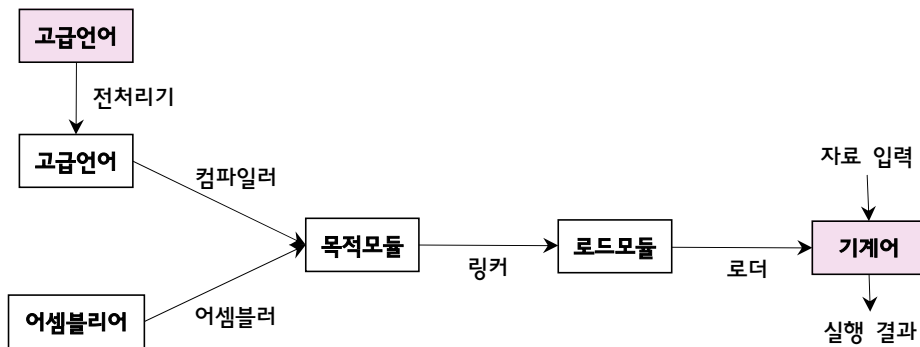


2. 언어 구현 기법

1. 번역 기법(translation)

번역 기법은 고급언어로 작성된 프로그램을 특정 컴퓨터의 기계어로 번역하여 동등한 의미의 기계어 프로그램을 만들어 실행시키는 방법으로 이러한 언어를 '컴파일러 언어'라고도 한다.

// 번역기



① 전처리기(preprocessor)

- 전처리기는 고급언어로 작성된 프로그램을 대응되는 **다른 고급언어**로 번역한다.
- 전처리기가 다루는 것은 원시언어와 목적언어가 모두 **고급언어** 형태이다.

② 컴파일러(compiler)

- 고급언어를 **준기계어** 형태인 **목적코드(목적모듈)**로 번역한다.
- 보통 '목적코드나 어셈블리어'가 준기계어 형태에 해당한다.

③ 어셈블러(assembler)

- 특별히, **어셈블리어**를 목적코드로 번역하는 것을 어셈블러라 한다.
- 기본적으로, 어셈블리어는 기계어와 1:1 대응한다.

④ 링커(linker) 또는 링케이지 에디터(linkage editor)

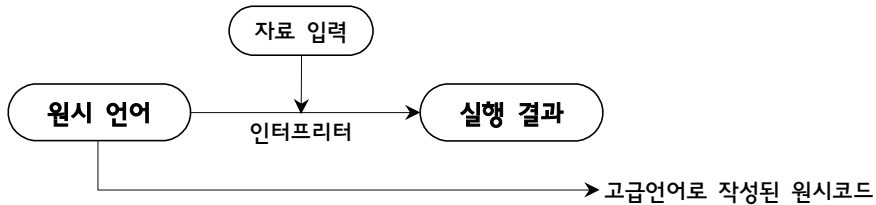
- 링커는 목적코드와 라이브러리 프로그램을 결합하여 **로드모듈**로 번역한다.
- 링커는 어느 정도 실행 가능한 형태인 로드모듈로 번역한다.

⑤ 로더(loader)

- 로드모듈이 실제 실행되려면 주기억장치에 적재되어 '프로세스'가 되어야 한다.
- 로더는 로드모듈을 실제 실행되는 **기계어**로 번역해서 주기억장치에 적재한다.
- 목적모듈이 바로 실행파일이 되는 것이 아니고, 링커와 로더가 실행파일을 생성한다.

2. 인터프리터(interpreter; 해석기)

인터프리터는 고급언어로 작성된 프로그램을 읽어서 기계어 수행과 같은 알고리즘으로 원시코드를 해독하여 수행시키므로써 고급언어를 시뮬레이션하게 되는 것이다.



고급언어의 원시코드를 **중간코드**로 번역해서 **바로 실행**시키는 언어를 **인터프리터 언어**라 한다.

