

3장. 실전문제

박 종 혁 교수
(서울과기대 컴퓨터공학과)

Tel: 970-6702
Email: jhpark1@seoultech.ac.kr

실전문제(1)

- 다음에서 Person은 사람을, Family는 가족을 추상화한 클래스로서 완성되지 않은 클래스이다.

```

4 class Person{
5     string name;
6
7     public:
8         Person(string name){ this->name = name; }
9         string getName() { return name; }
10 };
11
12 class Family{
13     Person *p; // Person 배열 포인터
14     int size; // Person 배열의 크기, 가족 구성원 수
15
16     public:
17         Family(string name, int size); // size 개수만큼 Person 배열 동적 생성
18         void show(); // 모든 가족 구성원 출력
19         ~Family();
20 };

```

- 다음 main()이 작동하도록 Person과 Family 클래스에 필요한 멤버들을 추가하고 코드를 완성하라.

```

27 int main(){
28     Family *simpson = new Family("simpson", 3);
29     simpson->setName(0, "Mr. simpson");
30     simpson->setName(1, "Mrs. simpson");
31     simpson->setName(2, "Bart. simpson");
32     simpson->show();
33     delete simpson;
34 }

```

실전문제(1) - 소스코드

```
#include <iostream>
#include <string>
using namespace std;

class Person {
    string name;

public:
    Person() { name=""; }
    Person(string name) { this->name = name; }
    string getName() { return name; }
    void setName(string name) { this->name =
name; }
};

class Family {
    string name;
    Person* p; // Person 배열 포인터
    int size; // Person 배열의 크기. 가족 구성원 수

public:
    Family(string name, int size); // size 개수만큼
    Person 배열 동적 생성
    void setName(int index, string name);
    void show(); // 모든 가족 구성원 출력
    ~Family();

};

Family::Family(string name, int size) {
    this->name = name;
    this->size = size;
    p = new Person [size];
}
```

```
Family::~Family() {
    delete [] p;
}

void Family::setName(int index, string name) {
    p[index].setName(name);
}

void Family::show() {
    cout << name + " 가족은 다음과 같이 " << size << "명 입니다." << endl;
    for(int i=0; i<size; i++) {
        cout << p[i].getName() << '\t';
    }
    cout << endl;
}

int main() {
    Family *simpson = new Family("Simpson", 3); // 3명으로 구성된
    Simpson 가족
    simpson->setName(0, "Mr. Simpson");
    simpson->setName(1, "Mrs. Simpson");
    simpson->setName(2, "Bart Simpson");
    simpson->show();
    delete simpson;
}
```

실전문제(2)

- 다음은 반지름을 속성으로 가진 Circle 클래스 이다.

```
#include <iostream>
using namespace std;

class Circle {
    int radius; // 원의 반지름 값
public:
    void setRadius(int radius); // 반지름을 설정한다.
    double getArea(); // 면적을 리턴한다.
};
```

실전문제(2)

- 키보드에서 원의 개수를 입력받고, 그 개수만큼 원의 반지름을 입력받고, 다음과 같이 실행되도록 main()함수를 작성하라. Circle클래스를 완성하라

실행 결과

```
원의 개수 >> 3
원 1의 반지름 >> 10
원 2의 반지름 >> 2
원 3의 반지름 >> 1
100보다 큰 원을 검색합니다.
면적이 100보다 큰 원은 1개 입니다
```

실전문제(2) - 소스코드

```
#include <iostream>
using namespace std;

class Circle {
    int radius; // 원의 반지름 값
public:
    void setRadius(int radius); // 반지름을 설정한다.
    double getArea(); // 면적을 리턴한다.
};

void Circle::setRadius(int radius) {
    this->radius = radius;
}

double Circle::getArea() {
    return 3.14*radius*radius;
}
```

```
int main() {
    Circle *p;
    int size;
    cout << "원의 개수 >> ";
    cin >> size; // 원의 개수 입력
    if(size < 1)
        return 0;

    p = new Circle[size]; // 객체 배열 동적 생성
    int radius;
    for(int i=0; i<size; i++) {
        cout << "원 " << i+1 << "의 반지름 >> ";
        cin >> radius;
        p[i].setRadius(radius);
    }
    int count = 0;
    for(int i=0; i<size; i++) {
        if(p[i].getArea() > 100) {
            count++;
        }
    }
    cout << "면적이 100보다 큰 원은 " << count << "개 입니
다" << endl;
    delete [] p; // 객체 배열 소멸
}
```

