2번

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

class Converter {
protected:
    double ratio;
    virtual double convert(double src) =0; // src를 다른 단위로 변환
    virtual string getSourceString() =0; // src 단위 명칭
    virtual string getDestString() =0; // dest 단위 명칭
public:
    Converter(double ratio) {
        this->ratio = ratio; // ratio값은 1mile값
    }
    void run() {
        double src; // 입력받을 km 값 받을 변수src
        cout << getSourceString() << "을 " << getDestString() << "로 바꿉니다. ";
        cout << getSourceString() << "을 입력하세요>> ";
        cin >> src; // km값 입력받음
        cout << "변환 결과 : " << convert(src) << getDestString() << endl;
    }
};

class KmToMile : public Converter { // 추상클래스를 상속받은 KmToMile
public:
    KmToMile(double ratio = 0.0) : Converter(ratio) { } // src와 ratio가 double형이므로 초기값으로 0.0을 넣어줌

    string getSourceString() { return "Km"; } //  순수가상함수 구현 -> 온전한 클래스
    string getDestString() { return "Mile"; } //  순수가상함수 구현 -> 온전한 클래스

    double convert(double src) {// src는 변환하고 싶은 km이고 ratio는 1mile 값 // ratio값은 추상클래스 Converter에서 정의된 멤버
        return src / ratio;
    }
};

int main() {
    KmToMile toMile(1.609344); // 1 mile은 1.609344 km
    toMile.run();

    return 0;
}
```

```
C:₩WINDOWS₩system32₩cmd.exe
Km을 Mile로 바꿉니다. Km을 입력하세요>> 25
변환 결과 : 15.5343Mile
Press any key to continue . . .
```

4번

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

class LoopAdder { // 추상 클래스
    string name; // 루프의 이름
    int x, y, sum; // x에서 y까지의 합은 sum
    void read(); // x, y 값을 읽어 들이는 함수
    void write(); // sum을 출력하는 함수
protected:
    LoopAdder(string name ="") { // 루프의 이름을 받는다. 초깃값은 ""
        this->name = name;
    }
    int getX() { return x; }
    int getY() { return y; }
    virtual int calculate() = 0; // 순수가상함수. 루프를 돌며 합을 구하는 함수
public:
    void run(); // 연산을 진행하는 함수
};

void LoopAdder::read() { // x, y 입력
    cout << name << ":" << endl;
    cout << "처음 수에서 두번째 수까지 더합니다. 두 수를 입력하세요 >> ";
    cin >> x >> y; // 두 수를 입력 받음 // 여기서 x,y는 LoopAdder에 정의된 멤버
}
void LoopAdder::write() { // 결과 sum 출력
    cout << x << "에서 " << y << "까지의 합 = " << sum << "입니다" << endl;
}
void LoopAdder::run() {
    read(); // x, y를 읽는다.
    sum = calculate(); // 루프를 돌면서 계산한다.
    write(); // 결과 sum을 출력한다.
}
```

```
class WhileLoopAdder : public LoopAdder { // 추상클래스 LoopADder를 상속받는 WhileLoopAdder 클래스
public:
    WhileLoopAdder(string name = "") : LoopAdder(name) {} // 루프의 이름 초기값 설정
    int calculate() { // 순수가상함수인 calculate()를 파생클래스인 WhileLoopAdder에서 구현함으로써 온전한 클래스로 만듬
        int SumCount = 0; // sum합산을 저장할 SumCount 변수
        int i = getX(); // x~y까지의 값을 이어나갈 i변수 // 초기값은 x값으로 설정
        while ( i <= getY()) { // x~y까지 비교해서 x가 y를 넘으면 while문 탈출
            SumCount += i; // 합산 계산
            i++; // i값 증가
        }
        return SumCount; // 합계 리턴
    }
};

class DoWhileLoopAdder : public LoopAdder { // 추상클래스 LoopADder를 상속받는 DoWhileLoopAdder 클래스
public:
    DoWhileLoopAdder(string name = "") : LoopAdder(name) {} // 루프의 이름 초기값 설정
    int calculate() {  // 순수가상함수인 calculate()를 파생클래스인 DoWhileLoopAdder에서 구현함으로써 온전한 클래스로 만듬
        int SumCount = 0;  // sum합산을 저장할 SumCount 변수
        int i = getX(); // x~y까지의 값을 이어나갈 i변수 // 초기값은 x값으로 설정
        do {
            SumCount += i;  // 합산 계산
            i++; // i값 증가
        } while (i <= getY()); // do-while문이므로 do문 먼저 실행 후에 조건 검사
        return SumCount;  // 합계 리턴
    }
};

int main() {
    WhileLoopAdder whileLoop("while Loop"); //while문을 이용하여 합을 구하는 과정을 수행
    DoWhileLoopAdder doWhileLoop("Do while Loop"); // do-while문을 이용해 합을 구하는 과정을 수행

    whileLoop.run();
    doWhileLoop.run();

    return 0;
}
```

```
C:₩WINDOWS₩system32₩cmd.exe
while Loop:
처음 수에서 두번째 수까지 더합니다. 두 수를 입력하세요 >> 3 5
3에서 5까지의 합 = 12입니다
Do while Loop:
처음 수에서 두번째 수까지 더합니다. 두 수를 입력하세요 >> 10 20
10에서 20까지의 합 = 165입니다
Press any key to continue . . .
```

