|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2018\_2\_C++ \_13 | 학번 : |  | 이름 : |  |

* STL(Standard Template Library 활용 - map 컨테이너

|  |
| --- |
| #include <iostream>  #include <string>  #include <map>  using namespace std;  int main() {  map<string, string> dic; // 맵 컨테이너 생성. 키는 영어 단어, 값은 한글 단어  // 단어 3개를 map에 저장  dic.insert(make\_pair("love", "사랑")); // ("love", "사랑") 저장  dic.insert(make\_pair("apple", "사과")); // ("apple", "사과") 저장  dic["cherry"] = "체리"; // ("cherry", "체리") 저장  cout << "저장된 단어 개수 " << dic.size() << endl;  string eng;  while (true) {  cout << "찾고 싶은 단어>> ";  getline(cin, eng); // 사용자로부터 키 입력  if (eng == "exit")  break; // "exit"이 입력되면 종료  if(dic.find(eng) == dic.end()) // eng '키'를 끝까지 찾았는데 없음  cout << "없음" << endl;  else  cout << dic[eng] << endl; // dic에서 eng의 값을 찾아 출력  }  cout << "종료합니다..." << endl;  } |
| **[실행 결과]** |

* **auto를 이용한 변수 선언**

|  |
| --- |
| #include <iostream>  #include <vector>  using namespace std;  int square(int x) { return x\*x; }  int main() {  // 기본 타입 선언에 auto 활용  auto c = 'a'; // c는 char 타입으로 결정  auto pi = 3.14; // pi은 double 타입으로 결정  auto ten = 10; // ten은 int 타입으로 결정  auto \*p = &ten; // 변수 p는 int\* 타입으로 결정  cout << c << " " << pi << " " << ten << " " << \*p << endl;  // 함수의 리턴 타입으로 추론  auto ret = square(3); // square() 함수의 리턴 타입이 int 이므로 ret는 int로 결정  cout << \*p << " " << ret << endl;  vector<int> v = { 1,2,3,4, 5 }; //벡터 v에 5개의 원소, 1,2,3,4,5 삽입  vector<int>::iterator it;  for (it = v.begin(); it != v.end(); it++)  cout << \*it << " "; // 1 2 3 4 5 출력  cout << endl;  // 템플릿에 auto를 사용하여 복잡한 선언의 간소화  for (auto it = v.begin(); it != v.end(); it++)  cout << \*it << " "; // 1 2 3 4 5 출력  } |
| **[실행 결과]** |

* 람다식 활용 - 캡쳐리스트에 참조 활용

|  |
| --- |
| #include <iostream>  using namespace std;  int main() {  int sum = 0; // 지역 변수  [&sum](int x, int y) { sum = x + y; } (2, 3); // 합 5를 지역변수 sum에 저장  cout << "합은 " << sum;  } |
| **[실행 결과]** |

* STL 템플릿에 람다식 활용

|  |
| --- |
| #include <iostream>  #include <vector>  #include <algorithm> // for\_each() 알고리즘 함수를 사용하기 위함  using namespace std;  void print(int n) {  cout << n << " ";  }  int main() {  vector<int> v = { 1, 2, 3, 4, 5 };  // for\_each()는 벡터 v의 첫번째 원소부터 끝까지 검색하면서,  // 각 원소에 대해 print(int n) 호출. 매개 변수 n에 각 원소 값 전달  for\_each(v.begin(), v.end(), print);  } |
| **[실행 결과]** |

