

Advanced Programming

Assignment #2

Submission date : 2019. 5. 11

Affiliation : Department of Computer Software

Student ID : 2017203053

Student Name : 김 형 석

1. Check List

Case#1	Case#2	Case#3	Advanced	Wrapping in Class
✓	✓	✓	✓	✓

2. 각 Case 별 발견한 에러

(1) Case#1

- Token_stream 클래스의 멤버함수 Token get() 내부에서 Token(name,s)를 반환하는 부분이 있는데 이에 해당하는 정보가 Token 구조체에 정의 되었지 않아서 추가해주었습니다.

- Token_stream 클래스의 public 부분에서 bool 타입으로 선언된 full에 0값이 초기화 되어있어서 false로 변경했습니다.

- Token_stream 클래스의 멤버함수 Token get()의 정의 부분에서 함수명 부분에 에러가 있어서 Token Token_stream::get() 으로 변경해주었습니다.

- 함수 get()의 정의 부분에서 cin.unget()을 cin.putback(ch)로 변경해주었습니다. 이 부분은 변경하지 않아도 결과가 똑같이 나오는데 강의 자료에서 putback을 사용했기 때문에 변경했습니다.

- 함수 primary()의 정의 부분에서 case '(': 에 해당하는 부분에 반환 값이 없어서 반환 값을 넣어주었습니다.

- 함수 primary()의 정의 부분에서 case name: 에 해당하는 부분에 선언된 변수에 값을 저장하게 하는 조건문을 추가하였습니다.

- 함수 term()의 정의 부분에서 default: 부분에 ts.putback(t)를 추가하였습니다.

- 함수 expression()의 정의 부분에서 default: 부분에 ts.putback(t)를 추가하였습니다.

(2) Case#2

- 함수 primary()의 정의 부분에서 case '+': 에 해당하는 부분을 추가하였습니다.

(3) Case#3

- pi와 e에 해당하는 값을 전역변수로 선언 후 초기화 했습니다.
- 함수 `set_value(string s, double d)`의 정의 부분에서 `names` 벡터에 비교 대상이 되는 값을 넣어주는 코드를 추가 했습니다.
- `main` 함수에서 전역변수로 선언된 pi와 e의 값을 “pi”와 “e”의 이름으로 사용할 수 있도록 `set_value` 함수를 호출했습니다.

(4) Advanced

- 함수 `get()`의 정의 부분에서 `case '^':`를 추가하였습니다.
- 곱하기, 나누기 보다 우선순위가 높음을 나타내기 위해 새로운 함수 `pri_primary()`를 정의 했습니다. 내용은 기존의 `primary()` 함수 기능에 벡터에 '^'가 저장되면 전에 입력된 숫자에 '^' 다음에 입력된 숫자만큼 제곱을 시켜주는 기능을 추가시켰습니다.
- 함수 `term()`의 정의 부분에서 left에 `primary()` 값 대신 `pri_primary()`값을 저장하도록 바꿔주었습니다.

(5) Calculator 클래스로 통합

- '>', '='에 해당하는 상수 선언은 클래스에 넣고 이들을 제외한 상수 선언들은 전역변수로 선언했습니다.
- 사용되는 모든 함수(`calculate()`제외)와 구조체, 클래스들을 `Calculator` 클래스의 `private` 멤버함수로 선언 후 바깥에 정의했습니다.
- pi와 e값을 저장하는 `set_value` 호출은 `calculate()` 함수 내에서 호출시켰습니다.

3. 고찰

계산기 코드를 수정하면서 아직 잘 이해가 되지 않는 부분이 있었습니다. 생성자, 버퍼 등 잘 모르는 문법이 쓰여서 코드를 이해하는데 어려웠던거 같습니다. 이번 과제의 코드가 중간 고사 시험 범위였던 강의 자료에 있던 코드와 거의 비슷해서 몇 가지 에러는 금방 발견할 수 있었습니다.

이번 과제에서 가장 어려웠던 부분은 `exponential` 기능을 추가하는 `advanced` 단계였다고 생각합니다. 처음에는 `primary()`함수 내부에서 기능을 추가하려고 시도 했는데 잘 되지 않아서 새로운 함수를 구현했습니다.

하나의 클래스로 통합하는 단계는 수월하게 했습니다. 기능 자체는 똑같기 때문에 `advanced` 단계의 코드 내용을 사용했습니다.