|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2020\_2\_C++ \_03 | 학번 : | 20165133 | 이름 : | 방승원 |

* **강의 내용**

1. 클래스와 객체 활용 - TV 클래스 만들기

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  namespace CON {  enum { INC = 2, DEC = 2, VMAX = 100, CMAX = 999};  };  class TV {  bool on;  int channel;  int volume;  public:  void powerOn();  void powerOff() {  on = false;  }  void increaseChannel();  void decreaseChannel();  void increaseVolume();  void decreaseVolume();  void state();  };  void TV::powerOn() {  on = true;  channel = 1;  volume = 1;  }  void TV::increaseChannel() {  channel += CON::INC;  if (channel > CON::CMAX)  channel = CON::CMAX;  }  void TV::decreaseChannel() {  channel -= CON::DEC;  if (channel < 0)  channel = 0;  }  void TV::increaseVolume() {  volume += CON::INC;  if (volume > CON::VMAX)  volume = CON::VMAX;  }  void TV::decreaseVolume() {  volume -= CON::DEC;  if (volume < 0)  volume = 0;  }  void TV::state() {  cout << "power >> " << boolalpha << on << endl;  cout << "volume >> " << volume << endl;  cout << "channel >> " << channel << endl << endl;  }  int main() {  TV tv;  tv.powerOn();  tv.state();  tv.increaseChannel();  cout << "채널 " << CON::DEC << " 증가" << endl;  tv.state();  return 0;  } |
| **[실행 결과]** |

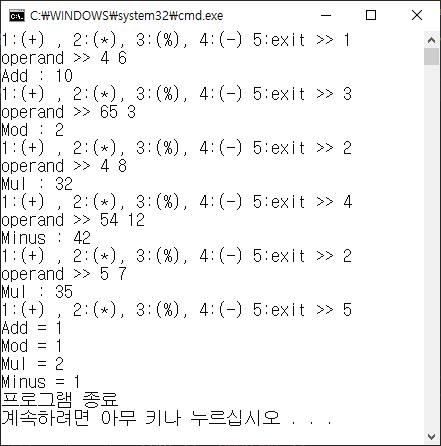
1. 생성자 활용 – Member 클래스

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Member {  private:  string id;  string pw;  public:  Member(); //생성자에 대한 정의가 하나라도 있다면 디폴트 생성자를 생성  Member(string id, string pw);  Member(string id);  bool isCheck(); // id, pw 비교  void disPlay(); //출력  };  Member::Member() : Member("null", "null"){ } //위임 생성자(타겟 생성자를 호출하는 생성자)  Member::Member(string i, string p) : id(i), pw(p) { } //타겟 생성자  Member::Member(string i) : id(i) {  } cout << "매개변수가 하나인 Member생성자 입니다. " << endl;  cout << "pw 입력 >> ";  cin >> pw;  bool Member::isCheck() {  if (id == pw)  return false;  else  return true;  //아이디랑 비밀번호가 같으면 사용 불가  }  void Member::disPlay() {  cout << "id >> " + id << endl;  cout << "pw >> " + pw << endl << endl;  }  int main() {  Member mem; //디폴트 생성자 호출해서 객체 생성  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 << endl;  }  mem2.disPlay();  return 0;  } |
| **[실행 결과]** |

* **프로그램 과제**

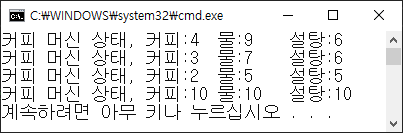
1. 제시된 결과처럼 메뉴에서 선택한 연산을 실행하는 프로그램을 작성하시오. 프로그램 종료 시 각 연산을 수행한 횟수를 출력하도록 한다. +, \*, %, - 연산을 하는 4개의 클래스를 각각 만들고, 각 클래스에는 다음과 같은 멤버를 갖는다. 제시된 조건 외에 필요에 따라 본인이 추가

