

프로그래밍 교육자를 위한 프로그래밍 언어의 크기 비교

김건우*, 우균*

*부산대학교 컴퓨터공학과

e-mail: {gunwoo, woogyun}@pusan.ac.kr

A Comparison of Programming Language Size for Programming Teacher

Gunwoo Kim*, Gyun Woo*

*Dept. of Computer Science and Engineering, Pusan National University

요 약

사람들은 모국어가 아닌 다른 언어를 배울 때 대부분 단어, 문법 등에 많은 부담을 느낀다. 그리고 이런 부분이 언어를 습득하는데 흥미를 반감시키고 쉽게 포기하게 하기도 한다. 다른 언어와 마찬가지로 프로그래밍 언어 또한 문법이 복잡하면 배우는 사람이 쉽게 지치기 마련이다. 이 논문에서는 프로그래밍 언어의 어휘적, 구문적 요소를 이용해 몇 가지 프로그래밍 언어의 크기를 비교하고, 어떤 언어가 프로그래밍 언어를 처음 배우는 사람에게 적합한지 보여준다. 그리고 어떤 순서로 프로그래밍 언어를 배우는 것이 효과적인지 프로그래밍 언어의 크기 측면에서 분석한다.

1. 연구 배경

프로그래밍 언어(programming language)를 처음 배우는 사람에게 복잡한 구문(syntax)과 방대한 기능은 큰 진입 장벽이다. 그리고 초보자 대부분이 이런 진입 장벽 때문에 학습을 포기하는 것 또한 사실이다. 즉 프로그래밍 언어 초보자에게 중요한 것은 현재 자신이 가진 논리적인 문제 해결 능력이 아니라 흥미를 느끼고 즐겁게 배울 수 있도록 쉽고 단순한 프로그래밍 언어를 선택하는 것이다 [1].

이 논문에서는 프로그래밍 언어의 크기 비교를 통해 어떤 언어가 초심자에게 적합하고, 어떤 순서로 프로그래밍 언어를 배우는 것이 효율적인지 보여준다. 우선 비교 대상인 프로그래밍 언어를 간단히 소개하고, 프로그래밍 언어의 크기를 비교하기 위해 예약어(keyword), 연산자와 구두점(operator and punctuator), 생성규칙(production rule)과 같은 기준을 제시한다. 그리고 이 기준에 따라 대상 프로그래밍 언어(C, C++, Java, Python, Go)를 비교함으로써 프로그래밍을 가르치는 교육자에게 도움을 주고자 한다.

2. 대상 프로그래밍 언어

2.1 C

C 언어는 1970년대 초 AT&T Bell 연구소에서 Dennis Ritchie와 Ken Thompson에 의해 개발되었다. C 언어는 대표적인 명령형(imperative) 언어이며 다른 프로그래밍 언어에도 많은 영향을 끼쳤다. C는 범용 프로그래밍 언어

로 UNIX 시스템 개발을 위해 사용되었고, 현재까지도 많은 개발자가 사용하는 프로그래밍 언어다[2]. 또한, C 언어는 고급 언어(high-level language)이면서 저급 언어(low-level language)의 특징도 가지고 있다. 그리고 C 언어의 설계 원칙은 간결성이기 때문에[3] 언어의 크기는 작을 것으로 추정된다.

2.2 C++

C++는 1979년 Bell 연구소의 Bjarne Stroustrup이 개발했다. C++는 C 언어에 객체지향(object-oriented)적 특징을 추가해 만든 언어로 다양한 프로그래밍 패러다임(paradigm)을 지원한다[4]. 또 C++는 C 언어를 바탕으로 하고 있기 때문에 C가 지닌 특징 대부분을 포함하고 있다. 따라서 언어의 크기가 C보다 상당히 클 것으로 예상된다.

2.3 Java

Java는 1995년 Sun Microsystems의 James Gosling이 개발한 언어다. Java는 객체지향 프로그래밍 언어이며 코드를 한 번 작성하면 이식(porting) 없이 어떤 플랫폼(platform)에서도 동작 가능하다는 특징이 있다[5]. 그리고 Java는 C, C++와 더불어 지금까지도 많은 개발자가 사용하는 언어다[6]. Java는 설계 초기부터 안정성을 위해 C++의 일부 기능을 축소한 언어이므로 언어의 크기가 C++보다 작을 것으로 추정된다.

