

| | | |
|----------------|----------|---------|
| 4과목 | 프로그래밍언어론 | (46~60) |
| 출제위원 : 방송대 김진욱 | | |

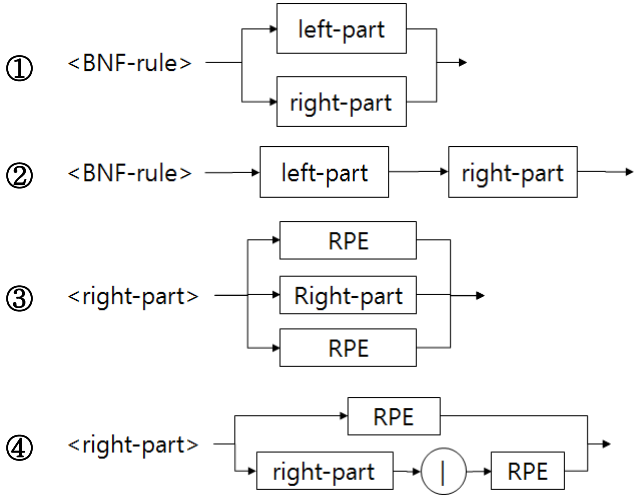
46. 프로그래밍 언어에 대한 설명과 거리가 먼 것은?
- ① 컴퓨터가 읽을 수 있고, 사람이 읽을 수 있는 형식으로 계산을 서술하는 표기체계
 - ② 컴퓨터에게 프로그래머의 의사를 전달하는 방법
 - ③ 오랜 시간을 통해 형성된 비형식적이고 의미적인 표현 방법
 - ④ C, C++, Java, Pascal 등이 있음
47. 자료형과 연산으로 이루어지는 객체에 기반하는 프로그래밍 언어는 무엇인가?
- ① 명령형 언어
 - ② 함수형 언어
 - ③ 논리형 언어
 - ④ 객체 지향 언어
48. 프로그래밍 언어의 전형과 예가 바르게 짝지어진 것은 무엇인가?
- ① 명령형 언어 - C
 - ② 함수형 언어 - Prolog
 - ③ 논리형 언어 - LISP
 - ④ 객체 지향 언어 - Fortran
49. 프로그래밍 언어의 설계기준 중 프로그래머에게 언어의 특징을 추가할 수 있는 권한을 부여하는 것은 무엇인가?
- ① 효율성
 - ② 직교성
 - ③ 확장성
 - ④ 부분성
50. 다음 프로그램에 대한 분석이 잘못된 것은 무엇인가?

```
int start;
```

- ① 문자: i, n, t, s 등
 - ② 어휘: int, start 등
 - ③ 예약어: start
 - ④ 구문: <변수선언> ::= <자료형> <변수> ;
- ※ (51~53) EBNF로 표현된 다음 구문을 예제로 질문에 답하시오.
- <BNF-rule> ::= <left-part> ‘::=’ <right-part>
<right-part> ::= <RPE> | <right-part> ‘|’ <RPE>

51. 위 구문에 대한 설명으로 올바른 것은 무엇인가?
- ① <BNF-rule>은 단말 기호임
 - ② ‘::=’는 메타 기호로 정의를 의미
 - ③ | 는 비단말 기호임
 - ④ 서로 다른 비단말 기호가 네 가지 있음
52. 위 구문의 <right-part>와 정의가 동일한 것은 무엇인가?
- ① <right-part> ::= <RPE> [‘|’ <RPE>]
 - ② <right-part> ::= <RPE> { ‘|’ <RPE> }
 - ③ <right-part> ::= [<RPE>]
 - ④ <right-part> ::= { <RPE> }

53. 다음은 위 구문을 구문 도표로 나타낸 것이다. 올바른 것은?



54. 동일 표현에 대해 서로 다른 파스 트리가 만들어지는 문법을 무엇이라고 하는가?
- ① 명확한 문법
 - ② 모호한 문법
 - ③ 형식 문법
 - ④ 추상 문법

55. 컴파일 기법에 대한 설명이 아닌 것은 무엇인가?
- ① 추가 기억장소가 필요 없음
 - ② 반복 처리가 많은 프로그램에 효율적
 - ③ 실행 시간이 중요한 경우에 적합
 - ④ 한번 번역된 코드는 다시 번역할 필요 없음

56. 변수의 네 가지 요소에 해당하지 않는 것은 무엇인가?
- ① 식별자
 - ② 자료속성의 집합
 - ③ 주소
 - ④ 예약어

57. 바인딩 시간과 예가 올바르게 짝지어진 것은 무엇인가?
- ① 동적 바인딩 - 변수의 값
 - ② 정적 바인딩 - 변수의 주소
 - ③ 언어의 구현 시간 - 프로그램 내의 상수를 10진수로 간주
 - ④ 언어의 정의 시간 - 정수의 자릿수

58. 선언문의 목적과 거리가 먼 것은 무엇인가?
- ① 주기억장치 사용의 효율성
 - ② 주기억장치 접근방법의 효율성
 - ③ 동적 형 검사 가능
 - ④ 주기억장치 경영의 효율성

59. 할당문의 종류와 예가 올바르게 짝지어진 것은 무엇인가?
- ① 단순 할당문: A += B
 - ② 다중 목적변수 할당문: A, B = C
 - ③ 복합 할당 연산자: A++
 - ④ 혼합형 할당문: flag ? A : B = C

60. 제어문과 예가 올바르게 짝지어진 것은 무엇인가?
- ① 조건문: for
 - ② 반복문: while
 - ③ 할당문: =
 - ④ 반복문: switch