

LAB #08 (11/08 목)

1. 아래의 코드를 기반으로 Polygon class을 상속받는 Rectangle class를 선언하고 둘레와 넓이를 구하는 프로그램을 작성하라. 단, 정사각형이라 가정한다.

```
int main()
{
    Polygon pol;
    Rectangle rec(4,10);

    cout << "--- Polygon class ---" << endl;
    pol.calcGirth();
    pol.calcArea();
    cout << "--- Rectangle class ---" << endl;
    rec.calcGirth();
    rec.calcArea();

    return 0;
}
```

[참조 1]

```
class Polygon {
public:
    Polygon();
    Polygon(int point, float length);
    ~Polygon();

    virtual void calcGirth();
    virtual void calcArea();

protected:
    int mPoint; // 꼭지점 갯수
    double mLength; // 한 변의 길이
};
```

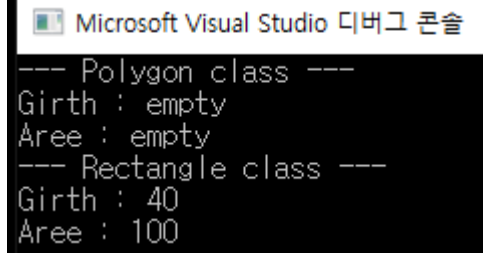
[참조 2]

```
class Rectangle : public Polygon
{
public:
    Rectangle() {};
    Rectangle(int point, float length) : Polygon(point, length);
    ~Rectangle() {};

    void calcGirth() override;

    void calcArea() override;
};
```

1 – 출력화면 :



```
Microsoft Visual Studio 디버그 콘솔
--- Polygon class ---
Girth : empty
Aree : empty
--- Rectangle class ---
Girth : 40
Aree : 100
```

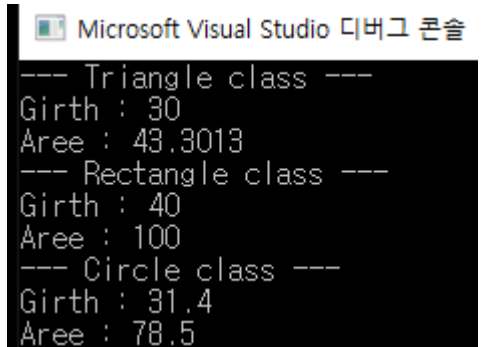
2. 1번 문제에서 작성한 프로그램을 기반으로 Polygon class를 상속하는 Triangle, Rectangle, Circle class를 선언하고 둘레와 넓이를 구하는 프로그램을 작성하라. 단, 모든 도형은 정다각형이라 가정한다.

```
int main()
{
    Triangle tri(3, 10);
    Rectangle rec(4, 10);
    Circle cir(0, 5);

    cout << "--- Triangle class ---" << endl;
    tri.calcGirth();
    tri.calcArea();
    cout << "--- Rectangle class ---" << endl;
    rec.calcGirth();
    rec.calcArea();
    cout << "--- Circle class ---" << endl;
    cir.calcGirth();
    cir.calcArea();

    return 0;
}
```

2 – 출력화면 :



```
Microsoft Visual Studio 디버그 콘솔
--- Triangle class ---
Girth : 30
Aree : 43.3013
--- Rectangle class ---
Girth : 40
Aree : 100
--- Circle class ---
Girth : 31.4
Aree : 78.5
```

3. Ktx 열차가 출발역, 종착역을 포함하여 총 5개의 역에 정차한다. 이 때 각 역에서는 입력된 숫자만큼의 사람이 내리거나 탑승한다. 이 기차는 아래와 같은 조건을 만족하면서 운행된다고 가정한다.

1. 기차는 역 번호 순서대로 운행한다.
2. 출발역에서 내린 사람 수와 종착역에서 탄 사람 수는 0이다.

3. 각 역에서 현재 기차에 있는 사람보다 더 많은 사람이 내리는 경우는 없다.
 4. 기차의 정원은 최대 300명이고, 정원을 초과하여 타는 경우는 없다.
- 5개의 역에 대해 기차에서 내린 사람 수와 탄 사람 수가 주어졌을 때, 기차에 사람이 가장 많을 때의 사람 수를 계산하는 프로그램을 작성하라.

```
int main()
{
    Ktx k;

    // ???

    return 0;
}
```

[참조 1]

```
class Train {
public:
    Train() {}
    Train(int people)
    {
        mPeople = people;
    }
    ~Train() {}

    virtual int station(int takeOff, int takeOn);

protected:
    int mPeople; // 사람 수
};
```

[참조 2]

```
class Ktx : public Train
{
public:
    Ktx() : Train(0) {}
    Ktx(int people) : Train(people)
    {}
    ~Ktx() {}

    // 기차에 사람이 타고 내리는 함수
    int station(int takeOff, int takeOn);

    int getPeople();
};
```

3 – 출력화면 :

```

Microsoft Visual Studio 디버그 콘솔
1번역 : 0 210
2번역 : 40 63
3번역 : 50 20
4번역 : 27 25
5번역 : 201 0
가장 많은 사람이 탑승 했을 때의 사람 수 = 233

Microsoft Visual Studio 디버그 콘솔
1번역 : 0 250
2번역 : 18 84
정원초과입니다

Microsoft Visual Studio 디버그 콘솔
1번역 : 0 210
2번역 : 143 34
3번역 : 200 18
정원미달입니다

```

4. [사진관리 프로그램 작성을 위한 단계 3]

lab #7에서 작성한 프로그램을 다음과 같이 확장한다.

