

기초프로그래밍

제4장 상수

Sangsoo Lim

CSAI

Dongguk University

차례

- 상수
- 리터럴 상수와 심볼릭 상수

상수란

- 상수
 - 어떤 상황에서도 변하지 않는 값을 의미
- 프로그램에서 데이터는 **변수** 또는 **상수**의 형태로 사용한다
 - **변수의 경우**

```
int i;  
i=3;  
i=4;
```

→ 데이터를 올바르게 사용한 경우

- **상수의 경우**

```
10=5;
```

→ 데이터를 올바르게 사용하지 못한 경우

리터럴 상수와 심볼릭 상수

- 상수의 종류
 - **리터럴 상수**: 글자 그대로의 의미가 있어서 이름이 없는 상수
 - 정수형 상수
 - 실수형 상수
 - 문자 상수
 - 문자열 상수
 - **심볼릭 상수**: 상수를 기호화하여 변수처럼 이름이 있는 상수
 - `const` 키워드 이용하기
 - `#define`문 이용하기

리터럴 상수와 심볼릭 상수

- 리터럴 상수 - 정수형 상수 예제 실습

```
/* 4-1.c */
#include<stdio.h>
int main(void)
{
    printf("10진수 정수형 상수 %d + %d = %d 입니다. \n", 10, 20, 10+20);
    printf("16진수 정수형 상수 %x + %x = %x 입니다. \n", 0x10, 0x20, 0x10+0x20);
    printf(" 8진수 정수형 상수 %o + %o = %o 입니다. \n", 010, 020, 010+020);
    return 0;
}
```

리터럴 상수와 심볼릭 상수

- 리터럴 상수 - 실수형 상수 예제 실습

```
/* 4-2.c */
#include<stdio.h>
int main(void)
{
    printf("실수형 상수 %lf + %lf = %lf 입니다. \n", 3.1, 4.1, 3.1+4.1);
    return 0;
}
```

리터럴 상수와 심볼릭 상수

- 리터럴 상수 – 문자 상수 예제 실습
 - 알파벳과 특수 기호가 왜 문자 상수 인가?

```
/* 4-3.c */
#include<stdio.h>
int main(void)
{
    printf("문자 상수 %c %c %c 입니다.\n", 'a', 'b', 'c');
    printf("문자 상수 %c %c %c 입니다.\n", '!', '@', '#');
    return 0;
}
```

문자 상수 a b c 입니다.
문자 상수 ! @ # 입니다.

리터럴 상수와 심볼릭 상수

- 리터럴 상수 – 문자 상수 예제 실습
 - ASCII 코드

| dec | hex | oct | char | dec | hex | oct | char | dec | hex | oct | char | dec | hex | oct | char |
|-----|-----|-----|------|-----|-----|-----|-------|-----|-----|-----|------|-----|-----|-----|------|
| 0 | 0 | 000 | NULL | 32 | 20 | 040 | space | 64 | 40 | 100 | @ | 96 | 60 | 140 | ` |
| 1 | 1 | 001 | SOH | 33 | 21 | 041 | ! | 65 | 41 | 101 | A | 97 | 61 | 141 | a |
| 2 | 2 | 002 | STX | 34 | 22 | 042 | " | 66 | 42 | 102 | B | 98 | 62 | 142 | b |
| 3 | 3 | 003 | ETX | 35 | 23 | 043 | # | 67 | 43 | 103 | C | 99 | 63 | 143 | c |
| 4 | 4 | 004 | EOT | 36 | 24 | 044 | \$ | 68 | 44 | 104 | D | 100 | 64 | 144 | d |
| 5 | 5 | 005 | ENQ | 37 | 25 | 045 | % | 69 | 45 | 105 | E | 101 | 65 | 145 | e |
| 6 | 6 | 006 | ACK | 38 | 26 | 046 | & | 70 | 46 | 106 | F | 102 | 66 | 146 | f |
| 7 | 7 | 007 | BEL | 39 | 27 | 047 | ' | 71 | 47 | 107 | G | 103 | 67 | 147 | g |
| 8 | 8 | 010 | BS | 40 | 28 | 050 | (| 72 | 48 | 110 | H | 104 | 68 | 150 | h |
| 9 | 9 | 011 | TAB | 41 | 29 | 051 |) | 73 | 49 | 111 | I | 105 | 69 | 151 | i |
| 10 | a | 012 | LF | 42 | 2a | 052 | * | 74 | 4a | 112 | J | 106 | 6a | 152 | j |
| 11 | b | 013 | VT | 43 | 2b | 053 | + | 75 | 4b | 113 | K | 107 | 6b | 153 | k |
| 12 | c | 014 | FF | 44 | 2c | 054 | , | 76 | 4c | 114 | L | 108 | 6c | 154 | l |
| 13 | d | 015 | CR | 45 | 2d | 055 | - | 77 | 4d | 115 | M | 109 | 6d | 155 | m |
| 14 | e | 016 | SO | 46 | 2e | 056 | . | 78 | 4e | 116 | N | 110 | 6e | 156 | n |
| 15 | f | 017 | SI | 47 | 2f | 057 | / | 79 | 4f | 117 | O | 111 | 6f | 157 | o |
| 16 | 10 | 020 | DLE | 48 | 30 | 060 | 0 | 80 | 50 | 120 | P | 112 | 70 | 160 | p |
| 17 | 11 | 021 | DC1 | 49 | 31 | 061 | 1 | 81 | 51 | 121 | Q | 113 | 71 | 161 | q |
| 18 | 12 | 022 | DC2 | 50 | 32 | 062 | 2 | 82 | 52 | 122 | R | 114 | 72 | 162 | r |
| 19 | 13 | 023 | DC3 | 51 | 33 | 063 | 3 | 83 | 53 | 123 | S | 115 | 73 | 163 | s |
| 20 | 14 | 024 | DC4 | 52 | 34 | 064 | 4 | 84 | 54 | 124 | T | 116 | 74 | 164 | t |
| 21 | 15 | 025 | NAK | 53 | 35 | 065 | 5 | 85 | 55 | 125 | U | 117 | 75 | 165 | u |
| 22 | 16 | 026 | SYN | 54 | 36 | 066 | 6 | 86 | 56 | 126 | V | 118 | 76 | 166 | v |
| 23 | 17 | 027 | ETB | 55 | 37 | 067 | 7 | 87 | 57 | 127 | W | 119 | 77 | 167 | w |
| 24 | 18 | 030 | CAN | 56 | 38 | 070 | 8 | 88 | 58 | 130 | X | 120 | 78 | 170 | x |
| 25 | 19 | 031 | EM | 57 | 39 | 071 | 9 | 89 | 59 | 131 | Y | 121 | 79 | 171 | y |
| 26 | 1a | 032 | SUB | 58 | 3a | 072 | : | 90 | 5a | 132 | Z | 122 | 7a | 172 | z |
| 27 | 1b | 033 | ESC | 59 | 3b | 073 | ; | 91 | 5b | 133 | [| 123 | 7b | 173 | { |
| 28 | 1c | 034 | FS | 60 | 3c | 074 | < | 92 | 5c | 134 | \ | 124 | 7c | 174 | |
| 29 | 1d | 035 | GS | 61 | 3d | 075 | = | 93 | 5d | 135 |] | 125 | 7d | 175 | } |
| 30 | 1e | 036 | RS | 62 | 3e | 076 | > | 94 | 5e | 136 | ^ | 126 | 7e | 176 | ~ |
| 31 | 1f | 037 | US | 63 | 3f | 077 | ? | 95 | 5f | 137 | - | 127 | 7f | 177 | DEL |

