

C++ 프로그래밍 및 실습

바이오차 혼입 시멘트 (콘크리트) 실험 도우미

진척 보고서 #2

제출일자: 12/17

제출자명: 박경준

제출자학번: 192195

1. 프로젝트 목표

1) 배경 및 필요성

현재 조경학과에 속한 조경공학연구실 학부연구생으로 '바이오차 혼입 시멘트(콘크리트) 개발을 위한 실험'을 진행 중이다. 해당 실험은 탄소 절감을 목적으로 기존 (포장)콘크리트에 바이오차를 혼입하여 최적의 대체율을 찾는 실험이다. 이에 따른 재료들의 혼합을 하기 전 바이오차 대체율에 따른 각각의 재료들의 양 설정이 중요하다. 또한, 대체율에 따른 테스트 값들을 통해 분석을 진행할 수 있다.

2) 프로젝트 목표

프로젝트를 통해 직접 계산하던 방식에서 벗어나 준비과정부터 테스트 값들을 통한 유추를 자동화하여 실험 시간 단축을 목표로 한다.

3) 차별점

해당 프로젝트는 실험을 진행하면서 시간 소요가 있던 과정을 경험하고 필요에 의해서 만든 프로젝트로 최종적으로 단순히 값을 출력하는 것이 아닌 값의 범위를 통해서 대체율의 적합성 결과를 출력하여 차별성이 존재한다.

2. 기능 계획

1) 대체율에 따른 재료의 양 설정

- 바이오차 대체율(0~6%)에 따른 물, 시멘트, 잔골재의 양 설정

(1) 대체율에 따른 실험체(공시체)명 설정

- 바이오차는 시멘트를 대체, 즉 바이오차와 시멘트는 값이 유동적으로 변함

- 반면, 물과 잔골재의 양은 고정값

2) 시간대 별 Flow Test

- 0 ~ 40min, 각 10분 간격으로 Flow Test 진행하며 나온 값을 입력 후 평균값을 구하여 최종 Flow Test 값 파악

3) 압축강도

- 압축강도 실험을 통해서 얻은 값들을 입력하여 평균값을 얻은 뒤 최종. 압축강도 값을 파악

4) 적합성 파악

- 최적의 값을 설정한 뒤 얻어진 Flow Test, 압축강도 각각의 평균값을 통해 적합성 파악

5) 데이터들 최종 출력

- 대체율, 시험체 명, Flow Test 값, 압축강도, 적합성 모두 출력

3. 진척사항

1) 재료양 출력

(1) 대체율에 따른 출력

- 클래스 Setter, Getter 함수, 함수 정의, 입출력
 - 사용자가 대체율(%)를 입력할 경우 이에 따라 재료양이 설정되고, 출력
 - printSpecimenName() : 시험체(공시체)명 출력
 - printMaterials() : 재료의 양 출력
 - 출력함수들은 setter, getter 함수에서 계산된 값들을 출력
- 코드 스크린샷

(1) setter & getter

```
// setter : 멤버 변수 값 설정
void setFlowTestValue(double value) { flowTestValue = value; }
void setCompressiveStrength(double value) { compressiveStrength = value; }
void setSuitability(string value) { suitability = value; }
void setCement() {
    // Cement는 bioRate 비율에 따라 값이 줄어듦
    // 현재는 프로젝트를 진행 중이므로 시멘트의 기본값 1000(g)으로 설정
    cement = 1000 * (1.0 - bioRate);
}

// 시험체(공시체)명 설정
void setSpecimenName() {
    specimenName = "WB-C-" + to_string(static_cast<int>(bioRate * 100));
}

void setBioRate(int rate) {
    bioRate = static_cast<double>(rate) / 100;
}

// getter : 멤버 변수 값 반환
double getBioRate() { return 100 * bioRate; }
double getBioAsh() { return 1000 * bioRate; }
double getWater() { return water; }
double getCement() { return cement; }
double getFineAggregate() { return fineAggregate; }
double getFlowTestValue() { return flowTestValue; }
double getCompressiveStrength() { return compressiveStrength; }
string getSuitability() { return suitability; }
string getSpecimenName() { return specimenName; }
};
```

