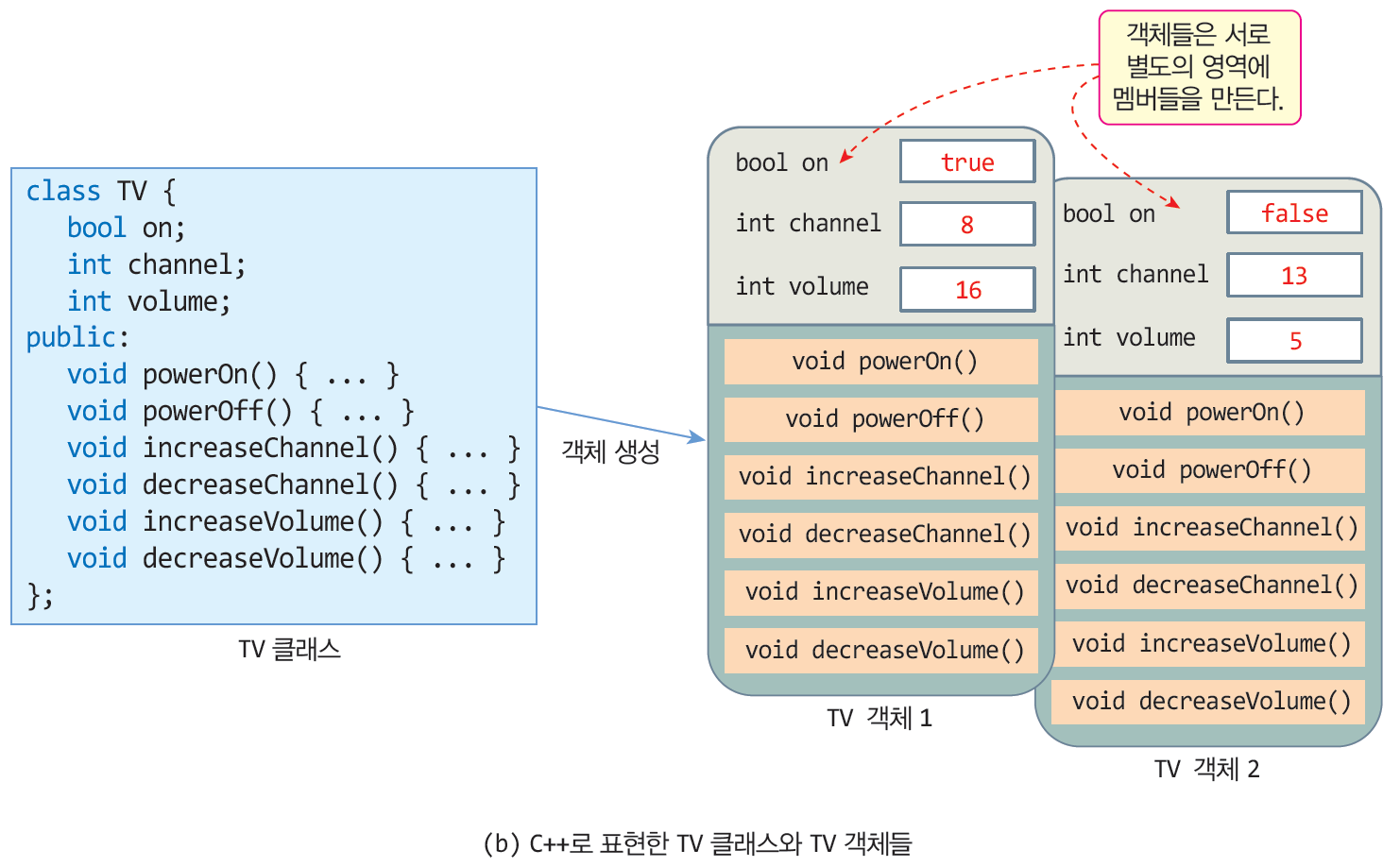
1. 다음 main() 함수가 잘 작동하도록 TV클래스를 작성하시오. 열거형 클래스도 추가하여 볼륨과 채널 증가 감소에 적용하시오.

단, 클래스 선언부와 구현부를 분리하여 작성

생성자와 소멸자 추가

모든 멤버에 대한 상수화 된 getter 함수 추가 – 상수화 함수로 할 것

모든 멤버에 대한 setter 함수 추가

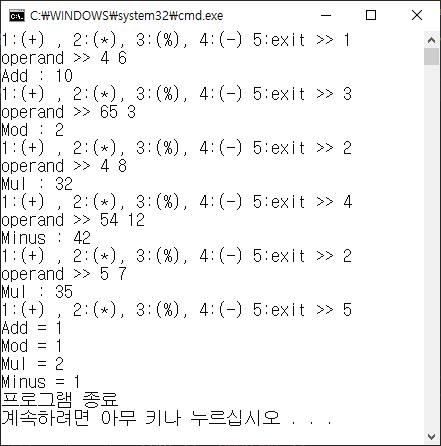


|  |
| --- |
| int main() {  TV tv;  tv.powerOn();  tv.state(); //on, volume, channel 값 출력  tv.increaseChannel();  cout << "채널 "<< CON::INC << " 증가" << endl;  tv.state();  return 0;  } |

