

배우는 내용

1. 프로그래밍언어 개요
2. 구문 구조-언어가 제공하는 의미와 형식 개요
3. 변수-속성, 바인딩, 타입검사..
4. 타입-문자열, 배열, 포인터, 구조적 타입도
5. 제어구조-수식, assignment, 반복문
6. **부프로그램, argument** 구조적 언어
7. 객체, 추상데이터타입
8. 함수형 (functional) 언어
9. 논리적 (logical)언어 새로운 개념

9 Subprogram 다형성

- Adhoc polymorphism (특이 다형성) ~= Overloaded subprogram(중복 부프로그램)

```
double max(double x, double y)           // max#2
{ return x>y ? x : y; }
int      max(int      x, int      y, int z) // max#3
{ return x>y ? (x>z ? x : z) : (y > z ? y : z); }
```

- Parametric polymorphic program (인자 다형성)

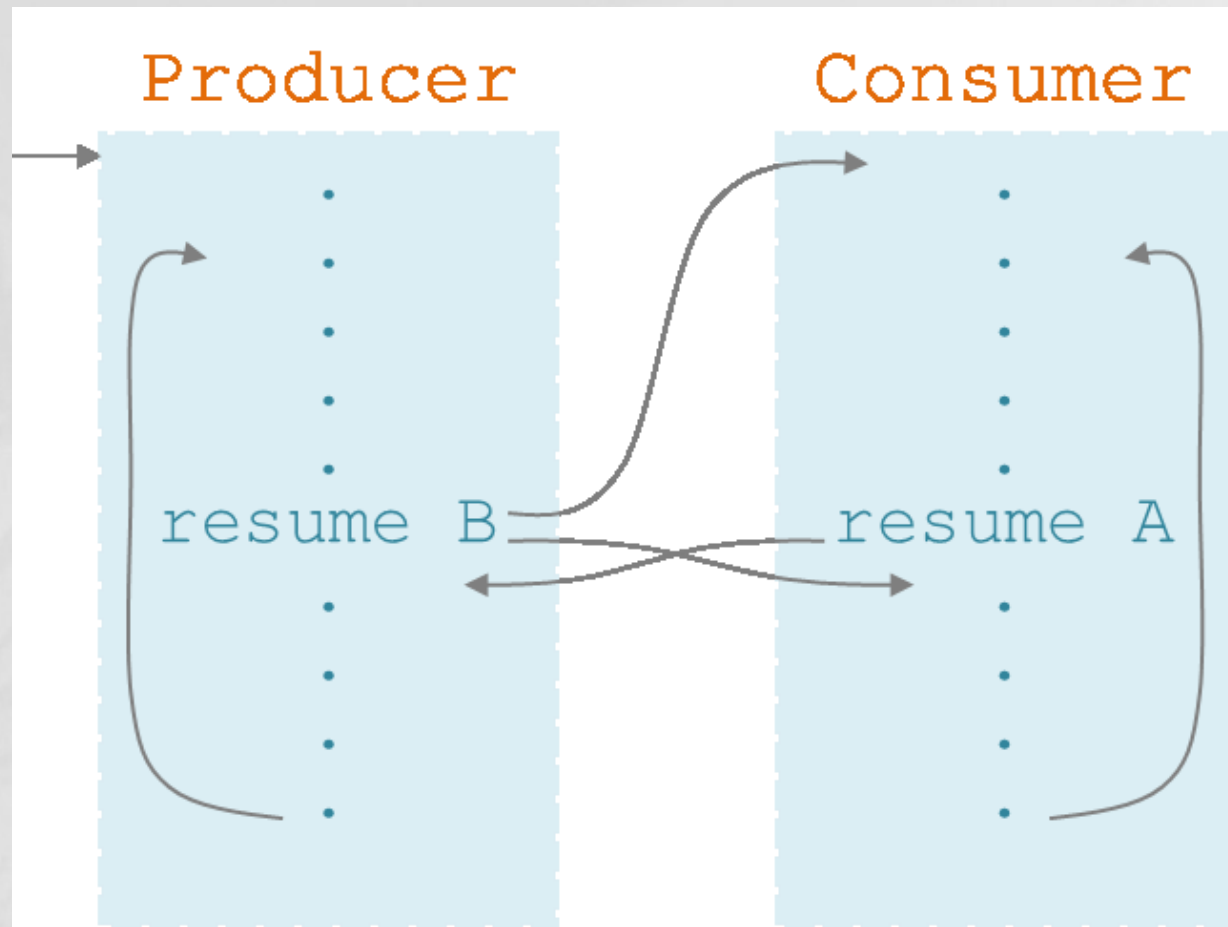
```
public static <T> T goo(T[] ga) // T: generic
parameter
{ ... }
```

- 호출하기

```
String[] names = {"Kim", "Lee", "Park"};
String result = <String> goo(names);
```

그 밖에

- ◉ 함수 정의시 고려사항
- ◉ 사용자 정의 operator overloading
- ◉ Coroutine



예) C#에서는 return 외에 yield return 이라는 것이 존재한다. return 후 다시 불리게 되면 함수 처음부터 수행되는 것이 아니라 직전에 return 한 부분 이후부터 수행하게 된다. -> 이게 coroutine