프로그래밍 언어론

2017.03.15

20154073

강다현

Q. C, C++, Java를 언어 특징 별로 비교하고, 이를 이용하여 판독성, 작성력, 신뢰성의 3가지 평가 기준 관점에서 비교하라.

<언어 평가 테이블>

|  |  |  |  |
| --- | --- | --- | --- |
| 특징 / 언어 | C | C++ | Java |
| 단순성 | 3 | 1 | 2 |
| 직교성 | 1 | 2.5 | 4 |
| 데이터 타입 | 1 | 3 | 2 |
| 구문 설계 | 2 | 2 | 3 |
| 추상화 지원 | 1 | 3 | 2 |
| 표현력 | 3 | 3 | 3 |
| 타입 검사 | 2 | 2 | 3 |
| 예외처리 | 1 | 2 | 3 |
| 제한된 별칭 | 1 | 2 | 2 |

1. 단순성

프로그램의 규모가 클수록 프로그래머가 일부분만 배우게 되는데 이 때, 작성자와 판독자가 알고 있는 부분이 다를 경우 판독성에 영향을 미치므로, 프로그램의 크기에 따라 점수를 주었습니다.

일단 C언어는 다른 두 언어와 다르게 객체지향 개념이 빠져있으므로 크기가 작고, 그 다음으로 java인데, java는 포인터 개념이 빠져있으므로 C++보다 규모가 작다고 판단 각각 5점, 3점, 1점을 주었습니다.

2. 직교성

언어의 구성자들이 각각의 의미를 가진 채 결합하는 성질로 구성자간의 상호작용 또는 문맥의 사용이 예상 밖의 상황을 야기하지 않아야 합니다. 그러므로 여러 구성자들간의 상호작용에 따라 점수를 주었습니다.

먼저 C언어는 call by value를 바탕으로 인수를 전달하지만, 배열의 경우 직접 주소를 전달해야 하고, 함수에서 반환할 때도 마찬가지로 포인터(주소)로 반환합니다. 그 밖에 C언어에서는 직교성 측면에서 부족한 면을 보입니다. 다음으로 C++은 STL(표준 템플릿 라이브러리)가 컨테이너와 알고리즘이 분리되어 있어 C언어에 비해 좀 더 나은 직교성을 가진다고 판단했습니다. 다음으로 Java에서는 C언어나 C++이 갖는 직교적 특성과 메소드와 public 및 static을 적용할 때 서로 간섭하지 않는 것에서 세 언어 중 가장 나은 직교성을 갖는다고 생각해 각각 1점. 2.5점, 4점을 주었습니다.

3. 데이터 타입

프로그램 언어가 데이터 타입과 데이터 구조를 정의할 수 있는 충분한 장치를 제공할 때 판독성에 도움이 되므로 타입의 다양성을 비교하였습니다.

C언어에서는 bool타입이 없어 int형으로 플래그를 표현하지만, 그에 반해 C++과 java는 모두 bool타입이 있고, C와 C++에는 unsigned가 있어 수의 표현범위를 넓힐 수 있지만 그에 반해 java는 unsigned가 없어 데이터 타입의 다양성이 C++보다는 낮다고 생각하여 각각에 1점, 3점, 2점을 주었습니다.

4. 구문 설계

언어 요소들에 대한 형식도 마찬가지로 판독성에 영향을 미칩니다. 그 중 특수어가 판독성에 가장 영향을 많이 주기 때문에, 문장들의 그룹을 형식화하는 방법이 중요합니다. 예시로 특수어나 기호들로 구성된 서로 매칭되는 쌍을 사용해 문장 그룹을 형식화하는데 세 언어는 모두 중괄호를 사용해 복합문을 나타냅니다. 세 언어는 이 특수어에서 공통점을 갖고 있습니다. 이 때 끝을 특수어가 어떤 복합문의 끝을 나타내는지 알기 어려워집니다. 형식과 의미는 문장의 형태가 부분적으로 목적을 나타낼 수 있어서 언어 구조가 형태는 동일하나 문맥에 따라 다른 의미를 가질 때 위반됩니다. 예를 들면 static의 경우 문장이 쓰이는 문맥에 따라 의미가 달라집니다. C와 C++의 경우 static의 의미가 문맥에 따라 달라지므로 각각 2점, 2점, 3점을 주었습니다.