실전문제(3)

- 다음은 커피자판기로 작동하는 프로그램을 만들기 위해 필요한 두 클래스이다.

```

4 class CoffeeVendingMachine{ // 커피자판기를 표현하는 클래스
5     Container tong[3]; // tong[0]는 커피, tong[1]은 물, tong[2]는 설탕통을 나타냄
6     void fill(); // 3개의 통을 모두 10으로 채움
7     void selectEspresso(); // 에스프레소를 선택한 경우, 커피 1, 물 1 소모
8     void selectAmericano(); // 아메리카노를 선택한 경우, 커피 1, 물 2 소모
9     void selectSugarCoffee(); // 설탕커피를 선택한 경우, 커피 1, 물 2, 설탕 1 소모
10    void show(); // 현재 커피, 물, 설탕의 잔량 출력
11    public:
12        void run(); // 커피 자판기 작동
13 };
14
15 class Container{ // 통 하나를 나타내는 클래스
16     int size; // 현재 저장량, 최대 저장량은 10
17     public:
18         Container(){ size=10; }
19         void fill(); // 최대량(10)으로 채우기
20         void consume(); // 1만큼 소모하기
21         int getSize(); // 현재 크기 리턴
22 }

```

실전문제(3)

- 다음과 같이 실행되도록 main() 함수와 CoffeeVendingMachine, Container를 완성하라. 만일 커피, 물, 설탕 중 잔량이 하나라도 부족해 커피를 제공할 수 없는 경우 ‘원료가 부족합니다.’를 출력하라

실행 결과

***** 커피자판기를 작동합니다 *****

메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 4

커피 10, 물 10, 설탕 10

메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 1

에스프레소 드세요

메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 4

커피 9, 물 9, 설탕 10

메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 3

에스프레소 드세요

메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 4

커피 8, 물 7, 설탕 9

메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>> 5

커피 10, 물 10, 설탕 10

메뉴를 눌러주세요(1:에스프레소, 2:아메리카노, 3:설탕커피, 4:잔량보기, 5:채우기)>>

실전문제(4)

- 겐블링 게임을 만들어보자. 두 사람이 게임을 진행하며, 선수의 이름을 초기에 입력받는다. 선수가 번갈아 자신의 차례에서 <Enter> 키를 치면 랜덤한 3개의 수가 생성되고 모두 동일한 수가 나오면 게임에서 이기게 된다. 숫자의 범위는 0~2로 제한한다. 선수는 Player 클래스로 작성하고, 2명의 선수는 배열로 구성하라. 그리고 게임은 GamblingGame 클래스로 작성하라
- (힌트)
 - 랜덤 수를 발생시키기 위해 srand()와 rand() 함수를 활용

실행 결과

***** 겐블링 게임을 시작합니다. *****

첫번째 선수 이름>>수연이

두번째 선수 이름>>제갈이

수연이:<Enter>

2 1 2 아쉽군요!

제갈이:<Enter>

1 0 2 아쉽군요!

수연이:<Enter>

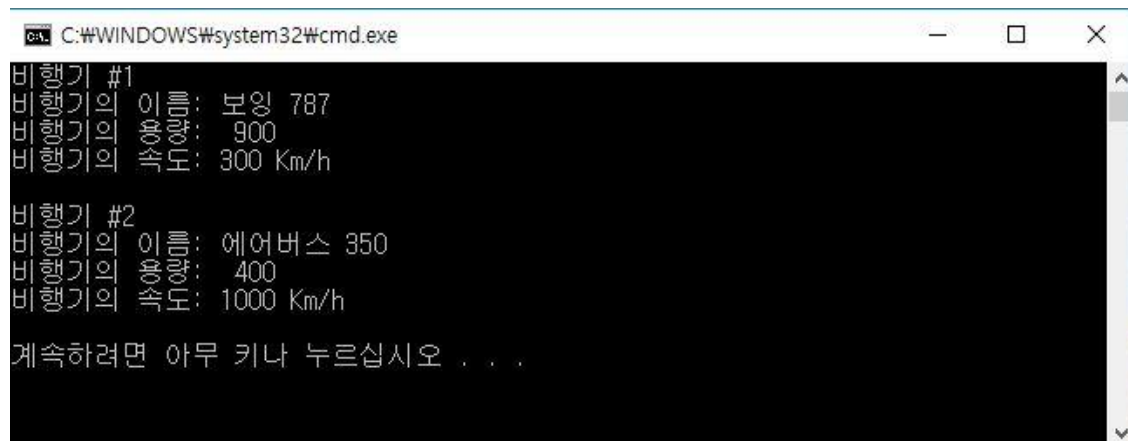
2 2 1 아쉽군요!