• 예 : Python, VB, Lisp, Snobol, APL, Scheme, JavaScript, PHP, Ruby, Bash script

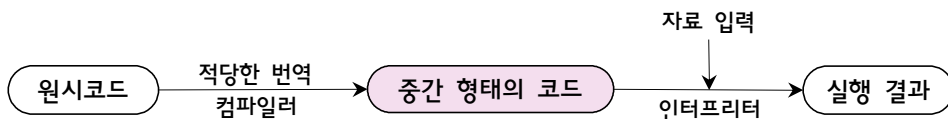
Tip	현재, 빠르게 변화하는 비즈니스 상황에서 짭사게 프로그램을 개발하기 위해 스크립트 언어가 많이 사용된다. 스크립트 언어는 거의 인터프리터 언어로 실행 속도는 늦다. 하지만, 컴퓨터의 성능이 빠르게 높아지고 있어서 느린 실행 속도는 어느 정도 해결되고 있다.
-----	---

3. 하이브리드 구현 기법

컴파일러와 인터프리터 기법을 절충한 형태로 ‘컴파일러-인터프리터’ 또는 ‘하이브리드 구현’ 기법이라고 한다.

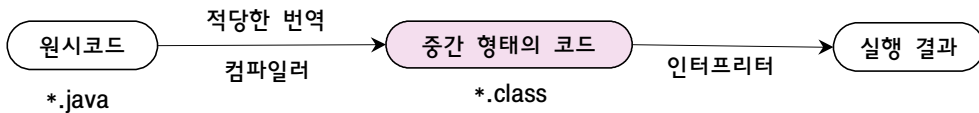
// 하이브리드 구현

고급언어로 작성된 원시코드를 입력자료로 읽어서 좀 더 실행시키기 쉬운 형태로 먼저 번역한다. 그리고 번역된 코드를 다시 해석하여 프로그램을 실행시킨다.



// JAVA

JAVA는 하이브리드 구현 기법을 사용하는 언어이다.



- **중간코드** : 중간 형태의 코드가 곧 "바이트코드(bytecode)"이다.
- **바이트코드**는 opcode가 한 바이트로 구성되어 있어서 붙여진 이름이다.
- **바이트코드**는 하드웨어와 독립적인 중간코드로 인터프리터에 의해 해석되어 실행된다.
- 한번 번역된 바이트코드는 기종에 상관없이 인터프리터만 있으면 실행 가능하다.
- 이러한 인터프리터(해석기)가 설치된 컴퓨터를 JVM이라 한다.
- 자바는 개발 환경과 실행 환경이 분리된 형태이다. WORA(write once, run anywhere)

Tip 하이브리드 기법은 번역 기법과 인터프리터 기법의 장점을 이용한다.
하이브리드 기법은 실행의 효율성, 융통성, 호환성 등을 높일 수 있다.

// 번역 기법과 인터프리터 기법의 비교

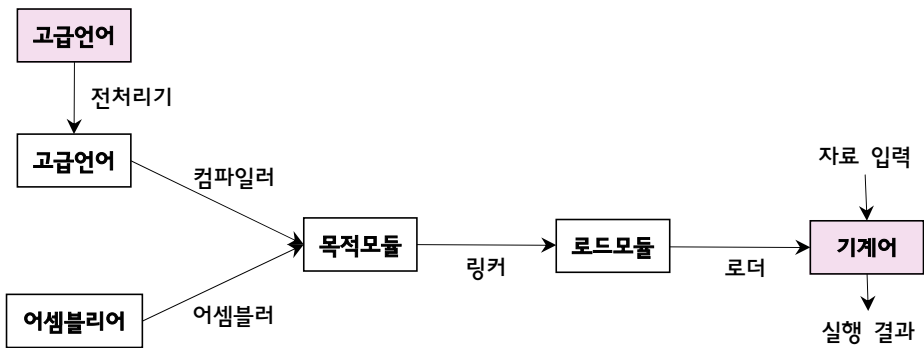
컴파일러	인터프리터
<ul style="list-style-type: none"> • 변수에 대한 타입은 번역시간에 검사된다. • 정적타입 검사이다. • 프로그램을 안정적인 구현할 수 있다. • 부프로그램 호출이나 반복처리 되는 프로그램에 적용하면 실행 효율이 좋다. • 번역된 프로그램이 저장될 큰 공간이 필요 • 심볼테이블은 번역 과정에서 데이터베이스로 사용된다. • 반복 운영 시스템에 적합 	<ul style="list-style-type: none"> • 실행 중에 자료의 동적 변화를 구현하므로 유연성이 좋다.(동적타입) • 실행시간에 오류 메시지를 참조할 수 있어서 디버깅하기가 쉽다. • 실행 중에 복잡한 고급 문장을 해석하므로 실행속도는 느리나 구현하기는 쉽다. • 실행 중에 원시코드와 심볼테이블이 기억공간에 존재해야 한다. • 개발 시스템, 교육 시스템에 적합