(2) 출력함수

```
// 대체물에 따른 실험재명 설정 함수
void printSpecimenName(ExperimentData& data) {
    cout << "실험체(공시체)명 : " << data.getSpecimenName() << endl;
}

// 바이오차 비율에 따른 시멘트량 및 재료량 출력
void printMaterials(ExperimentData& data) {
    cout << "바이오차 양 : " << data.getBioAsh() << "g" << endl;
    cout << "시멘트 양 : " << data.getCement() << "g" << endl;
    cout << "물 양 : " << data.getWater() << "g" << endl;
    cout << "잔골재 양 : " << data.getFineAggregate() << "g" << endl;
}
```

2) 시간대 별 Flow-Test

(1) Flow-Test 값 입력

- 클래스, 멤버변수, 변수 정의, 함수 정의, 리스트 getline(), stringstream, 입출력
 - 시간대 별 Flow-Test값을 입력받고 이를 시간에 맞게 flowTestValues에 저장
 - flowTestValues의 값들을 평균값 수식을 통해 setFlowTestValue 멤버 변수에 저장
- 코드 스크린샷

```
// 시간대 별 Flow-Test값 입력, 입력받은 값의 평균값 저장
void saveFlowTest(ExperimentData& data) {
    string input;
    double flowTestValues[4];
    // Flow-Test 시간분 고정값
    int times[4] = {10, 20, 30, 40};

    // for (int i = 0; i < 4; i++) {
    //     cout << times[i] << "분의 Flow-Test 값을 입력하세요 : ";
    //     cin >> flowTestValues[i];
    // }
    cout << "각 시간 별 Flow-Test 값을 공백으로 구분하여 cm 단위로 입력하세요 : ";
    // 해당 코드 작성하지 않을 경우 위 출력 문장 후 프로그램 종료
    cin.ignore();
    // 입력 한 줄로 받기
    getline(cin, input);

    // 입력 stringstream에 저장
    stringstream ss(input);

    // stringstream에서 Flow-Test 값 추출
    for (int i = 0; i < 4; i++) {
        ss >> flowTestValues[i];
    }

    double sum = 0;
    for (int i = 0; i < 4; i++) {
        sum += flowTestValues[i];
    }

    double average = sum / 4;
    // setFlowTestValue() 함수 호출하여 평균값 설정
    data.setFlowTestValue(average);
}
```

3) 압축강도

(1) 압축강도 값 입력

- 클래스, 멤버변수, 변수 정의, 함수 정의, 리스트 getline(), stringstream, 입출력
 - 압축강도 값을 입력받고 이를 시간에 맞게 compressiveStrengthValues에 저장
 - compressiveStrengthValues의 값들을 평균값 수식을 통해 setCompressiveStrength 멤버 변수에 저장
- 코드 스크린샷

```
// 실험체의 압축강도 출력, 입력받은 값의 평균값 저장
void saveCompressiveStrength(ExperimentData& data) {
    string input;
    double compressiveStrengthValues[4];

    cout << "실험체의 압축강도를 공백으로 구분하여 MPa 단위로 입력하세요 : ";
    // 해당 코드 작성하지 않을 경우 위 출력 문장 후 프로그램 종료
    cin.ignore();
    // 입력 한 줄로 받기
    getline(cin, input);

    // 입력 stringstream에 저장
    stringstream ss(input);

    // stringstream에서 압축강도 값 추출
    for (int i = 0; i < 4; i++) {
        ss >> compressiveStrengthValues[i];
    }

    double sum = 0;
    for (int i = 0; i < 4; i++) {
        sum += compressiveStrengthValues[i];
    }

    double average = sum / 4;
    // setCompressiveStrength() 함수 호출하여 평균값 설정
    data.setCompressiveStrength(average);
}
}
```

