C++&Qt_day1 과제 1 보고서

로빛 20 기 인턴

정보융합학부

2025404008

김다은

과제 1 문제

정수들을 입력 받아 최대값, 최소값, 합계, 평균(소수 6 자리까지)를 계산하여 출력하는 프로그램

1. 전체 코드 알고리즘

1.1프로그램 전체 흐름

1. ArrayCalculator 객체를 생성한다
   - 모든 멤버 변수를 0 또는 NULL 로 초기화

2. 배열 크기를 입력받는다
   2-1. "몇 개의 원소를 할당하겠습니까?"라고 질문
   2-2. 사용자가 입력한 문자열을 받는다
   2-3. 입력이 1 이상의 정수가 맞는지 확인한다
   2-4. 틀리면 오류 메시지를 보여주고 다시 입력받는다
   2-5. 맞으면 그 크기만큼 메모리를 할당한다

3. 배열의 각 원소 값을 입력받는다
   3-1. 배열 크기만큼 반복한다
   3-2. "정수형 데이터 입력:"이라고 질문
   3-3. 사용자가 입력한 문자열을 받는다
   3-4. 입력이 정수(음수 포함)가 맞는지 확인한다
   3-5. 틀리면 오류 메시지를 보여주고 같은 자리를 다시 입력받는다
   3-6. 맞으면 배열에 저장하고 다음 자리로 넘어간다

4. 통계값들을 계산하고 결과를 보여준다
   4-1. 최대값, 최소값, 합계, 평균을 계산한다
   4-2. 계산된 결과를 화면에 출력한다

5. 프로그램이 끝날 때 할당된 메모리를 자동으로 해제한다

1.2 문자열을 숫자로 바꾸는 알고리즘

(양의 정수 변환)

1. 결과를 저장할 변수를 0 으로 초기화한다
2. 문자열의 첫 번째 글자부터 시작한다
3. 현재 글자를 숫자로 바꾼다 (예: '5' → 5)
   3-1. 현재 글자에서 '0'을 빼면 숫자가 된다
4. 지금까지의 결과에 10 을 곱하고 새 숫자를 더한다
   4-1. 예: 결과가 12 이고 새 숫자가 3 이면
   4-2. 12 × 10 + 3 = 123 이 된다
5. 다음 글자로 넘어가서 3-4 를 반복한다
6. 모든 글자를 처리하면 최종 숫자를 얻는다

Ex) "123"
result = 0
'1' 처리: result = 0 × 10 + 1 = 1
'2' 처리: result = 1 × 10 + 2 = 12
'3' 처리: result = 12 × 10 + 3 = 123
최종 결과: 123

(음수 포함 정수 변환)
1. 음수인지 확인한다 (첫 글자가 '-'인가?)
2. 음수이면 음수 표시를 기억하고 두 번째 글자부터 시작한다
3. 음수가 아니면 첫 번째 글자부터 시작한다
4. 위의 "양의 정수 변환 알고리즘"을 사용해서 절댓값을 구한다
5. 음수였다면 최종 결과에 마이너스를 붙인다

Ex) "-456"을 숫자로 바꾸기
첫 글자가 '-'이므로 음수로 표시
"456" 부분을 양의 정수로 변환: 456
음수 표시가 있으므로 최종 결과: -456

1.3 최대, 최소 값 찾기 알고리즘

최댓값
1. 배열의 첫 번째 값을 현재 최댓값으로 정한다
2. 배열의 두 번째 값부터 끝까지 하나씩 확인한다
3. 현재 확인하는 값이 지금까지의 최댓값보다 크면
   3-1. 그 값을 새로운 최댓값으로 업데이트한다
4. 모든 값을 확인하면 최종 최댓값을 얻는다

최솟값
1. 배열의 첫 번째 값을 현재 최솟값으로 정한다
2. 배열의 두 번째 값부터 끝까지 하나씩 확인한다
3. 현재 확인하는 값이 지금까지의 최솟값보다 작으면
   3-1. 그 값을 새로운 최솟값으로 업데이트한다
4. 모든 값을 확인하면 최종 최솟값을 얻는다

Ex) [10, -3, 25, 7, 15]에서 최댓값 찾기
max = 10 (첫 번째 값)
-3 확인: -3 < 10 이므로 max = 10 유지
25 확인: 25 > 10 이므로 max = 25 로 업데이트
7 확인: 7 < 25 이므로 max = 25 유지
15 확인: 15 < 25 이므로 max = 25 유지

최종 결과: 25

1.4 합계 계산 알고리즘

1. 합계를 저장할 변수를 0 으로 초기화한다
2. 배열의 첫 번째 값부터 끝까지 하나씩 확인한다
3. 각 값을 합계에 더한다
4. 모든 값을 더하면 전체 합계를 얻는다

Ex) [10, -3, 25, 7, 15]의 합계 계산
sum = 0
10 더하기: sum = 0 + 10 = 10
-3 더하기: sum = 10 + (-3) = 7
25 더하기: sum = 7 + 25 = 32
7 더하기: sum = 32 + 7 = 39
15 더하기: sum = 39 + 15 = 54
최종 결과: 54