기출문제 분석

1. 재배치 가능한 형태의 기계어로 된 오브젝트 코드나 라이브러리 등을 입력받아 이를 묶어 실행 가능한 로드 모듈로 만드는 번역기는? [2019년 국가 9급]

- ① 링커(linker)
- ② 어셈블리(assembly)
- ③ 컴파일러(compiler)
- ④ 프리프로세서(preprocessor)

☞ 언어 구현 기법 - 번역기



① 전처리기(preprocessor)

- 전처리기는 고급언어로 작성된 프로그램을 대응되는 **다른 고급언어**로 번역한다.
- 전처리기가 다루는 것은 원시언어와 목적언어가 모두 **고급언어** 형태이다.

② 컴파일러(compiler)

- 고급언어를 **준기계어** 형태인 **목적코드(목적모듈)**로 번역한다.
- 보통 '목적코드나 어셈블리어'가 준기계어 형태에 해당한다.

③ 어셈블러(assembly)

- 특별히, **어셈블리어**를 목적코드로 번역하는 것을 어셈블러라 한다.
- 기본적으로, 어셈블리어는 기계어와 1:1 대응한다.

④ 링커(linker) 또는 링커리지 에디터(linkage editor)

- 링커는 목적코드와 라이브러리 프로그램을 결합하여 **로드모듈**로 번역한다.
- 링커는 어느 정도 실행 가능한 형태인 로드모듈로 번역한다.

⑤ 로더(loader)

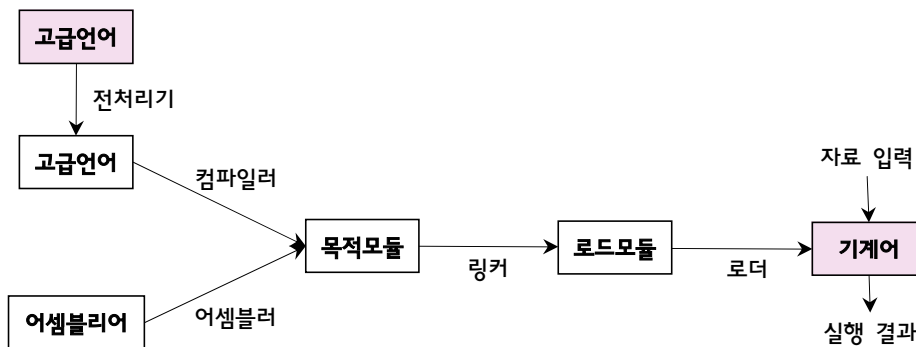
- 로드모듈이 실제 실행되려면 주기억장치에 적재되어 '프로세스'가 되어야 한다.
- 로더는 로드모듈을 실제 실행되는 **기계어**로 번역해서 주기억장치에 적재한다.
- 목적모듈이 바로 실행파일이 되는 것이 아니고, 링커와 로더가 실행파일을 생성한다.

