

Programming Language

Assignment #2

- Copy 시 소스 제공자와 카피 당사자 공히 해당 과제 전체 0점 처리.
- 반드시 자기 자신의 생각과 글로 작성할 것. 다른 곳에서 자료 인용 시 출처 반드시 기재 (미기재 시 50% 감점)
- 한글/영문 상관 없음.

1. Write a short history of the Fortran 0, Fortran I, Fortran II and Fortran IV systems

Fortran 0

1954년 11월에 IBM의 John Backus와 그의 그룹은 "The IBM Mathematical FORMula TRANslating System: Fortran" (IBM, 1954, Fortran 0)의 제목을 갖는 보고서를 작성하였다. 이 문서와 Fortran 개발 초기의 문서에는 Fortran이 직접 코딩된 프로그램의 효율성과 해석적인 의사코드 시스템이 갖는 프로그램의 용이성을 모두 제공할 것이라는 내용과 함께, J. Halcombe Laning이 개발한 컴퓨터에 방정식을 더 쉽게 입력할 수 있는 아이디어에 대한 개념이 포함되었고, Fortran에 대한 전반적인 설계가 포함되었다. (2글자로 제한된 변수 이름 등)

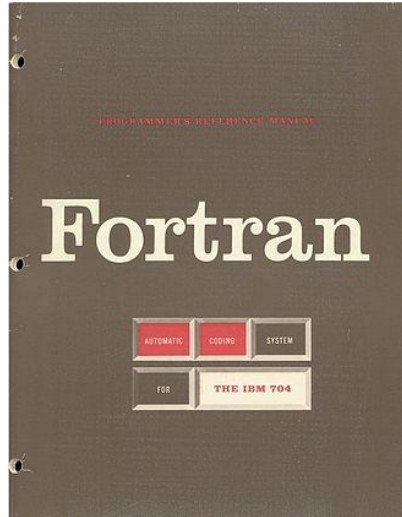
Fortran 0은 1957년 4월 Fortran의 실제 구현체(컴파일러)인 Fortran 1이 나오기 전까지 계속 수정되었다.

Fortran I

1954년 11월 Fortran의 사양 초안이 완성된 이후, 1956년 10월 Fortran에 대한 첫번째 매뉴얼이, 1957년 4월에 첫 번째 Fortran 컴파일러가 발표되었다.

(Fortran I)

Fortran I는 입출력 형식화와 최대 여섯 글자로 제한된 변수 이름, 사용자-정의 서브루틴(개별 컴파일 불가), If 선택문, Do 루프문을 포함했다.



<1956년 10월에 출판된 첫 번째 Fortran “프로그래머의 참고 매뉴얼” (IBM, 1956)>

Fortran II

Fortran II 컴파일러는 1958년 봄에 배포되었다. Fortran II는 Fortran I 컴파일 시스템에 있던 많은 버그들을 수정하였고, 서브루틴 독립 컴파일을 지원하여 컴파일 과정을 효율적으로 개선하였고 이를 통해 더 큰 프로그램을 개발하는 것이 가능하게 되었다. 또한 SUBROUTINE, FUNCTION, END, CALL, RETURN, COMMON의 6개 구문이 추가되었으며, 이후 몇 년 동안의 개선으로 DOUBLE PRECISION과 COMPLEX 데이터 유형에 대한 지원이 Fortran II에 추가되었다.

Fortran IV

1961년부터 개발이 시작된 Fortran IV는 1962년 IBM 7030 컴퓨터 용으로 출시되었다. (이후 IBM 7090, IBM 9094, IBM 1401용으로 추가적으로 출시됨) Fortran IV는 Fortran II에 비해 여러 면에서 향상이 있었다. 변수에 대한 명시적 타입 선언, 논리 If문, 논리 Bool 표현식, 부프로그램 매개변수 전달 등의 기능이 추가되었으며 Fortran II의 기계 종속적인 기능(입력 테이프 읽기)이 삭제되었다. 이 시기의 Fortran IV는 중요한 교육 도구가 되기 시작했으며, 당시에 가장 널리 사용된 프로그래밍 언어들 중 하나가 되었다.