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  #include <iomanip>  using namespace std;  class TV{  bool on;  int channel;  int volume;  public:  TV():on(true),channel(0),volume(0){};  ~TV(){};  void powerOn();  void powerOff();  void increaseChannel();  void decreaseChannel();  void increaseVolume();  void decreaseVolume();  void state();  bool getOn() const;  int getChannel() const;  int getVolume() const;  void setOn(bool on);  void setChannel(int channel);  void setVolume(int volume);  };  void TV::powerOn(){  on = true;  }  void TV::powerOff(){  on = false;  }  void TV::increaseChannel(){  channel++;  }  void TV::decreaseChannel(){  channel--;  }  void TV::increaseVolume(){  volume++;  }  void TV::decreaseVolume(){  volume--;  }  void TV::state(){  cout<<"전원 :" <<boolalpha<< on <<" 볼륨 :" << volume << " 채널 :" <<channel <<endl;  }  bool TV::getOn() const  {  return on;  }  int TV::getChannel() const  {  return channel;  }  int TV::getVolume() const  {  return volume;  }  void TV::setOn(bool on)  {  on = on;  }  void TV::setChannel(int channel)  {  channel = channel;  }  void TV::setVolume(int volume)  {  volume = volume;  }  int main()  {  enum CON {INC =1,DEC = -1,INV = 1,DEV = -1};  TV tv;  tv.powerOn();  tv.state(); //on, volume, channel 값 출력  tv.increaseChannel();  cout << "채널 " << CON::INC << " 증가" << endl;  tv.state();  return 0;  } |
| [실행결과]  이미지 |

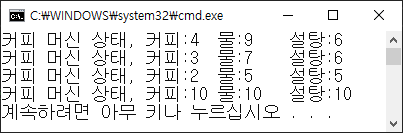
1. 제시된 결과처럼 메뉴에서 선택한 연산을 실행하는 프로그램을 작성하시오. 프로그램 종료 시 각 연산을 수행한 횟수를 출력하도록 한다. +, \*, %, - 연산을 하는 4개의 클래스를 각각 만들고, 각 클래스에는 다음과 같은 멤버를 갖는다. 제시된 조건 외에 필요에 따라 본인이 추가

* 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>  #include <string>  using namespace std;  class Add{  int a, b;  public :  void setValue(int x,int y){a = x, b = y;}  int calculate(){return a + b;}  };  class Minus{  int a, b;  public:  void setValue(int x, int y) { a = x, b = y; }  int calculate() { return a - b; }  };  class Mul{  int a, b;  public:  void setValue(int x, int y) { a = x, b = y; }  int calculate() { return a \* b; }  };  class Mod{  int a, b;  public:  void setValue(int x, int y) { a = x, b = y; }  int calculate() { return a % b; }  };  int main()  {  int i,j,k, ad=0,mu=0,mo=0,mi=0;  for(;;){  cout << "1 :(+), 2: (\*), 3: (%), 4: (-),5: exit >>";  cin >> i;  switch (i)  {  case 1:  cout << "operand >>";  cin >> j >> k;  Add add;  add.setValue(j, k);  cout << "Add :" << add.calculate() << endl;  ad++;  break;  case 2:  cout << "operand >>";  cin >> j >> k;  Mul mul;  mul.setValue(j, k);  cout << "Mul :" << mul.calculate() << endl;  mu++;  break;  case 3:  Mod mod;  cout << "operand >>";  cin >> j >> k;  mod.setValue(j,k);  cout << "Mod :"<<mod.calculate()<<endl;  mo++;  break;  case 4:  cout << "operand >>";  cin >> j >> k;  Minus minus;  minus.setValue(j, k);  cout << "Minus :" << minus.calculate() << endl;  mi++;  break;  case 5:  cout << "Add = "<<ad<<endl;  cout << "Mod = "<<mo<<endl;  cout << "Mul = "<<mu<<endl;  cout << "Minus = "<<mi<<endl;  cout <<"프로그램 종료"<<endl;  return 0;      }  }  } |
| [실행결과]  이미지 |

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

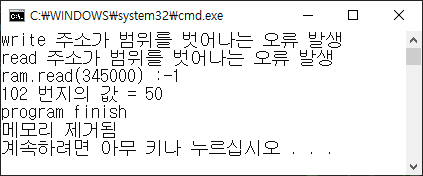
java.fill(); // 커피 10, 물 10, 설탕 10으로 채우기

java.show(); // 현재 커피 머신의 상태 출력

}

|  |
| --- |
| [프로그램 소스]  #include <iostream>  #include <string>  using namespace std;  class CoffeeMachine  {  int coffee, water, sugar; // 전체 커피, 물량, 설탕 량  public:  CoffeeMachine(int c, int w, int s):coffee(c),water(w),sugar(s){};  void drinkEspresso(){coffee--,water--;} // 커피 1, 물 1 소비  void drinkAmericano(){coffee--,water-=2;} // 커피 1, 물 2 소비  void drinkSugarCoffee(){coffee--,water-=2,sugar--;} // 커피 1, 물 2, 설탕 1 소비  void fill(){coffee=10,water=10,sugar=10;} // 커피 10, 물 10, 설탕 10으로 채우기  void show(){cout <<"커피 머신 상태, 커피:"<<coffee<<" 물:"<<water<<" 설탕:"<<sugar<<endl;} // 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  return 0;  } |
| [실행결과]  이미지 |

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;

}

|  |
| --- |
| #include <iostream>  #include <string>  using namespace std;  class Ram  {  char mem[100 \* 1024]; // 100KB 메모리  int size;  public:  Ram(){\*mem = {0},size = \*mem;} // mem을 0으로 초기화하고 size를 100\*1024로 초기화  ~Ram(){cout << "메모리 제거됨"<<endl;} // "메모리 제거됨" 문자열 출력  char read(int address){  if(address < 100 \* 1024)  return mem[address];  else  return -1;  } // address 주소의 메모리를 읽어 리턴  void write(int address, char value){  if(address < 100\*1024)  mem[address] =value;  else  cout << "write 주소가 범위를 벗어나는 오류 발생" << endl;  } // 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  if ((int)(ram.read(345000)) == -1)  cout << "read 주소가 범위를 벗어나는 오류 발생" << endl;  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;  return 0;  } |
| [실행결과]  이미지 |