5. 추상화의 지원

추상화는 외부에서 세부사항들을 변경할 수 없도록 데이터 구조나 연산을 정의하여 사용할 수 있는 능력을 말합니다. C는 추상화를 지원하지 않는 반면 C++과 Java는 기본적으로 추상화를 지원하기 때문에 C에는 1점을, C++과 Java가 동시에 추상화를 지원하긴 하지만 C++에는 STL을 통해 완벽한 추상화를 지원하기 때문에 Java 보다 높은 3점을, Java에는 자연히 2점을 주었습니다.

6. 표현력

표현력은 일반적으로 프로그램을 짤 때 작은 프로그램으로 많은 양의 계산을 할 수 있도록 하는 연산자가 존재함을 나타냅니다. 따라서 계산에 용이한 연산자를 바탕으로 비교하였습니다.

예를 들면 a = 1+1이 a = 1++;로 표현되는 것이 표현력의 예라고 할 수 있는데, 이런 것들은 세 언어가 다들 비슷하게 갖고 있기 때문에 세 언어에 동일한 점수를 주었습니다.

7. 타입 검사

타입검사는 컴파일러에 의해서나, 프로그램 실행 중에 프로그램에 포함된 타입 오류를 검사하는 것입니다. 실행-시간 타입검사는 비용(프로그램 실행시간)이 비싸기 때문에 컴파일-시간 타입 검사가 더 효과적입니다. 최근 버전의 C, C++, java는 최근 타입 검사가 지원되지만 실행-시간 검사인지 아니면 컴파일-시간 타입검사인지의 차이만 존재합니다.

그 중 java가 컴파일-시간 타입 검사여서 다른 실행-시간 타입 검사보다 높은 점수를 받았습니다.

8. 예외 처리

예외 처리는 프로그램 실행 도중 오류를 발견했을 때 이를 처리하여 프로그램을 계속 실행하거나, 우아하게 종료하도록 하는 것입니다. C는 예외처리가 없고, C++과 java는 둘 다 예외처리가 존재하는데, C++은 모든 종류의 타입을 예외로 던질 수 있지만 java는 객체만을 예외로 던질 수 있습니다. 게다가 C++은 에러가 unchecked 상태밖에 없는데, java는 에러가 unchecked와 checked 상태 두 가지가 있어 예외처리를 좀 더 세분화 시킬 수 있습니다. 그래서 각각 1점, 2점, 3점으로 점수를 주었습니다.

9. 별칭

별칭은 동일한 기억 장소에 접근하는데 사용할 수 있는 두 개 이상의 다른 이름을 갖는 것으로, C와 C++는 포인터가 있어 새로운 포인터 변수에 주소값을 저장하여 여러 별칭을 가질 수 있습니다. java에서는 포인터가 없지만, 새로운 인스턴스(객체)에 기존의 객체를 대입하면, 새로운 인스턴스는 기존 객체와 같은 주소를 갖는 별칭이 됩니다. 언어의 신뢰성을 향상시키기 위해서는 별칭을 제약하는 경우가 있어 C언어에 1점, C++과 java에 2점을 주었습니다.

판독성의 평가기준은 단순성, 직교성, 데이터 타입,구문 설계가 있는데 이 기준들을 코드를 보았을 때 쉽게 이해할 수 있는지를 알려주는 척도로 java가 가장 높은 점수를 받았습니다. 다음 작성력은 판단력의 평가기준을 포함하여 추상화와 표현력이 있습니다. 작성력에서도 객체지향 언어가 좀 더 높은 점수를 받았는데 실제 C++과 java의 수업을 들으면서 코딩을 할 때 java를 코딩할 때 좀 더 편했던 것 같습니다. 마지막으로 신뢰성의 평가기준은 판독성과 작성력을 포함하여 타입검사, 예외처리, 별칭의 평가기준을 가지는데 아무래도 java가 포인터가 없다는 점에서 높은 신뢰성을 갖는다고 할 수 있습니다. 위의 점수를 각각 종합했을 때 C++과 java의 점수가 비슷하지만 그래도 더 우위에 있는 것은 java라고 할 수 있습니다.