2.4 Python

Python은 1991년 Guido van Rossum이 개발한 언어로 범용의 고급 프로그래밍 언어이다. Python은 객체지향형, 명령형, 함수형(functional) 프로그래밍과 같은 다양한 프로그래밍 패러다임을 지원한다. 또한, Python은 동적 언어(dynamic language)로써 C로 표현 가능한 코드를 더 짧게 표현할 수 있다[7,8]. 따라서 Python은 언어의 크기가 C와 유사할 것으로 추정된다.

2.5 Go

Go 언어는 2007년 구글(Google)의 Robert Griesemer, Rob Pike, Ken Thompson이 만든 언어이다. 2009년 처음으로 외부에 공개되었으며 현재 구글의 다양한 서비스에 사용되고 있다. Go 언어는 동적 언어 Python처럼 프로그래밍하지만, 실제로는 컴파일 언어(compiled language)다. 그리고 직관적인 동시성(concurrency)을 제공해 쉽게 병렬(parallel) 프로그래밍을 할 수 있다는 점이 특징이다. 그러므로 프로그래밍 언어의 크기가 Python과 유사하거나 더 작을 것으로 예상된다.

3. 비교 기준

여기서는 프로그래밍 언어의 크기를 비교하는 데 필요한 몇 가지 기준을 제시한다. 이 기준은 프로그래밍 언어의 어휘적(lexical), 구문적(syntactic) 요소를 포함한다. 그 이유는 이런 요소들이 해당 프로그래밍 언어로 프로그램을 작성하기 위해 알아야 하는 기본적인 규칙이기 때문이다.

3.1 예약어의 개수

예약어는 프로그래밍 언어가 사용하려고 선점한 토큰(token)이기 때문에 식별자(identifier)로 사용할 수 없다. 그리고 예약어는 주로 기본 자료형(primitive data type), 제어 흐름(control flow) 등에 사용하려고 정의한 것으로 프로그래밍 언어가 제공하는 기능을 나타낸다. 따라서 예약어의 개수는 프로그래밍 언어의 크기를 정하는 데 꼭 필요한 요소이며 예약어가 많을수록 초심자가 언어를 빠르게 습득하는데 방해가 된다.

3.2 연산자의 개수

연산자는 예약어와 더불어 프로그램을 작성하기 위해 꼭 필요한 어휘적 요소의 하나이다. 하지만 많은 연산자를 가지고 있는 언어는 그만큼 많은 연산자를 알아야 하므로 초보자에게는 큰 부담이 될 수 있다. 그리고 프로그래밍 언어 대부분이 연산자와 구두점(또는 분리자)을 함께 취급하기 때문에 이 두 요소의 개수도 프로그래밍 언어의 크기를 규정하는 데에 중요한 부분이라 할 수 있다.

3.3 생성규칙의 개수

프로그래밍 언어의 문법(grammar) 역시 언어의 크기를

규정하기 위한 필수 요소다. 문법은 프로그래밍 언어의 구문적 요소로 모든 프로그램은 정해진 구문에 맞게 작성해야 한다. 따라서 문법의 생성규칙은 프로그래밍 언어에서 아주 중요하다. 그리고 생성규칙이 많다는 것은 알아야 할 문법이 많다는 것을 의미한다. 그러므로 생성규칙이 많은 언어는 언어의 크기가 크다고 할 수 있다.

4. 프로그래밍 언어의 크기 비교

2절에서 제시한 각 프로그래밍 언어에 대해 3절의 비교 기준을 적용하여 표 1로 표현하였다. 프로그래밍 언어 역시 계속해서 업데이트되기 때문에 이 논문에서는 가장 흔히 쓰이고 있는 버전을 기준으로 하였다. 따라서 C 언어는 1999년 표준, C++는 2003년 표준, Java는 7 버전, Python은 2.7.3, Go는 1.0.3 버전으로 비교하였다.

<표 1> 비교 기준에 따른 프로그래밍 언어의 크기 비교 결과. 비율은 C++를 기준으로 한 백분율을 의미한다.