1.5 평균 계산 알고리즘

1. 위에서 계산한 전체 합계를 가져온다
2. 배열의 크기(원소 개수)로 나눈다
3. 정확한 소수점 계산을 위해 소수 타입으로 변환한다

Ex) 합계가 54 이고 원소 개수가 5 개일 때
평균 = 54 ÷ 5 = 10.8

1.6 동적할당 메모리 알고리즘

(동적 메모리 할당 알고리즘)
1. 사용자가 원하는 배열 크기를 입력받는다
2. 그 크기만큼 정수를 저장할 수 있는 메모리 공간을 요청한다
3. 운영체제가 메모리를 할당해주면 그 주소를 포인터에 저장한다
4. 이제 그 메모리 공간을 배열처럼 사용할 수 있다

(메모리 해제)
1. 프로그램이 끝날 때 자동으로 실행된다 (소멸자)
2. 할당된 메모리가 있는지 확인한다 (포인터가 NULL 이 아닌가?)
3. 메모리가 할당되어 있으면 운영체제에게 반납한다
4. 포인터를 NULL 로 설정해서 실수로 다시 사용하는 것을 방지한다

2. 예외 처리 알고리즘

1.1 양의 정수 인지 확인 하는 함수 (isValidPInteger)

1) 입력된 문자열이 비어있는지 확인→ 비어있으면 거짓(false) 반환

2) 첫 번째 검증 단계 - 숫자 형식 검사

2-1) 문자열의 첫 번째 문자부터 시작

2-2) 현재 문자가 '0'~'9' 범위에 있는지 확인

2-3) 범위를 벗어나면 거짓(false) 반환

2-4) 문자열 끝까지 반복

3) 두 번째 검증 단계 - 의미있는 양수 검사

3-1) 문자열의 첫 번째 문자부터 다시 시작

3-2) 현재 문자가 '0'이 아닌지 확인

3-3) '0'이 아닌 문자를 발견하면 참(true) 반환

3-4) (3-2 에서 0 과 관련하여 예외를 처리 했음에도 불구하고 0 을 여러 개 입력 할 시 예외 처리를 하지 못하여)모든 문자가 '0'이면 거짓(false) 반환

- 빈 문자열을 입력하거나, 문자를 포함한 숫자, 0 으로만 이루어진 수, 음수를 포함한 수를 입력 받을 시 거짓을 반환하도록 하였습니다.

1.2 음의 정수인지 양의 정수인지 확인 하는 함수 (isValidInteger)

1. 입력된 문자열이 비어있는지 확인 → 비어있으면 거짓(false) 반환

2. 음수 기호 처리

2-1. 첫 번째 문자가 '-'인지 확인

2-2. '-'이면서 문자열 길이가 1 이면 거짓 반환 (단독 "-" 입력)

2-3. '-'가 있으면 검사 시작 위치를 두 번째 문자로 설정

2-4. '-'가 없으면 검사 시작 위치를 첫 번째 문자로 설정

3. 숫자 형식 검증

3-1. 설정된 시작 위치부터 문자열 끝까지 반복

3-2. 각 문자가 '0'~'9' 범위에 있는지 확인

3-3. 범위를 벗어나면 거짓(false) 반환

3-4. 모든 문자가 유효하면 참(true) 반환

- 빈문자열, -만 입력했을 때, 소수점을 포함한 수, 문자를 포함한 수, -기호가 잘못된 위치에 있을 때 와 같은 예외 사항을 입력했을 시 거짓을 반환하게 하였습니다.

1.3 위에서 검증한 내용중 거짓을 입력 받을 시 입력을 재시도 할 수 있도록 하는 알고리즘

(배열 크키 입력 시 예외처리)

1. 무한 루프 시작
2. 사용자에게 배열 크기 입력 요청
3. 한 줄 전체를 문자열로 입력받기 (getline 사용)
4. 양의 정수 검증 함수 호출
5. 검증 실패 시:
   5-1. 오류 메시지 출력
   5-2. 루프 시작점으로 돌아가기 (continue)
6. 검증 성공 시:
   6-1. 문자열을 정수로 변환
   6-2. 동적 메모리 할당
   6-3. 루프 종료 (break)

(배열 원소 입력 시 예외 처리)

1. 인덱스를 0 으로 초기화
2. 인덱스가 배열 크기보다 작은 동안 반복:
   2-1. 사용자에게 정수 데이터 입력 요청
   2-2. 한 줄 전체를 문자열로 입력받기
   2-3. 정수 검증 함수 호출
   2-4. 검증 실패 시:

- 오류 메시지 출력

- 같은 인덱스에 대해 다시 입력 받기 (continue)

2-5. 검증 성공 시:

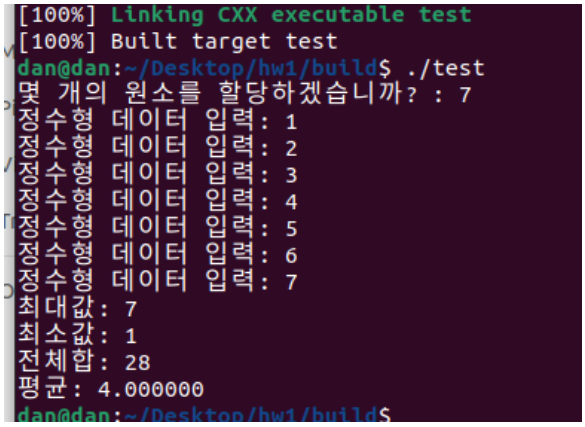
- 문자열을 정수로 변환 (음수 처리 포함)

- 배열에 저장

- 다음 인덱스로 진행 (i++)

결론:

과제 예시와 동일한 출력 값 확인



정수가 아닌 다른 값이 입력이 들어오면 예외처리 결과