4) 데이터 최종 출력

(1) 데이터 표 형태 출력

- 클래스, 입출력, 반복문

- 출력사항 : 실험체 명, 바이오차 대체율, 시멘트양, 물양, 잔골재양, 플로우테스트 평균값, 압축강도 평균값

- 반복문을 이용하여 구분선 출력

- 코드 스크린샷

```
void printExperimentData(ExperimentData& data) {
    // 실험명 출력
    cout << "실험체명          바이오차 대체율  시멘트 양      물 양      잔골재 양      플로우 테스트 평균값  압축강도 평균값\n";

    // 구분선 출력
    for (int i = 0; i < 112; i++) cout << '-';
    cout << '\n';

    // 실험 데이터 출력
    cout << data.getSpecimenName() << string(25 - data.getSpecimenName().length(), ' ') // 실험체명
        << data.getBioRate() << string(20 - to_string(data.getBioRate()).length(), ' ') // 바이오차 대체율
        << data.getCement() << string(21 - to_string(data.getCement()).length(), ' ') // 시멘트 양
        << data.getWater() << string(20 - to_string(data.getWater()).length(), ' ') // 물 양
        << data.getFineAggregate() << string(26 - to_string(data.getFineAggregate()).length(), ' ') // 잔골재 양
        << data.getFlowTestValue() << string(25 - to_string(data.getFlowTestValue()).length(), ' ') // 플로우 테스트 평균값
        << data.getCompressiveStrength() << string(20 - to_string(data.getCompressiveStrength()).length(), ' ') << '\n'; // 압축강도 평균값
    // << data.getSuitability() << '\n'; // 적합성 판별
}
}
```

2) 테스트 결과

(1) 대체율에 따른 재료양 출력

- 바이오차 대체율이 나머지 바이오차와 시멘트양에 영향을 끼쳐 이에 따른 재료양을 출력

- 테스트에서는 3%로 설정

- 테스트 결과 스크린샷

```
바이오차 대체율(%)을 입력하시오 : 3
실험체(공시체)명 : WB-C-3
바이오차 양 : 30g
시멘트 양 : 970g
물 양 : 1g
잔골재 양 : 3g
각 시간 별 Flow-Test 값을 공백으로 구분하여 cm 단위로 입력하세요 :
```

(2) 플로우 테스트 및 압축강도 값 입력

- 테스트 결과 스크린샷

```
바이오차 대체율(%)을 입력하시오 : 3
실험체(공시체)명 : WB-C-3
바이오차 양 : 30g
시멘트 양 : 970g
물 양 : 1g
잔골재 양 : 3g
각 시간 별 Flow-Test 값을 공백으로 구분하여 cm 단위로 입력하세요 : 50 50 50 50
실험체의 압축강도를 공백으로 구분하여 MPa 단위로 입력하세요 : 23 23 23 23
```

(3) 데이터 최종 출력

- 테스트 결과 ‘-‘을 이용하여 상단 구분선 생성

- 적절한 공백 조절을 통해 값들 출력

- 테스트 결과 스크린샷

```
바이오차 대체율(%)을 입력하시오 : 3
실험체(공시체)명 : WB-C-3
바이오차 양 : 30g
시멘트 양 : 970g
물 양 : 1g
잔골재 양 : 3g
각 시간 별 Flow-Test 값을 공백으로 구분하여 cm 단위로 입력하세요 : 50 50 50 50
실험체의 압축강도를 공백으로 구분하여 MPa 단위로 입력하세요 : 23 23 23 23
실험체명      바이오차 대체율  시멘트 양    물 양      잔골재 양    플로우 테스트 평균값  압축강도 평균값
-----
WB-C-3        3                970         1          3            50                  18
```

4. 계획 대비 변경 사항

진척보고서#1 해당 내용

1) 최종 출력

- 이전 : 모든 값들을 문장으로만 출력
- 이후 : '-'를 이용한 Frame을 만들어서 출력
- 사유 : 표를 통해 출력하면 사용자가 값들을 파악하기 쉬움

진척보고서#2 해당 내용

1) 유효성 검사

- 이전 : 바이오차 대체율만 유효성 검사를 실행
- 이후 : Flow-Test 및 압축강도 유효성 검사 실행
- 사유 : Flow-Test와 압축강도 값도 사용자가 잘 못 입력했을 경우가 있으므로 유효성 검사를 실시

5. 프로젝트 일정

(진행한 작업과 진행 중인 작업 등을 표기)

업무		11/3	11/12	11/19	11/26	12/3	12/10	12/15	12/23
제안서 작성		완료							
기능1	세부 기능		완료						
기능2					완료				
기능3						완료			
기능4	세부 기능						-----	-----	----->
기능5								-----	----->