프로그래밍 언어	예약어		연산자		생성규칙	
	개수	비율	개수	비율	개수	비율
C99	37	59	54	77	292	50
C++03	63	100	70	100	579	100
Java SE 7	50	83	46	66	444	77
Python 2.7.3	31	49	51	73	312	54
Go 1.0.3	25	40	47	67	289	50
최대 격차	38	60	24	34	290	50

표 1의 2열에 나타난 기준 3.1 예약어의 개수를 보면 C를 확장한 C++가 약 2배의 예약어를 가지고 있음을 알 수 있다[9,10]. 그리고 Java는 C, C++의 영향을 받았지만, 저급 언어의 특징을 몇 가지 제거하여 C++보다 조금 적은 수의 예약어를 가지고 있다[11]. 무엇보다 두드러진 점은 동적 언어인 Python과 컴파일 언어지만 동적 언어처럼 프로그래밍하도록 설계된 Go 언어가 가장 적은 수의 예약어를 가지고 있다는 점이다[12,14]. 그리고 예약어의 개수가 가장 많은 C++와 예약어의 개수가 가장 적은 Go 언어는 약 60%의 차이를 보였다.

기준 3.2 구두점을 포함한 연산자의 개수는 C++가 70개로 가장 많고, 나머지 언어는 약 50개로 비슷한 수의 연산자를 가지고 있다[9,10,11,12,14]. 컴퓨터 프로그램에 필요한 연산이 대부분 비슷하기 때문에 연산자, 구두점의 개수는 프로그래밍 언어마다 큰 차이가 없는 것으로 보인다. 따라서 가장 많은 연산자를 가진 C++와 가장 적은 연산자를 가진 Java의 차이는 34% 정도로 나타났다.

표 1의 4열에 나타난 기준 3.3 문법의 생성규칙 개수는 C++가 가장 많고 그다음이 Java, Python, C, Go의 순서다[9,10,11,13,14]. 기준 3.1과 유사하게 동적 언어를 지향하는 언어가 생성규칙 또한 적은 경향을 보이고 있다. 그리고 생성규칙이 가장 많은 C++와 생성규칙이 가장 적은 C,

Go 언어는 약 50%의 차이를 보인다.

결론적으로 어떤 언어가 가장 크기가 작은지 한눈에 알아보기 위해 각 기준에 점수를 부여한 결과를 표 2로 나타냈다. 기준마다 알맞은 점수를 부여하기 위해 5개의 대상 언어로 동일한 기능을 하는 스택(stack)을 구현하고, 구현한 프로그램에 사용된 예약어, 연산자, 생성규칙의 개수를 세어보았다. 그리고 각각이 사용된 비율은 예약어가 3.456%, 연산자가 14.09%, 생성규칙이 82.438%로 나타났다. 따라서 프로그래밍 언어의 크기를 나타내는 점수는 각각의 개수에 0.05, 0.15, 0.8을 곱하여 모두 합친 값으로 표현하였다.

<표 2> 프로그래밍 언어의 크기 점수. 총점이 작을수록 크기가 작은 프로그래밍 언어다.

프로그래밍 언어	예약어	연산자	생성규칙	총점
C99	1.85	8.1	233.6	243.55
C++03	3.15	10.5	463.2	476.85
Java SE 7	2.5	6.9	355.2	364.6
Python 2.7.3	1.55	7.65	249.6	258.8
Go 1.0.3	1.25	7.05	231.2	239.5

표 2의 2, 3, 4열은 표 1의 2, 3, 4열의 개수에 각각 0.05, 0.15, 0.8을 곱한 값이다. 그리고 5열의 총점은 2, 3, 4열의 값을 모두 합친 값으로 프로그래밍 언어의 크기를 의미한다. 따라서 Go 언어가 가장 크기가 작은 언어라고 할 수 있다. 그리고 그다음이 C, Python, Java, C++의 순서다.

5. 결론

C, C++를 아는 사람이라면 누구나 짐작할 수 있듯이 C++가 크기가 더 큰 언어라는 것을 수치로 확인할 수 있다. 그리고 C++처럼 객체지향적이지만 C, C++의 저급 언어 기능을 삭제한 Java가 C보다는 크고 C++보다 작다는 것도 알 수 있다. 또한, 스크립트 언어인 Python이 C와 유사한 크기의 프로그래밍 언어라는 것을 표 2를 통해 확인할 수 있다.

이 결과를 토대로 프로그래밍 언어를 처음 배우는 사람에게 쉽게 학습할 수 있는 언어를 추천한다면 Go나 Python을 추천할 수 있을 것이다. C 언어는 전통적이고 뒤이어 C++를 배우기 쉽다는 장점이 있지만, Go나 Python과 같은 언어가 예약어, 연산자나 구문에 얽매이지 않고 좀 더 빠르게 프로그래밍을 배울 수 있다는 부분에서 초보자에게 더 권유할 만하다.