* 프로그램 과제

1. 교재 528p : 문제 10번

|  |
| --- |
| **[프로그램 소스]**  #include <iostream>  #include <string>  #include <vector>  #include <ctime>  #include <cstdlib>  using namespace std;  class Nation {  string nation;  string capital;  public:  Nation(string nation, string capital) {  this->nation = nation;  this->capital = capital;  }  string getCapital() { return capital; }  string getNation() { return nation; }  void show() {  cout << '(' << nation << ',' << capital << ')';  }  };  class NationGame {  vector<Nation> v;  void input();  void list();  void quiz();  bool exist(string nation);  public:  NationGame();  void run();  };  NationGame::NationGame() {  Nation n[] = { Nation("미국", "와싱턴"), Nation("영국", "런던"), Nation("프랑스", "파리"),  Nation("중국", "베이찡"), Nation("일본", "도쿄"), Nation("러시아", "모스크바"),  Nation("브라질", "브라질리아"), Nation("독일", "베를린"), Nation("멕시코", "멕시코시티")};  for(int i=0; i<9; i++)  v.push\_back(n[i]);  srand((unsigned)time(0)); // 시작할 때마다, 다른 랜덤수를 발생시키기 위한 seed 설정  }  void NationGame::run() {  cout << "\*\*\*\*\* 나라의 수도 맞추기 게임을 시작합니다. \*\*\*\*\*" << endl;  while(true) {  int cmd;  cout << "정보 입력: 1, 퀴즈: 2, 종료: 3 >> ";  cin >> cmd;  switch(cmd) {  case 1 : input(); break;  case 2 : quiz(); break;  case 3 : return;  }  }  }  void NationGame::quiz() {  while(true) {  int index = rand() % v.size(); // 0에서 RAND\_MAX(32767) 사이의 랜덤한 정수가 n에 발생  cout << v[index].getNation() << "의 수도는?";  string capital;  cin >> capital;  if(capital == "exit")  return; // quiz 그만  if(v[index].getCapital() == capital)  cout << "Correct !!" << endl;  else  cout << "NO !!" << endl;  }  }  void NationGame::input() {  string nation, capital;  cout << "현재 " << v.size() << "개의 나라가 입력되어 있습니다." << endl;  cout << "나라와 수도를 입력하세요(no no 이면 입력끝)" << endl;  while(true) {  cout << v.size()+1 << ">>";  cin >> nation >> capital;  if(nation == "no" && capital=="no")  break;  if(exist(nation)) {  cout << "already exists !!" << endl;  continue;  }  Nation n(nation, capital);  v.push\_back(n);  }  // list();  return;  }  bool NationGame::exist(string nation) {  for(int i=0; i<v.size(); i++) {  if(v[i].getNation() == nation)  return true;  }  return false;  }  void NationGame::list() {  for(int i=0; i<v.size(); i++) {  v[i].show();  cout << endl;  }  }  int main() {  NationGame game;  game.run();  } |
| **[실행 결과]** |

1. 교재 530p : 문제 14번

|  |
| --- |
| **[프로그램 소스]**  ============== Password.h ==============  #ifndef PASSWORD\_H  #define PASSWORD\_H  #include <string>  #include <map>  using namespace std;  class PasswordManager {  string program;  map<string, string> passwordMap;  void add();  void search();  bool checkInputError();  public:  PasswordManager(string);  void run();  };  #endif  ========== password.cpp ===============  #include <iostream>  #include <string>  #include <map>  using namespace std;  #include "Password.h"  PasswordManager::PasswordManager(string program) {  this->program = program;  }  void PasswordManager::add() {  string name;  string password;  cout << "이름 암호>> ";  cin >> name >> password;  passwordMap[name] = password;  }  void PasswordManager::search() {  string name;  string inputPassword, storedPassword;  cout << "이름? ";  cin >> name;  while (true) {  cout << "암호? ";  cin >> inputPassword;  storedPassword = passwordMap[name];  if (inputPassword == storedPassword) {  cout << "통과!!" << endl;  break;  }  else {  cout << "실패~~" << endl;  }  }  }  bool PasswordManager::checkInputError() {  if (cin.fail()) {  cin.clear();  cin.ignore(100, '\n');  cout << "입력 오류" << endl;  return true; // 오류 있음  }  else  return false; // 오류 없음  }  void PasswordManager::run() {  int menu;  cout << "\*\*\*\*\* 암호관리 프로그램 " << program << "를 시작합니다 \*\*\*\*\*" << endl;  while (true) {  cout << "삽입:1, 검사:2, 종료:3>> ";  cin >> menu;  // 단순하게 작성하려는 경우 다음 2 라인은 호출하지 않아도 됨  if (checkInputError()) // 메뉴(menu)를 사용자가 숫자를 입력하지 않는 경우, 오류 발생  continue;  switch (menu) {  case 1: add(); break;  case 2: search(); break;  case 3: cout << "프로그램을 종료합니다..."; return;  }  }  }  ========== main.cpp ===========  #include "Password.h"  int main() {  PasswordManager mm("WHO");  mm.run();  } |
| **[실행 결과]** |