창의적 소프트웨어 설계

7주차 실습 – Class and Inheritance

노인우, inwoo13@hanyang.ac.kr
한중수, soohan@hanyang.ac.kr

Overview

목표

- C++ struct & class
- Memory management
- Pointer, reference, and const
- C++ STL
- Inheritance

C++ struct and class

```
#include <string>
#include <vector>

#include <iostream>

struct StudentInfo
{
   StudentInfo(int _id, std::string _name, int _score){
      id = _id;
      name = _name;
      score = _score;
   }

   int ShowInfo() const {
      std::cout << id << " | " << name << " | " << score << std::endl;

      return 0;
   }

   int id;
   std::string name;
   int score;
};
```

C++ struct and class

```
class StudentVector{
public:
    StudentVector(){}
    int AddStudent(const std::string& _name, const int _id, const int _score);
    bool CheckDuplicateId(int _id) const;
    int RemoveStudent(const std::string& str);
    int ShowStudents() const;
    int GetStudentByIndex(int num) const;

private:
    std::vector<StudentInfo> student_vec;
};
```

C++ struct and class

```
bool StudentVector::CheckDuplicateId(int _id) const {
   std::vector<StudentInfo>::const_iterator pos;

   for(pos = student_vec.begin() ; pos != student_vec.end() ; ++pos){
      if(pos->id == _id){
         return false;
      }
   }

   return true;
}


int StudentVector::AddStudent(const std::string& _name, const int _id, const int _score){
   if(CheckDuplicateId(_id) == false){
      std::cout << "[Error] Dulicated ID: " << _id << ", Name: " << _name << std::endl;
      return -1;
   }

   StudentInfo si(_id, _name, _score);
   student_vec.push_back(si);

   return 0;
}
```

C++ struct and class

```
int StudentVector::ShowStudents() const {
   std::vector<StudentInfo>::const_iterator pos;

   std::cout << "\nStudent List: " << std::endl;

   for(pos = student_vec.begin() ; pos != student_vec.end() ; ++pos){
      pos->ShowInfo();
   }

   return 0;
}

int main(){
   StudentVector sv;

   sv.AddStudent("Yuna Kim", 1, 100);
   sv.AddStudent("JS Han", 2, 99);
   sv.AddStudent("IW Ro", 2, 77);

   sv.ShowStudents();

   return 0;
}
```

Pointer and Reference

```
int StudentVector::AddStudent(const std::string& _name, const int _id, const int _score){
    if(CheckDuplicateId(_id) == false){
        std::cout << "[Error] Dulicated ID: " << _id << ", Name: " << _name << std::endl;
        return -1;
    }

    StudentInfo si(_id, _name, _score);
    student_vec.push_back(si);

    si.name = "Sherlock";

    return 0;
}
```

Pointer and Reference

```
int StudentVector::AddStudent(const std::string& _name, const int _id, const int _score){
    if(CheckDuplicateId(_id) == false){
        std::cout << "[Error] Dulicated ID: " << _id << ", Name: " << _name << std::endl;
        return -1;
    }

    StudentInfo si(_id, _name, _score);
    student_vec.push_back(si);

    si.name = "Sherlock";

    return 0;
}
```

Pointer and Reference

```
int StudentVector::AddStudent(const std::string& _name, const int _id, const int _score){
   // do it with pointer vector!
   StudentInfo *psi = new StudentInfo(_id, _name, _score);
}
```

참고자료

1. **preprocessor technique**, https://stackoverflow.com/questions/3246803/why-use-ifndef-class-h-and-define-class-h-in-h-file-but-not-in-cpp


Q&A

Appendix #1. C/C++ 동시 고려 헤더 작성

```
 // test.h
#ifndef _TEST_H
#define _TEST_H

#ifdef __cplusplus
extern “C” {
#endif

int test(int);

#ifdef __cplusplus
}
#endif

#endif
```