제갈이:<Enter>

0 0 0 수연이님 승리!

실전문제(5)

- 비행기를 나타내는 Airplane라는 이름의 클래스를 설계하라. Airplane 클래스는 이름(name), 승객수(capacity), 속도(speed)를 멤버 변수로 가지고 있다
 - 멤버 변수를 정의하라. 모든 멤버 변수는 전용 멤버로 하라
 - 모든 멤버 변수에 대한 접근자와 설정자 멤버 함수를 작성한다
 - Airplane 클래스의 생성자 몇 개를 중복 정의하라. 생성자는 모든 데이터를 받을 수도 있고 아니면 하나도 받지 않을 수 있다
 - Airplane 객체의 현재 상태를 콘솔에 출력하는 print() 함수도 포함시켜라
 - Main () 에서 Plane 객체 2개를 생성하고 접근자와 설정자를 호출하여 보라



```
C:\WINDOWS\system32\cmd.exe
비행기 #1
비행기의 이름: 보잉 787
비행기의 용량: 900
비행기의 속도: 300 Km/h

비행기 #2
비행기의 이름: 에어버스 350
비행기의 용량: 400
비행기의 속도: 1000 Km/h

계속하려면 아무 키나 누르십시오 . . .
```

실전문제(6)

- 상자를 나타내는 Box 라는 이름의 클래스를 설계하라. Box 클래스는 상자의 길이(length), 너비(width), 높이(height) 등의 변수를 가진다
 - 멤버 변수는 전용 멤버로 한다
 - Box 클래스의 생성자를 중복 정의하라. 생성자는 모든 데이터를 받을 수도 있고 아니면 하나도 받지 않을 수 있다. 멤버 초기화 리스트를 사용해본다
 - 접근자와 생성자를 추가할 박스가 비어 있는지 그렇지 않은지를 나타내는 empty () 라는 멤버 함수도 추가한다. 부피를 계산하는 getVolume () 멤버 함수도 추가한다. Box 객체의 현재 상태를 콘솔에 출력하는 print () 함수도 포함시켜라
 - Main () 에서 Box 객체 여러 개를 생성하고 접근자와 설정자를 호출하여 보라

```

상자 #1
상자의 길이: 0
상자의 너비: 0
상자의 높이: 0
상자의 부피: 0
  
```

```

상자 #2
상자의 길이: 3
상자의 너비: 2
상자의 높이: 4
상자의 부피: 24
  
```

실전문제(7)

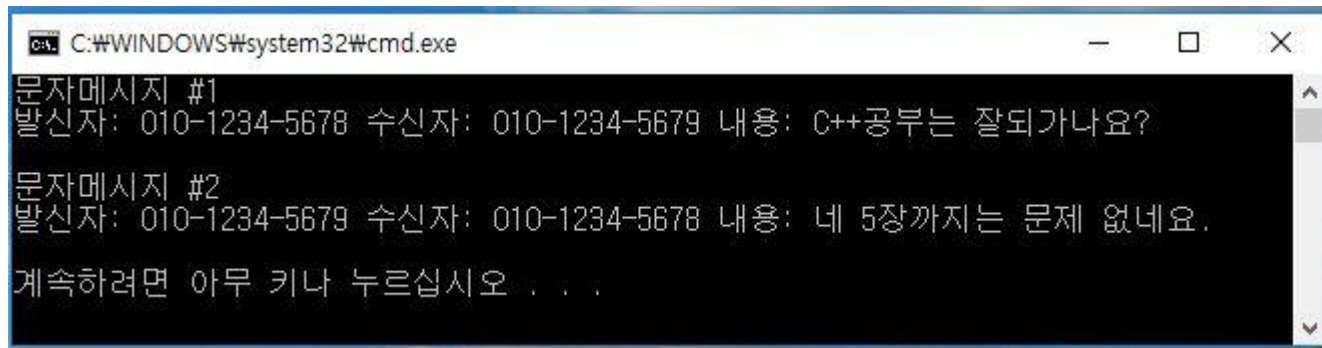
- 영화를 나타내는 Movie라는 이름의 클래스를 설계하라. 제목(title), 감독(director), 평점(rating)을 나타는 멤버 변수를 가진다
 - 멤버 변수는 전용 멤버로 한다
 - Movie 클래스의 생성자를 중복 정의한다. 생성자는 모든 데이터를 받을 수도 있고 아니면 하나도 받지 않을 수 있다
 - 접근자와 생성자를 비롯하여 필요한 멤버 함수들을 추가하라
 - Main () 에서 Movie 객체 여러 개를 생성하고 접근자와 설정자를 호출하여 보라

```
영화 #1
영화 제목 : 타이타닉
영화 감독 : 제임스 카메론
영화 평점 : 9.5
```

```
영화 #2
영화 제목 : 지오스톰
영화 감독 : 딘 데블린
영화 평점 : 8.34
```