8번

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

class Shape { // 최소 하나의 순수가상함수를 가지는 클래스이므로 Shape은 추상클래스
protected:
    string name; // 도형의 이름
    int width, height; // 도형이 내접하는 사각형의 너비와 높이
public:
    Shape(string n ="", int w=0, int h=0) { name = n; width = w; height = h; }
    virtual double getArea( ) = 0; // dummy값 리턴 // getArea( ) 함수를 =0; 을 취하므로 순수 가상함수로 만듦
    string getName( ) { return name; } // 이름 리턴
};

class Oval :public Shape { // 추상클래스 Shape을 상속받는 파생클래스 Oval
public:
    Oval(string n, int w, int h) : Shape(n, w, h) {} // 생성자로 3개의 인자를 받음
    double getArea( ) { // 순수가상함수 구현함으로써 Oval은 온전한 클래스 // 이때, virtual 속성은 상속되므로, 키워드 생략 가능
        return width * height * 3.14; // 타원 표현
    }
};
class Rect :public Shape { // 추상클래스 Shape을 상속받는 파생클래스 Rect
public:
    Rect(string n, int w, int h) : Shape(n, w, h) {} // 생성자로 3개의 인자를 받음
    double getArea( ) { // 순수가상함수 구현함으로써 Rect은 온전한 클래스 // 이때, virtual 속성은 상속되므로, 키워드 생략 가능
        return width * height; // 사각형 표현
    }
};
class Triangular :public Shape { // 추상클래스 Shape을 상속받는 파생클래스 Triangular
public:
    Triangular(string n, int w, int h) : Shape(n, w, h) {} // 생성자로 3개의 인자를 받음
    double getArea( ) { // 순수가상함수 구현함으로써 Triangular은 온전한 클래스 // 이때, virtual 속성은 상속되므로, 키워드 생략 가능
        return width * height * 0.5; // 삼각형 표현
    }
};

int main( ) {
    Shape* p[3]; // Shape형 포인터배열
    p[0] = new Oval("빈대떡", 10, 20);
    p[1] = new Rect("찰떡", 30, 40);
    p[2] = new Triangular("토스트", 30, 40);

    for (int i=0; i<3; i++)
        cout << p[i]->getName( ) << " 넓이는 " << p[i]->getArea( ) << endl;

    for (int i=0; i<3; i++) delete p[i]; // 동적해제
    return 0;
}
```

```
C:₩WINDOWS₩system32₩cmd.exe
빈대떡 넓이는 628
찰떡 넓이는 1200
토스트 넓이는 600
Press any key to continue . . .
```