- 사진파일 정보를 class로 표현한다.
- Main 함수에 선언했던 변수(객체)와 함수들을 포함하는 application class 정의한다.
- Lab 7에서 구현하였던 모든 기능을 class를 이용하여 구현한다.

보다 구체적인 내용은 아래와 같다.

1) 하나의 사진파일 정보와 이를 이용하기 위한 기능을 포함하는 class(photoType)을 정의한다.

- photoType class는 다음과 같은 멤버변수를 포함한다.
 - i. 파일명(photoName): string Type, 촬영한 날자로 생성한 primary key
 - ii. 파일크기(size): int type, 파일의 크기
 - iii. 이벤트명(eventName): string type, "2017 수학여행"과 같은 이벤트 명
 - iv. 포함된 내용(contents): string Type, 사진에 있는 모든 사람들의 이름과 장소명을 포함하는 text file. 이름은 space로 분리한다.
 1. 예: 홍길동 김복동 남한산성
- photoType class의 예는 [참조 1]과 같다.

2) Lab 7에서 작성한 main()에 선언했던 변수와 함수를 포함하는 application class를 정의한다.

- 멤버 변수로는 다음과 같은 것이 포함된다.
 - i. photoList: 사진정보를 저장할 list로 vector를 이용하고 구현
 - ii. inFileName, outFileName : 입력파일과 출력파일의 이름
 - iii. length: 현재 저장된 사진 수
- Application class의 예는 [참조 2]와 같다

- 3) 프로그램을 단순화하기 위하여 실습 8에서는 contents가 하나의 단어로 구성된다고 가정한다. 한단어로 구성될 경우에는 읽기와 쓰기가 단순해진다.
- 4) 구현된 프로그램 실행에는 [참조 3]과 같다.

[참조 1] RecordType class ADT

```
using namespace std;

class RecordType {
public:
    int ReadRecordFromKB(); // read a record from keyboard
    int ReadPhotoNameFromKB(); // read photo name from keyboard
    int ReadEventNameFromKB(); // read event name from Keyboard
    int ReadRecordFromFile(ifstream& inFile); // read a record from file
    void WriteRecordToFile(ofstream& outFile); // write a record to file
    void GenFileName(string& name); //generate an unique file name using the
current time.
    void DisplayOnScreen(); // display a record on screen
    bool IsEqual(string name); // compare photoName with input string
private:
    string photoName;
    string eventName;
    int size;
    string contents;
};
```

[참조 2] ApplicationType class ADT

```
using namespace std;

class ApplicationType
{
public:
    ApplicationType(){ photoList.clear(); length=0; inFileName="test.txt";
outFileName="test.txt";};
    int AddRecordToList(); // read a record from keyboard and add to the list
    int ReadAllFromFile(); // read all the records in file and store in vector arrays
    void DisplayAllOnScreen(); // write all the record in the list on the screen
    int WriteAllToFile(); // write all photo records to disk.
    int GetCommand(); // get an integer representing a proper command
    void Run(); // select a command and execute
    int SearchByPrimaryKey(string inName); // Search a record by file name and return
index of the record.
    // Search a record by file name and return index of the record.
    void FindAndDisplay(); // find a record by name and display
    void DeleteByName(); // find a record by name and delete
private:
    vector<RecordType> photoList;
    int length;
    string inFileName;
```

```

        string outFileNames;
};

```

[참조 3]

```

***** [   Display on screen   ] *****
Record 0   : aaa               , e111 12  c111

Record 1    : ccc               , e333 23  c333

Record 2    : bbb               , e222 323 c222

--- ID - Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Find by file name and display on screen |
| 6. Find by file name and delet from list |
| 0. Quit |
-----
Choose a Command--> 1

***** [   Add a Record to list   ] *****
Photo Name(Enter 'gen' to generate) --> eeeee
Event Nmae                          --> 졸업식
Event Size                          --> 23
Contents                            --> 김병식

--- ID - Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Find by file name and display on screen |
| 6. Find by file name and delet from list |
| 0. Quit |
-----
Choose a Command--> 4

***** [   Display on screen   ] *****
Record 0   : aaa               , e111 12  c111

Record 1    : ccc               , e333 23  c333

Record 2    : bbb               , e222 323 c222

Record 3    : eeeee           , 졸업식 23  김병식

--- ID - Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Find by file name and display on screen |
| 6. Find by file name and delet from list |
| 0. Quit |
-----
Choose a Command--> 3

***** [   Write to disk   ] *****

```

```
----- [ Write to disk ] -----
--- ID - Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Find by file name and display on screen |
| 6. Find by file name and delet from list |
| 0. Quit |
-----
```

Choose a Command--> 2

```
***** [ Read from disk ] *****
```

```
--- ID - Command -----
| 1. Add a new record to list |
| 2. Read all records from disk |
| 3. Write all records to disk |
| 4. Display all record on screen |
| 5. Find by file name and display on screen |
| 6. Find by file name and delet from list |
| 0. Quit |
-----
```

Choose a Command--> 4

```
***** [ Display on screen ] *****
```

Record 0	:	aaa	,	e111	12	c111
Record 1	:	ccc	,	e333	23	c333
Record 2	:	bbb	,	e222	323	c222
Record 3	:	eeee	,	졸업식	23	김병식