2. Describe in detail the two most important reasons, in your opinion, why Speedcoding did not become a very widely used language.

첫번째로는 속도이다. Speedcoding은 기계 코드로 프로그래밍하는 어려움을 타파하고자, 시스템 리소스를 희생시키더라도 사용 편의점에 중점을 두어 개발된 언어로 다양한 기능을 포함하여 사용자의 수고를 덜어주었지만, 인터프리터로 구현된 점을 포함하여 이러한 개발 목표를 달성하기 위해서 속도 측면에서 너무 느렸다. 일반적인 IBM 701의 컴퓨팅 속도가 덧셈 연산은 0.06MS, 곱셈 연산은

0.456MS 정도의 수준이었는데, Speedcoding은 덧셈 4.2MS, 곱셈 3.5MS의 수준으로 일반적인 기계 코드의 실행 시간의 10~20배 정도 느렸다. 이는 당시 1950년대의 컴퓨팅 성능이 현대에 비해 매우 제한되었던 점을 고려하면, 널리 사용되는데 큰 장애물이 되었던 요인으로 생각된다.

두번째로는 메모리이다. 제곱근, 사인, 아크 탄젠트, 지수, 로그 연산과 부동-소수점 산술 연산을 위한 의사 명령어, 조건 분기, 무조건 분기, 입출력 변환 등의 여러 기능들을 가상 기계 구조로 구현해낸 Speedcoding 인터프리터는 상당히 무거웠으며, 인터프리터가 적재된 후 남아있는 사용 가능한 메모리는 700워드밖에 남지 않았다고 한다. (IBM 701 기준, Speedcoding 인터프리터는 IBM 701에서 사용 가능한 메모리의 약 30%인 310 워드를 사용함) 이는 Speedcoding으로 다양한 프로그램이 만들어지지 못하게 하는 큰 제약 요소로 작용했을 것이다.

이러한 느린 속도, 큰 메모리 사용 등의 문제로 Speedcoding은 널리 사용된 언어가 되지 못한 것이라고 생각한다.

3. Why, in your opinion, did Fortran allow names that begin with I, J, K, L, M and N as implicitly integer type?

당시 과학자와 공학자들이 보통 i, j, k 등의 문자를 변수 아래 첨자로 사용하였기 때문이다. 이런 묵시적 선언은 컴파일 프로세스가 철자나 프로그래머 오류를 탐지하는 것을 방해하기 때문에 신뢰성에 유해할 수 있으나, 프로그래머에게 약간의 편리성을 제공한다. (원래는 "integer :: i, j, k"와 같이 선언 후 사용해야 하는 것을, Integer 타입의 변수로 묵시적으로 선언해주기 때문에, 선언 구문이 필요하지 않게 된다)

4. What is the primary reason why C became more widely used than Fortran?

과학, 상업 계산을 주요한 목적으로 개발되고 사용된 한정적인 Fortran의 사용처와 달리 C언어는 1972년에 유닉스 운영체제를 개발하기 위한 목적으로 개발되어 이후 유닉스 기반 운영체제가 성공적으로 PC 시장에서 주요한 운영체제로 안착하였고, 범용적인 절차적 언어이자, 효율적으로 설계된 C 언어의 장점과 함께 이와 관련된 응용 프로그램을 개발하는 수요가 늘면서 폭발적으로 C 언어에 대한 사용이 증가하였다. 또한 당시 시장 상황에서 컴파일러는 컴퓨터 구매의 번들로 제공되거나, 개별적인 구매가 필요했는데, C 언어는 1980년대 이식성이 뛰어나고,

무료에다가 고품질인 GNU Compiler collection(GCC)가 공개되면서 조금 더 폭 넓은 접근성과 확장성을 가지게 되었다.

이러한 이유들로 인해 포트란에 비해 C 언어가 더 널리 사용되게 되었다고 생각한다.

5. What is your opinion of the argument that languages that are too complex are too dangerous to use, and we should therefore keep all languages small and simple?