정답 : ①

2. 다음은 컴퓨터 언어처리에 관련된 시스템 S/W의 기능을 설명한 것이다. 옳지 않은 것은?
 [2017년 서울 9급]

- ① 컴파일러 : 고급언어를 이진목적모듈로 변환기능
- ② 어셈블러 : 객체지향언어를 이진목적모듈로 변환기능
- ③ 링커 : 여러 목적모듈을 통합하여 실행 가능한 하나의 모듈로 변환기능
- ④ 로더 : 실행 가능한 모듈을 주기억장치에 탑재기능

☞ 언어 구현 기법 - 번역기



① 전처리기(preprocessor)

- 전처리기는 고급언어로 작성된 프로그램을 대응되는 **다른 고급언어**로 번역한다.
- 전처리기가 다루는 것은 원시언어와 목적언어가 모두 **고급언어 형태**이다.

② 컴파일러(compiler)

- 고급언어를 **준기계어** 형태인 **목적코드(목적모듈)**로 번역한다.
- 보통 ‘목적코드나 어셈블리어’가 준기계어 형태에 해당한다.

③ 어셈블러(assembly)

- 특별히, **어셈블리어**를 목적코드로 번역하는 것을 어셈블러라 한다.
- 기본적으로, 어셈블리어는 기계어와 1:1 대응한다.

④ 링커(linker) 또는 링케이지 에디터(linkage editor)

- 링커는 목적코드와 라이브러리 프로그램을 결합하여 **로드모듈**로 번역한다.
- 링커는 어느 정도 실행 가능한 형태인 로드모듈로 번역한다.

⑤ 로더(loader)

- 로드모듈이 실제 실행되려면 주기억장치에 적재되어 ‘프로세스’가 되어야 한다.
- 로더는 로드모듈을 실제 실행되는 **기계어**로 번역해서 주기억장치에 적재한다.
- 목적모듈이 바로 실행파일이 되는 것이 아니고, 링커와 로더가 실행파일을 생성한다.

3. 프로그래밍 언어 번역 프로그램에 대한 설명으로 옳지 않은 것은? [2022년 계리]

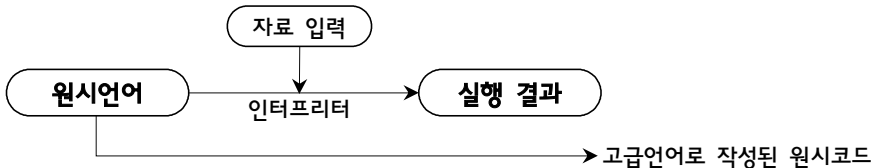
- ① 인터프리터(interpreter)는 고급언어로 작성된 원시프로그램을 함수 단위로 읽어 기계어로 번역하는 프로그램이다.
- ② 컴파일러(compiler)는 고급언어로 작성된 원시프로그램을 기계어나 어셈블리어로 된 목적 프로그램으로 바꾸는 프로그램이다.
- ③ 어셈블러(assembler)는 어셈블리어로 작성된 원시프로그램을 기계어로 번역하는 프로그램이다.
- ④ 프리프로세서(preprocessor)는 컴파일러가 컴파일을 수행하기 전에 원시프로그램의 내용을 변경하는 것이다.

☞ 프로그래밍 언어 번역 프로그램

· 인터프리터(interpreter)는 고급언어로 작성된 원시프로그램을 함수 단위로 읽어 기계어로 번역하는 프로그램이다.(x) → 함수 단위가 아니고, 줄 단위로 읽어 기계어로 해독하여 바로 수행한다.

