



5장. 변수

- 5.1 변수, 바인딩
- 5.2 선언, 블록, 영역
- 5.3 할당



5.1 변수

변수 ?

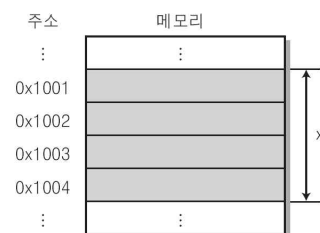
■ 변수의 다양한 속성들



■ 예 `int x;`

① 이름 : 이름은 x

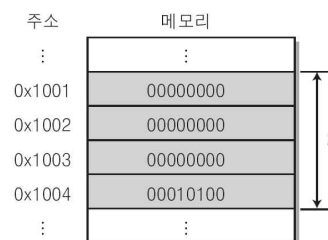
② 주소 : 변수의 값이 저장될 메모리 주소
0x1001 ~ 0x1004



변수 ?

③ 값 : 변수의 값은 배정문에 의해 부여, 변수 x의 값은 20

`x = 20;`



④ 타입 : 변수가 가질 수 있는 값의 범위와 이 값에 대한 연산들의 집합을 의미

⑤ 영역 : 변수 x의 사용이 허락되는 범위

⑥ 수명 : 변수 x가 메모리 주소에 할당되어 있는 기간



변수의 속성-주소

■ 주소

- 변수의 값이 저장될 메모리 주소
- 예

```
int x;  
char ch;
```



- C/C++에서 변수의 시작 주소

```
printf("%p", &x);
```



변수의 속성-값

■ 값

- 변수의 주소에 저장된 내용
- 변수의 값은 배정문에 의해 부여

```
x = 20;
```

 x의 주소에 값 20을 저장한다는 의미

```
x = y;
```

 y의 값을 x의 주소에 저장하는 것을 의미

- l-value와 r-value(l : left, r : right)



변수의 속성-값

■ 값

- 변수의 주소에 저장된 내용
- 변수의 값은 배정문에 의해 부여

`x = 20;` x의 주소에 값 20을 저장한다는 의미

`x = y;` y의 값을 x의 주소에 저장하는 것을 의미

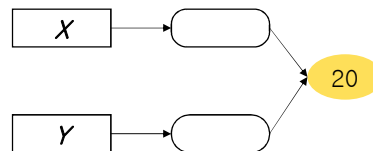
- l-value와 r-value(l : left, r : right)



변수의 속성-값

- 주소와 값
- 별명(aliasing) : 둘이상의 이름이 동일 주소를 참조할 때.

```
int *x, *y;  
x = (int *) malloc(sizeof(int));  
*x = 10;  
y = x;  
*y = 20;  
printf("%d\n", *x);
```



- 부작용(side effect) : 문장의 실행 후 값이 변경되는 오류



바인딩(Binding)

- 바인딩은 무엇인가?
 - 이름이 의미하는 대상 혹은 이름의 속성을 결정하는 것
- 바인딩은 왜 필요할까?
 - 프로그램에 등장하는 이름들이 다양하다.
 - 변수, 상수, 클래스, 메소드, 인터페이스, 매개변수 등.
 - 하나의 이름이 여러 용도로 사용될 수도 있다.
 - 따라서 이름이 나타나면 그 대상 및 속성이 결정되어야 한다.
- 바인딩 시간
 - 정적 바인딩
 - compile time
 - 동적 바인딩
 - execution time

바인딩	설명
정적 바인딩	실행 시간 전에 일어나고 프로그램 실행 과정에서 변하지 않은 상태로 유지되는 바인딩
동적 바인딩	실행 시간 중에 일어나거나 프로그램 실행 과정에서 변경되는 바인딩



정적 바인딩(Static Binding)

- 컴파일 시간에 이루어지는 바인딩
- 정적 속성(Static attribute)
 - 정적으로 바인딩 되는 속성
- 예제
 - `const int n=2;`
 - `int x;`
 - 정적으로 결정되는 것은 무엇인가?



동적 바인딩(Dynamic Binding)

- 실행 시간에 이루어진 바인딩.
- 동적 속성
 - 동적으로 바인딩 되는 속성
- Java 예제

```
C x;  
if (...) x = new C( );      /* memory allocation */  
else x = new C1( );        /* C1 is a subclass of C */  
x.m( ... );                /* method dispatch */
```
- 동적으로 바인딩되는 것은 무엇인가?

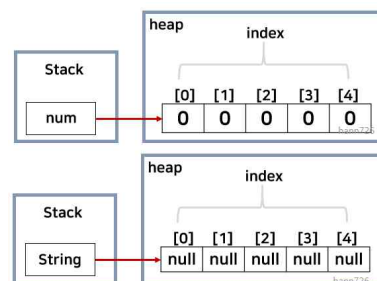


메모리 영역-확인 (JAVA-'new')

```
int[] array;  
array = { 10, 20, 30, 40, 50 }; // 컴파일 에러!
```

↓

```
int[] array;  
array = new int[] {10, 20, 30, 40, 50};
```





바인딩 시간 세분화

바인딩 시간	설명
언어 정의 시간	프로그래밍 언어를 정의할 때 이루어지는 바인딩
언어 구현 시간	언어에 대한 번역기를 구현할 때 이루어지는 바인딩
번역 시간	원시 프로그램을 번역할 때 이루어지는 바인딩
링크 시간	프로그램을 라이브러리와 링크할 때 이루어지는 바인딩
적재 시간	프로그램의 실행을 위해 메모리로 적재할 때 이루어지는 바인딩
실행 시간	프로그램을 실행할 때 이루어지는 바인딩

- 바인딩 시간의 예
 - 언어 정의 시간
 - C 언어의 int 타입이 정수 값을 가진다는 의미를 바인딩
 - '+' 기호에 덧셈 연산자라는 의미를 바인딩



바인딩 시간 세분화

- 바인딩 시간의 예
 - 언어 정의 시간
 - C 언어의 int 타입이 정수 값을 가진다는 의미를 바인딩
 - '+' 기호에 덧셈 연산자라는 의미를 바인딩
 - 언어 구현 시간
 - C 언어의 int 타입에 가능한 값의 범위를 바인딩
 - 정수에 컴퓨터 내에서의 표기법을 바인딩
 - 번역 시간
 - 다음 C 선언문에서 변수 x에 타입 int를 바인딩
int x;
 - 링크 시간
 - 라이브러리 부프로그램 호출에 부프로그램 코드를 바인딩
 - 적재 시간
 - 전역 변수에 메모리 주소를 바인딩
 - 실행 시간
 - 다음 C 배정문에서 변수 x에 값 20을 바인딩
x = 20;



바인딩 시간 세분화(예)

■ $X := Y + 10;$

바인딩 시간	설명
언어 정의 시간	X, Y의 가능한 타입의 집합 / $:=$ 및 $+$ 기호의 선택 십진수의 표현방법 / 평가순서
언어 구현 시간	X, Y가 갖을 수 있는 가능한 값의 집합 / 더하기의 의미 배정문에 대해 생성되는 코드 / 10을 표현하기 위한 비트의 나열 방법
컴파일 시간	X, Y와 타입의 바인딩(X, Y와 관련된 특정 타입결정) 실수, 정수, 복소수 등의 덧셈 연산 결정
링크/로드 시간	X, Y를 기억 장소에 바인딩 코드를 기억 장소에 위치시키고 절대값을 주소에 배정
실행 시간	Y가 평가됨 Y의 값을 10 증가시키고 X라는 기억 장소에 할당