실제로 대학교 대부분이 프로그래밍을 처음 배우는 학생들에게 C, C++, Java의 순서로 언어를 가르친다. 하지만 이런 순서의 교과과정은 프로그래밍 언어가 개발된 순서에 따른 것으로 초심자의 입장에서 전혀 생각하지 않은 것이라 할 수 있다. 따라서 프로그래밍 언어의 크기 비교를 통해 다른 순서의 교과과정을 생각해 볼 수 있다.

최근 MIT에서도 학생들의 첫 프로그래밍 강의에서 사용하는 언어를 Scheme에서 Python으로 바꾸었다. 그리고 캐나다의 Laval, 프랑스의 Bordeaux 1, Paris 13, 스페인의 Polytechnical 대학교에서도 첫 프로그래밍 언어로 Python이 사용되고 있다[16]. 표 2의 결과를 보면 Python이 C보다 조금 더 큰 언어로 나왔지만, 다양한 라이브러리(library)와 동적 언어라는 것이 Python이 첫 프로그래밍 언어로 인기를 끌고 있는 이유다.

그리고 대학생 대부분이 C++를 배운 후 Java를 배우지만, 프로그래밍 언어의 크기를 생각해 보면 C++보다는 Java를 먼저 배우는 것이 언어의 어휘적, 구문적 요소에 구애받지 않고 객체지향 개념을 배우는데 더 도움이 된다. 결론적으로 프로그래밍 언어의 크기와 몇 가지 추가적인 요소를 고려해보면 Python, C, Java, C++의 순서로 프로그래밍 언어를 배우는 것이 초보자가 언어에 부담을 갖지 않고 프로그래밍을 배울 수 있는 좋은 선택이 될 것이다.

끝으로 최근 구글이 늘어나는 코어(core) 수와 네트워크 시스템, 웹 등 계속해서 거대해져 가는 현재의 컴퓨팅 환경에 적합한 프로그래밍 언어가 없다고 주장하며 개발한 Go 언어도 크기 측면에서 초보자에게 아주 적합한 언어다. 또 구글에서 Go 언어에 대해 컴파일 언어지만, 동적 언어처럼 쉽게 프로그래밍할 수 있다고 말한 것도 언어의 크기 측면에서 Python과 비교해보면 타당하다고 할 수 있다[15]. 따라서 Go 언어도 Python과 더불어 프로그래밍 초보자가 배우기에 적합한 언어다.

참고문헌

- [1] 이찬도, “처음 배우는 프로그래밍 언어로 무엇을 어떻게 가르칠까?”, 정보과학회지, 제28권 제4호, pp 24-27, 2010.
- [2] Brian W. Kernighan, Dennis M. Ritchie, *The C Programming Language*, 2nd Edition, Prentice Hall.
- [3] Terrence W. Pratt, Marvin V. Zelkowitz, *Programming Languages : Design and Implementation*, 4th Edition, Prentice Hall.
- [4] Bjarne Stroustrup, *The C++ Programming Language*, 3rd Edition, Addison-Wesley.
- [5] Ken Arnold, James Gosling, David Holmes, *The Java Programming Language*, 4th Edition, Addison-Wesley.
- [6] TIOBE Programming Community Index for March 2013, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.
- [7] Mark Summerfield, *Rapid GUI Programming with Python and Qt*, 1st Edition, Prentice Hall.
- [8] Steve McConnell, *Code Complete*, 2nd Edition, Microsoft Press.
- [9] ISO/IEC JTC1/SC22/WG14, “Programming languages — C,” ISO/IEC 9899:TC2, 2005.

- [10] ISO/IEC JTC1/SC22/WG21, “Programming languages — C++,” ISO/IEC 14882:2003(E), 2003.
- [11] James Gosling, Bill Joy, Gilad Bracha, Alex Buckley, “The Java Language Specification Java SE 7 Edition,” 2013.
- [12] Lexical Analysis — Python v2.7.3 documentation, http://docs.python.org/2/reference/lexical_analysis.html.
- [13] Full Grammar specification — Python v2.7.3 documentation, <http://docs.python.org/2/reference/grammar.html>.
- [14] The Go Programming Language Specification, <http://golang.org/ref/spec>, 2012.
- [15] The Go Programming Language Documentation, <http://golang.org/doc/>.
- [16] Schools using Python, <http://wiki.python.org/moin/SchoolsUsingPython>