어느정도 동의하는 바이다. 이는 현대 스크립트(Python, Ruby, Perl) 언어들의 디자인 방향과 어느정도 일맥상통한다고 생각한다. 큰 복잡성을 가지는 프로그래밍 언어는 생산성이 떨어지고, 규모가 큰 프로그램(성능과 관계없이)을 짜기 위해서 훨씬 많은 노력이 소요될 것이다. 따라서 개발과 생산성 측면에서 프로그래밍 언어는 규모가 작고 단순한 편이 효율적이라고 생각한다.

다만 현대에는 다양한 목적의 언어들이 있고, 이런 언어들 중에는 복잡성이 언어 구현 또는 목적에 부합하거나 필요악적으로 존재하게 되는 경우가 있다. 가령 C++은 언어가 다소 복잡해질 수 있더라도, C 언어와 이전 버전의 C++에 대한 하위 호환을 지원함과 동시에 현대적인 프로그래밍 개념이나 기능들을 추가하는 개발 철학을 가지고 있다. 그럼에도 C++이 (이러한 철학을 좋아하지 않는 사람도 분명 있겠지만) 아직까지도 널리 사용되고 있는 점을 감안하면, 가능한 한 단순하게 유지하는 것은 좋으나, 그 방향이 언어 목표와 부합하고 복잡성을 가지지 않음으로써 잃는 것이 없는 상황일 때에 한해서 해당 주장이 옳다고 생각한다.

6. Compare the features of Fortran and C and BASIC. Read some of the document, which are available on the Internet, and write an analysis of their features.

C 언어는 응용 프로그램, 시스템, 저수준 제어 등의 범용 목적으로 1973년 개발된 프로그래밍 언어이다. 명령형, 절차 지향적으로 설계되었으며 컴파일 언어이다. 다양한 하드웨어로의 이식성이 뛰어나고, 저급 언어의 특징을 가지고 있어 저수준 제어에도 많이 사용된다. (Inline assembly 지원 등)

Fortran(IBM Mathematical **FOR**mula **TRAN**slating System)은 응용 프로그램, 수치 계산, 과학, 상업 등의 목적으로 1954년 개발된 프로그래밍 언어이다. 배열 기반, 벡터화, 병행, 분산/공유 메모리 병렬 처리, 절차 지향 등의 특징을 가진 컴파일

언어이다.

BASIC(Beginner's All-purpose Symbolic Instruction Code)은 1964년 교육용으로 개발된 프로그래밍 언어이다. 절차 지향 언어이자 교육용으로 개발되어 쉬운 문법을 가졌으며, 컴파일러와 인터프리터 방식 모두로 구현되었다.

7. Give a brief general description of the Java servlet.

Servlet은 웹 서버 시스템상에 상주하여 실행되는 Java 클래스의 사례이다. Servlet의 실행은 Web 브라우저에 의해서 출력되고 있는 마크업 문서에 의해서 요청된다. 이러한 Servlet들은 Servlet Container라고 불리는 웹 서버 프로세스 (Apache Tomcat 등)에서 제어된다.

```
public class JavaServlet extends HttpServlet {  
    protected void doGet(HttpServletRequest request, HttpServletResponse response) throws Exception {  
        response.getWriter().append("Hello World!");  
    }  
  
    protected void doPost(HttpServletRequest request, HttpServletResponse response) throws Exception {  
        doGet(request, response);  
    }  
}
```

<Java Servlet 클래스의 구현 예시>

8. Reference

IBM 704와 Fortran - 프로그래밍 언어론 제 10판 (Robert W. Sebesta) P46~48

[IBM 701 Feeds and speeds \(IBM Archives\)](#)

[Fortran \(Wikipedia\)](#)

The IBM 701 Speedcoding System (Backus, J. W.) P4-6

[GNU_Compiler Collection](#)

[C \(programming language, Wikipedia\)](#)

[BASIC \(Wikipedia\)](#)

[Comparison of programming languages \(Wikipedia\)](#)

정적 타입 바인딩 - 프로그래밍 언어론 제 10판 (Robert W. Sebesta) P227~228

[Java 7 Servlet Technology \(Oracle document\)](#)

<The End of the Assignment>