실전문제(8)

- 핸드폰의 문자 메시지를 나타내는 클래스 SNS를 작성하여 보자. SMS는 송신자의 전화번호(sender), 수신자의 전화번호(receiver), 메시지 텍스트(text) 등의 데이터를 가진다
 - 멤버 변수는 전용 멤버로 한다
 - 적절한 생성자를 추가한다
 - 멤버 함수 print () 는 메시지에 헤더를 붙여서 콘솔에 추가한다. 기타 적절한 접근자와 설정자를 추가한다
 - 몇 개의 문자 메시지를 생성하여서 테스트하라



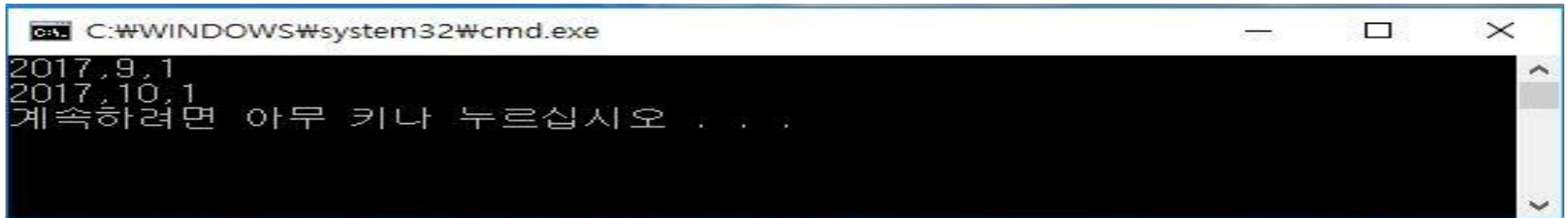
```
C:\WINDOWS\system32\cmd.exe
문자메시지 #1
발신자: 010-1234-5678 수신자: 010-1234-5678 내용: C++공부는 잘되어나요?

문자메시지 #2
발신자: 010-1234-5678 수신자: 010-1234-5678 내용: 네 5장까지는 문제 없네요.
계속하려면 아무 키나 누르십시오 . . .
```

실전문제(9)

- 날짜를 나타내는 Date 클래스를 작성해보자. Date 클래스는 year, month, day 를 멤버 변수로 가진다. 화면에 자기 자신의 상태를 출력하는 print () 함수를 가지고 있도록 가정한다. print () 는 다양한 형식으로 날짜를 출력할 수 있도록 중복 정의 해본다. nextMonth() 는 다음 달을 나타내는 객체를 반환한다. isLeapYear() 는 윤년이면 true 를 반환한다. 몇 개의 객체를 생성하여 테스트하라

```
Date nextMonth();
Bool isLeapYear(); // 윤년이면 true 반환
```



```
C:\WINDOWS\system32\cmd.exe
2017,9,1
2017,10,1
계속하려면 아무 키나 누르십시오...
```

```
Class Date {
    int year;
    int month;
    int day;
Public:
    Date();
    Date(int y, int m, int d);
    bool isLeapYear(int y);
    void print() const;
    Date nextMonth();
    ...
};
```

실전문제(10)

- 실습 문제 (4) 에서 우리는 핸드폰의 문자 메시지를 나타내는 클래스 SMS를 작성한 적이 있다. SMS 는 송신자의 전화 번호 (sender), 수신자의 전화 번호(receiver), 메시지 텍스트(text) 등의 데이터를 가진다. 몇 개의 문자 메시지를 다음과 같은 방법으로 생성하여서 테스트하라

- 전통적인 객체 배열을 이용하여 3개의 문자메시지를 생성하여 저장해보자
- 벡터를 이용하여 3개의 문자메시지를 동적으로 생성하여 저장해보자

발신자: 010-1234-5678 수신자: 010-1234-5679 내용: c++공부는 잘되나요?

