|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 <cstdlib>  #include<ctime>  using namespace std;  class Nation {  string nation;  string capital;  public:  Nation() { }    void setNation(string na, string ca) {  nation = na;  capital = ca;  }  string getNation() {  return nation;//nation 리턴  }  string getCapital() {  return capital; //capital 리턴  }  };  int main() {  vector<Nation> v; // 저장할 객체 생성  Nation n;  n.setNation("한국", "서울");  v.push\_back(n);  n.setNation("일본", "도쿄");  v.push\_back(n);  n.setNation("미국", "와싱턴");  v.push\_back(n);  n.setNation("중국", "베이징");  v.push\_back(n);  n.setNation("필리핀", "마닐라");  v.push\_back(n);  n.setNation("독일", "베를린");  v.push\_back(n);  n.setNation("스페인", "리스본");  v.push\_back(n);  n.setNation("이탈리아", "로마");  v.push\_back(n);  n.setNation("베트남", "하노이");  v.push\_back(n);  srand((unsigned)time(0)); // 중복없는 난수 생성  int sel;    bool exit = true; // 나가면 true    string na, ca;    int num = v.size() + 1;  int choice = 0;  int random;  cout << "\*\*\*\*\* 나라의 수도 맞추기 게임을 시작합니다. \*\*\*\*\*" << endl;  while (exit) {  cout << "정보 입력: 1, 퀴즈: 2, 종료: 3 >> ";  cin >> sel;  switch (sel) {    case 1:  cout << "현재 " << v.size() << "개의 나라가 입력되어 있습니다." << endl;    cout << "나라와 수도를 입력하세요(no no 이면 입력끝)" << endl;    while (1) {  cout << num << ">>";  cin >> na >> ca;  for (int i = 0; i < num - 1; i++) {  if (v[i].getNation() == na) {  cout << "already exists !!" << endl;  choice = 1;  break;  }  }    if (na == "no" && ca == "no") break;    else if (choice == 0) {  n.setNation(na, ca);  v.push\_back(n);  num++;  }    choice = 0;  }  break;    case 2:    while (1) {  random = rand() % v.size();  cout << v[random].getNation() << "의 수도는?";  cin >> ca;    if (ca == "exit") break;    if (v[random].getCapital() == ca)  cout << "Correct !!" << endl;    else  cout << "NO !!" << endl;  }  break;    case 3:  exit = false;    break;  }    }    char ch;  ch = getchar();  } |
| **[실행 결과]** |

1. 교재 530p : 문제 14번

|  |
| --- |
| **[프로그램 소스]** |
| **[실행 결과]** |