* 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 {  public:  int a, b;  void setValue(int x, int y);  int calculate();  };  class Minus {  public:  int a, b;  void setValue(int x, int y);  int calculate();  };  class Mod {  public:  int a , b ;  void setValue(int x, int y);  int calculate();  };  class Mul {  public:  int a , b ;  void setValue(int x, int y);  int calculate();  };  void Add :: setValue(int x, int y) {  a = x;  b = y;  }  int Add::calculate() {  return a + b;  }  void Minus::setValue(int x, int y) {  a = x;  b = y;  }  int Minus::calculate() {  return a - b;  }  void Mod::setValue(int x, int y) {  a = x;  b = y;  }  int Mod::calculate() {  return a % b;  }  void Mul::setValue(int x, int y) {  a = x;  b = y;  }  int Mul::calculate() {  return a \* b;  }  int main() {  Add add;  Minus min;  Mod mod;  Mul mul;  int num=0;  int q, w;  int cnt1 = 0, cnt2 = 0, cnt3 = 0, cnt4 = 0;  while (1) {  cout << "1: (+), 2:(\*), 3:(%), 4:(-), 5:exit >> ";  cin >> num;  if (num == 1) {  cout << "operand >> ";  cin >> q >> w;  add.setValue(q, w);  cout << "Add : " << add.calculate() << endl;  ++cnt1;  }  else if (num == 2) {  cout << "operand >> ";  cin >> q >> w;  mul.setValue(q, w);  cout << "Mul : " << mul.calculate() << endl;  ++cnt2;  }  else if (num == 3) {  cout << "operand >> ";  cin >> q >> w;  mod.setValue(q, w);  cout << "Mod : " << mod.calculate() << endl;  ++cnt3;  }  else if (num == 4) {  cout << "operand >> ";  cin >> q >> w;  min.setValue(q, w);  cout << "Minus : " << min.calculate() << endl;  ++cnt4;  }  else if (num == 5) {  cout << "Add = " << cnt1 << "번 사용하셨습니다. " << endl;  cout << "Mul = " << cnt2 << "번 사용하셨습니다. " << endl;  cout << "Mod = " << cnt3 << "번 사용하셨습니다. " << endl;  cout << "Minus = " << cnt4 << "번 사용하셨습니다. " << endl;  cout << "종료되었습니다." << endl;  break;  }  else {  cout << "잘못입력하였습니다. 다시 입력하세요. " << endl;  continue;  }  }  return 0;  } |
| **[실행 결과]** |

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

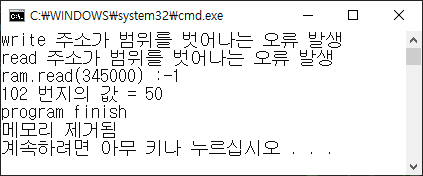
java.fill(); // 커피 10, 물 10, 설탕 10으로 채우기

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);  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() {  if(coffee < 1 || water < 1) return;  coffee -= 1;  water -= 1;  }  void CoffeeMachine::drinkAmericano() {  if(coffee < 1 || water < 2) return;  coffee -= 1;  water -= 2;  }  void CoffeeMachine::drinkSugarCoffee() {  if(coffee < 1 || water < 1 || sugar < 1) return;  coffee -= 1;  water -= 2;  sugar -= 1;  }  void CoffeeMachine::fill() {  coffee = 10;  water = 10;  sugar = 10;  }  void CoffeeMachine::show() {  cout << "커피 머신 상태, 커피:" << coffee << "\t물:" << water << "\t설탕:" << 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  return 0;  } |
| **[실행 결과]** |

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>  #include <string>  using namespace std;  class Ram {  char mem[100 \* 1024]; // 100kb 메모리. 한 번지는 한 바이트이므로 char 타입 사용.  int size;  public:  Ram(); // mem 배열을 0으로 초기화 하고 size를 100\*1024로 초기화  ~Ram(); // "메모리 제거됨" 문자열 출력  char read(int address); // address 주소의 메모리 바이트 리턴  void write(int address, char value); // address 주소에 한 바이트로 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 < 0 || address > size) {  cout << "read 주소가 범위를 벗어나는 오류 발생" << endl;  return -1;  }  else {  return mem[address];  }  }  void Ram::write(int address, char value) {  if (address > size)  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;  return 0;  } |
| **[실행 결과]** |