발신자: 010-1234-5679 수신자: 010-1234-5678 내용: 네 5장까지는 문제 없네요.

```

Class SMS
{
    string sender;
    string receiver;
    string text;

Public:
    SMS() {
        sender = "";
        receiver = "";
        text = "";
    }
    ...
}
  
```

실전문제(11)

- 학생을 나타내는 클래스 Student를 정의한다. Student는 이름 (name), 성적 (marks) 을 멤버 변수로 가진다. 벡터를 이용하여서 동적 객체 배열을 만든다. 사용자로부터 데이터를 입력 받아서 배열에 저장한 후에 sort() 함수를 사용하여 학점이 가장 높은 순으로 모든 학생들의 정보를 출력하라

정렬을 위해 `#include <algorithm>` 추가해야함
`sort(vec.begin(), vec.end(), compare);` 사용

```
Class Student {
    string name;
    double marks;
Public:
    Student(string n, double m) : name{n}, marks{m} { }
    ....
};
```

```
C:\WINDOWS\system32\cmd.exe
학생 1
이름 : 최 차영
학점 : 4.3
학생 2
이름 : 김 철 수
학점 : 3.9
학생 3
이름 : 김 영 희
학점 : 3.8
=====
이름 : 최 차영
학점 : 4.3
이름 : 김 철 수
학점 : 3.9
이름 : 김 영 희
학점 : 3.8
=====
계속하려면 아무 키나 누르십시오 . . .
```


실전문제(12)

- 다음과 같이 연락처를 나타내는 Contact 클래스를 작성한다. Contact 클래스는 이름(name)과 전화번호(tel)를 가지고 있다. 3개의 Contact 객체를 저장하는 동적 배열을 정의하고 사용자로부터 연락처 정보를 받아서 동적 배열 안에 저장한다. 이어서 사용자로부터 이름을 받아서 동적 배열에서 탐색한 후에 연락처 전화번호를 출력하는 프로그램을 작성한다

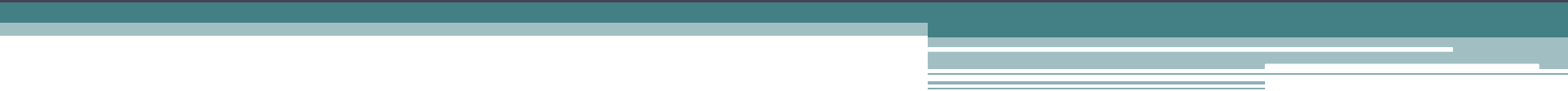
 선택 C:\WINDOWS\system32\cmd.exe

```
이름을 입력하시오 : 홍길동
전화번호를 입력하시오 : 010-1111-2222
이름을 입력하시오 : 김철수
전화번호를 입력하시오 : 010-1111-3333
이름을 입력하시오 : 최자영
전화번호를 입력하시오 : 010-1111-7777
탐색하고 싶은 이름을 입력하시오 : 홍길동
전화번호 : 010-1111-2222
계속하려면 아무 키나 누르십시오 . . .
```

```
Class Contact {
Private:
    string name;
    string tel;
    ...
}
```

참고문헌

- 뇌를 자극하는 C++ 프로그래밍, 이현창, 한빛미디어, 2011
- 열혈 C++ 프로그래밍(개정판), 윤성우, 오렌지미디어, 2012
- 객체지향 원리로 이해하는 ABSOLUTE C++ , 최영근 외 4명 , 교보문고, 2013
- C++ ESPRESSO, 천인국, 인피니티북스, 2011
- 명품 C++ Programming, 황기태 , 생능출판사, 2018
- 어서와 C++는 처음이지, 천인국, 인피니티북스, 2018



Q&A