|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2020\_2\_C++ \_03 | 학번 : | 20175105 | 이름 : | 곽영주 |

* **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::INC << " 증가" << endl;  tv.state();  return 0;  } |
| **[실행 결과]** |

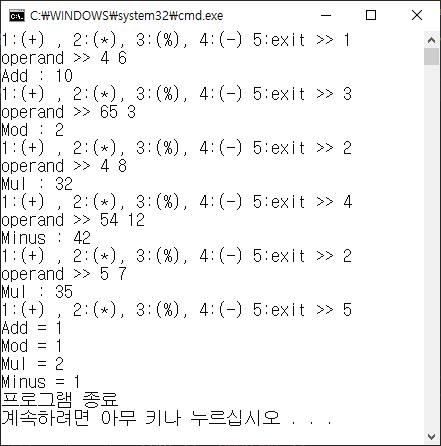
1. 생성자 활용 – Member 클래스

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  using namespace std;  class Member {  private :  string id;  string pw;  public :  Member();  Member(string id, string pw);  Member(string id);  bool isCheck();  void disPlay();  };  Member::Member() : Member("null", "null") {}  Member::Member(string i, string p) : id(i), pw(p) {}  Member::Member(string i) : id(i) {  cout << "매개변수가 하나인 Member 생성자 입니다" << endl;  cout << "pw 입력 >> ";  cin >> pw;  }  bool Member::isCheck() {  if (id == pw)  return false;  else  return true;  }  void Member::disPlay() {  cout << "id >> " + id << endl;  cout << "pw >> " + pw << endl << 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;  } |
| **[실행 결과]** |

* **프로그램 과제**

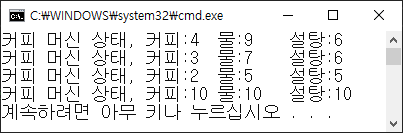
1. 제시된 결과처럼 메뉴에서 선택한 연산을 실행하는 프로그램을 작성하시오. 프로그램 종료 시 각 연산을 수행한 횟수를 출력하도록 한다. +, \*, %, - 연산을 하는 4개의 클래스를 각각 만들고, 각 클래스에는 다음과 같은 멤버를 갖는다. 제시된 조건 외에 필요에 따라 본인이 추가

* 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>  using namespace std;  class Add {  int a, b;  public :  void setValue(int x, int y); // 매개변수를 멤버 변수에 복사  int calculate(); // 연산 결과 반환  };  void Add::setValue(int x, int y) {  a = x;  b = y;  }  int Add::calculate() {  return a + b;  }  class Mod {  int a, b;  public:  void setValue(int x, int y);  int calculate();  };  void Mod::setValue(int x, int y) {  a = x;  b = y;  }  int Mod::calculate() {  return a % b;  }  class Mul {  int a, b;  public:  void setValue(int x, int y);  int calculate();  };  void Mul::setValue(int x, int y) {  a = x;  b = y;  }  int Mul ::calculate() {  return a \* b;  }  class Minus {  int a, b;  public:  void setValue(int x, int y);  int calculate();  };  void Minus::setValue(int x, int y) {  a = x;  b = y;  }  int Minus::calculate() {  return a - b;  }  int main() {  Add add;  Mod mod;  Mul mul;  Minus minus;  int choice, num1, num2;  int addCount = 0, modCount = 0, mulCount = 0, minusCount = 0; // 연산 수행 횟수  while (1) {  cout << "1:(+), 2:(\*), 3:(%), 4:(-), 5:exit >> ";  cin >> choice;  if (choice != 5) {  cout << "operand >> ";  cin >> num1 >> num2;  }  switch (choice) {  case 1: // Add  add.setValue(num1, num2);  cout << "Add : " << add.calculate() << endl;  addCount++;  break;  case 2: // Mul  mul.setValue(num1, num2);  cout << "Mul : " << mul.calculate() << endl;  mulCount++;  break;  case 3: // Mod  mod.setValue(num1, num2);  cout << "Mod : " << mod.calculate() << endl;  modCount++;  break;  case 4: // Minus  minus.setValue(num1, num2);  cout << "Minus : " << minus.calculate() << endl;  minusCount++;  break;  case 5: // Exit  cout << "Add = " << addCount << endl;  cout << "Mod = " << modCount << endl;  cout << "Mul = " << mulCount << endl;  cout << "Minus = " << minusCount << endl;  cout << "프로그램 종료" << endl << endl;  return 0;  }  }  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

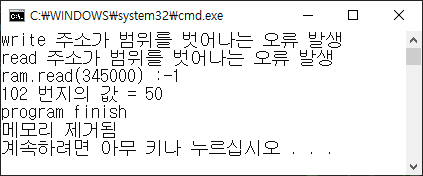
java.fill(); // 커피 10, 물 10, 설탕 10으로 채우기

java.show(); // 현재 커피 머신의 상태 출력

}

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  using namespace std;  class CoffeeMachine {  int coffee, water, sugar;  public:  CoffeeMachine(int c, int w, int s);  void drinkEspresso();  void drinkAmericano();  void drinkSugarCoffee();  void fill();  void show();  };  CoffeeMachine::CoffeeMachine(int c, int w, int s) : coffee(c), water(w), sugar(s) {}  void CoffeeMachine::drinkEspresso() {  coffee -= 1;  water -= 1;  }  void CoffeeMachine::drinkAmericano() {  coffee -= 1;  water -= 2;  }  void CoffeeMachine::drinkSugarCoffee() {  coffee -= 1;  water -= 2;  sugar -= 1;  }  void CoffeeMachine::fill() {  coffee = 10;  water = 10;  sugar = 10;  }  void CoffeeMachine::show() {  cout << "커피 머신 상태, " << "커피:" << coffee <<" 물:" << water << " 설탕:" << sugar << endl;  }  int main() {  CoffeeMachine java(5, 10, 6);  java.drinkEspresso();  java.show();  java.drinkAmericano();  java.show();  java.drinkSugarCoffee();  java.show();  java.fill();  java.show();    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>  using namespace std;  class Ram {  char mem[100 \* 1024];  int size;  public:  Ram();  ~Ram();  char read(int address);  void write(int address, char value);  };  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 < size)  return mem[address];  else {  cout << "read 주소가 범위를 벗어나는 오류 발생" << endl;  return -1;  }  }  void Ram::write(int address, char value) {  if (address < size)  mem[address] = value;  else  cout << "write 주소가 범위를 벗어나는 오류 발생" << endl;  }  int main() {  Ram ram;  ram.write(100, 20);  ram.write(101, 30);  ram.write(400000, 37);  cout << "ram.read(345000) :" << (int)(ram.read(345000)) << endl;  char res = ram.read(100) + ram.read(101);  ram.write(102, res);  cout << "102 번지의 값 = " << (int)(ram.read(102)) << endl;  cout << "program finish" << endl;  return 0;  } |
| **[실행 결과]** |