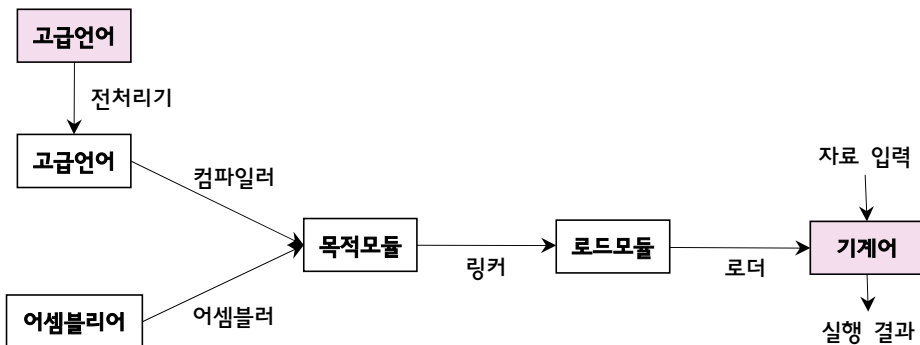
// 인터프리터 기법

인터프리터는 고급언어로 작성된 프로그램을 읽어서 기계어 수행과 같은 알고리즘으로 원시코드를 해독하여 수행시키는 방식으로 고급언어를 시뮬레이션하게 되는 것이다.



// 번역 기법

번역기는 원시언어로 작성된 프로그램을 읽어 들여 기능이 동등한 목적언어로 된 프로그램을 출력하는 언어 처리기이다.



4. 프로그램 구현 기법은 컴파일러를 이용한 기법, 인터프리터를 이용한 기법, 하이브리드(hybrid) 기법으로 구분된다. 이에 대한 설명으로 옳지 않은 것은? [2017년 지방 9급]

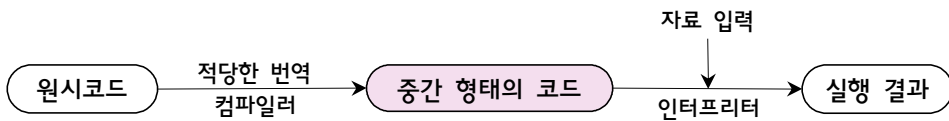
- ① 하이브리드 기법에서는 인터프리터가 중간언어로 번역된 프로그램을 해석하고 실행한다.
- ② 인터프리터를 이용한 기법에서는 고급언어 프로그램을 명령문 단위로 하나씩 해석하여 바로 실행한다.
- ③ 반복문이 많은 프로그램의 실행에서 컴파일러를 이용한 기법이 인터프리터를 이용한 기법보다 효율적이다.
- ④ 인터프리터를 이용한 기법은 번역된 프로그램을 저장하기 위한 큰 기억장소를 요구하는 단점이 있다.

☞ 컴파일러와 인터프리터

컴파일러	인터프리터
<ul style="list-style-type: none"> • 변수에 대한 타입은 번역시간에 검사된다. • 정적타입 검사이다. • 프로그램을 안정적인 구현할 수 있다. • 부프로그램 호출이나 반복처리 되는 프로그램에 적용하면 실행 효율이 좋다. • 번역된 프로그램이 저장될 큰 공간이 필요 • 심볼테이블은 번역 과정에서 데이터베이스로 사용된다. • 반복 운영 시스템에 적합 	<ul style="list-style-type: none"> • 실행 중에 자료의 동적 변화를 구현하므로 유연성이 좋다.(동적타입) • 실행시간에 오류 메시지를 참조할 수 있어서 디버깅하기가 쉽다. • 실행 중에 복잡한 고급 문장을 해석하므로 실행속도는 느리나 구현하기는 쉽다. • 실행 중에 원시코드와 심볼테이블이 기억공간에 존재해야 한다. • 개발 시스템, 교육 시스템에 적합

// 하이브리드 구현

고급언어로 작성된 원시코드를 입력자료로 읽어서 좀 더 실행시키기 쉬운 형태로 먼저 번역한다. 그리고 번역된 코드를 다시 해석하여 프로그램을 실행시킨다.



• JAVA는 하이브리드 구현 기법을 사용하는 언어이다.