Java를 알고 C배우기

컴퓨터프로그래밍3

week 7-3 포인터와 함수-const

2022.1학기
충남대 조은선

포인터에 Const 선언

- const의 위치는 두 가지 - 의미가 다르다

```
int num = 20;
const int * ptr1  = & num;
int * const ptr2 = & num;
```

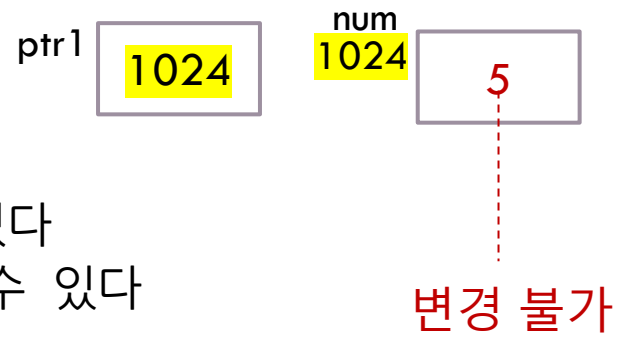
차이는?

참고) 두가지를 동시에 선언할 수도 있음

```
const int * const ptr2  = & num;
```

(1) const int * ptr

- 의미: “ptr1을 통해 꺼내올 값은 변경될 수 없습니다.”




```
int num = 5;
const int * ptr1 = &num;
*ptr1 = 30;            // 오류!- ptr1을 통해서는 변경될 수 없다
num = 40;              // 이건 성공 - num을 통해서는 변경될 수 있다
```

- 상수로 선언된 것의 주소를 가져올 수도 있음

```
const int num = 5;
int * ptr = &num;            // 오류!
const int * ptr = &num       // 이건 성공- const int * 이므로 const int 변수의 주소값을 담을 수 있다
```

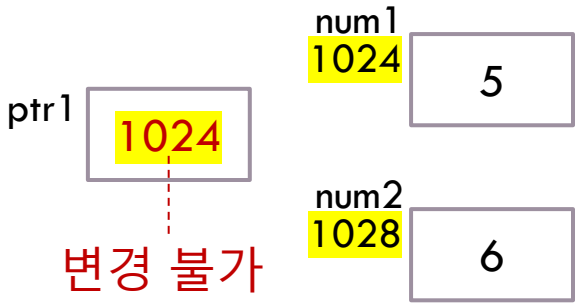
- 자기 자신은 변경가능

```
const int num1 = 5;
const int num2 = 10;
const int * ptr = &num1;
ptr = &num2;                 // 얼마든지 다른 주소값으로 재지정하는 것 가능
```

(2) int * const ptr

- 의미: "ptr2에 저장된 주소값 자체는 변경이 불가능합니다."

```
int num1 = 5;
int * const ptr2 = &num1;      // const double pi = 3.14; 처럼 선언할 때 초기화 해야함
ptr2 = ptr1;                   // 오류! 새로운 주소값으로 재지정하는 것은 불가능
ptr2 = ptr2 + 1;               // 오류! 어떤 값으로 지정해도
ptr2 = &num2;                  // 오류! 변경 불가능
```



- 배열 이름

```
int arr[] = {1, 2, 3};
int * ptr;
ptr = arr;      // 가능
arr = ptr;      // 불가능! 배열 이름은 일반 포인터와는 달리 변경 불가능
                // 즉 int arr[] 는 알고보면 int * const 타입이다!
```

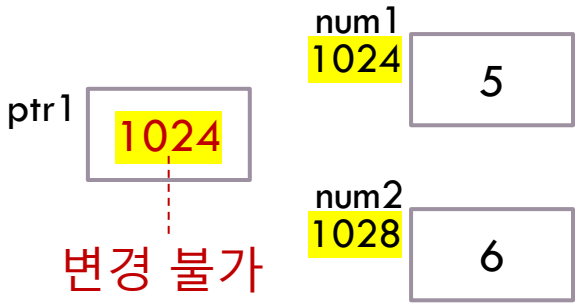
- 포인터를 통해 얻은 값을 변경하는 것은 가능

```
int num = 5;
int * const ptr = &num;
*ptr = 6;       // 얼마든지 해당 주소에 들어있는 값을 바꾸는 것은 가능
```

(2) int * const ptr

- 의미: "ptr2에 저장된 주소값 자체는 변경이 불가능합니다."

```
int num1 = 5;
int * const ptr2 = &num1;      // const double pi = 3.14; 처럼 선언할 때 초기화 해야함
ptr2 = ptr1;                   // 오류! 새로운 주소값으로 재지정하는 것은 불가능
ptr2 = ptr2 + 1;               // 오류! 어떤 값으로 지정해도
ptr2 = &num2;                  // 오류! 변경 불가능
```



- 배열 이름

```
int arr[] = {1, 2, 3};
int * ptr;
ptr = arr;      // 가능
arr = ptr;      // 불가능! 배열 이름은 일반 포인터와는 달리 변경 불가능
                // 즉 int arr[] 는 알고보면 int * const 타입이다!
```

- 포인터를 통해 얻은 값을 변경하는 것은 가능

```
int num = 5;
int * const ptr = &num;
*ptr = 10;      // 얼마든지 해당 주소에 들어있는 값을 바꾸는 것은 가능
```

문자열 변수 복습

- 문자 배열로 선언

```
char arr1[] = “first”;
arr1[2] = ‘p’; // OK
arr1 = “second”; // 오류
```

즉, arr1은 **char * const** 타입

- 자동할당 문자열과 char* 포인터로 선언

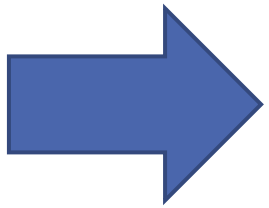
```
char * arr2 = “first”;
arr2[2] = “p”;          // 오류
arr2 = “second”         // OK
```

즉, arr2는 **const char *** 처럼 쓰임

안전한 코드

- const 선언이 갖는 의미: 안전성이 향상된다 (중요)

```
int main() {
  double PI = 3.1415; // 절대로 값이 바뀌면 안된다고 개발자 혼자 생각함
  double r;
  PI = 3.07; // 실수로 잘못 삽입, 컴파일러가 발견 못함
  scanf("%lf", &r);
  printf("area %f \n", r*r*PI);
  return 0;
```



```
int main() {
    const double PI = 3.1415;
    double r;
    PI = 3.07;      // 컴파일러가 오류를 냄!
    scanf("%lf", &rad);
    printf("area %f \n", r*r*PI);
    return 0;
```

좀 어려워도 잘 배워둡시다

Quiz