리터럴 상수와 심볼릭 상수

- 리터럴 상수 – 문자 상수 예제 실습

```
/* 4-4.c */
#include<stdio.h>
int main(void)
{
    printf("문자 상수 %c %c %c 는 \n", 'a', 'b', 'c');
    printf("ASCII 코드10진수로 %d %d %d \n", 'a', 'b', 'c');
    printf("ASCII 코드16진수로 %x %x %x \n", 'a', 'b', 'c');

    printf("\n-----\n");

    printf("문자 상수 %c %c %c 는 \n", '!', '@', '#');
    printf("ASCII 코드 10진수로 %d %d %d \n", '!', '@', '#');
    printf("ASCII 코드 16진수로 %x %x %x \n", '!', '@', '#');

    return 0;
}
```

문자 상수 a b c 는
ASCII 코드10진수로 97 98 99
ASCII 코드16진수로 61 62 63

문자 상수 ! @ # 는
ASCII 코드 10진수로 33 64 35
ASCII 코드 16진수로 21 40 23

리터럴 상수와 심볼릭 상수

- 리터럴 상수 - 문자열 상수 예제 실습

```
/* 4-5.c */
#include<stdio.h>
int main(void)
{
    printf("문자열 상수는 %s 입니다. \n", "A");
    printf("문자열 상수는 %s 입니다. \n", "10+10");
    printf("문자열 상수는 %s 입니다. \n", "Hi, everyone");
    return 0;
}
```

문자열 상수는 A 입니다.
문자열 상수는 10+10 입니다.
문자열 상수는 Hi, everyone 입니다.

리터럴 상수와 심볼릭 상수

- **심볼릭 상수:** 상수를 기호화하여 변수처럼 이름이 있는 상수

- ① **const 키워드 이용하기**
- ② **#define문 이용하기**

```
/* 4-6.c */
#include<stdio.h>
int main(void)
{
    const int NUM = 100;
    const double PI = 3.14;

    // NUM = 200;
    // PI = 4.14;

    return 0;
}
```

```
/* 4-8.c */
#include <stdio.h>

#define PI 3.14
#define NUM 100
#define BUFFER_SIZE 200

int main()
{
    printf("%lf \n", PI);
    printf("%d \n", NUM);
    printf("%d \n", BUFFER_SIZE);

    return 0;
}
```

Summary

- 상수의 개념
- 리터럴 상수의 종류와 의미
- 심볼릭 상수의 종류와 만드는 방법

1. 정수 리터럴 상수 응용

- 예제 1-1: 다양한 진법 상수와 연산

```
#include <stdio.h>

int main(void)
{
    // 10진수, 16진수, 8진수 리터럴
    int dec = 10;      // 10진수
    int hex = 0x10;    // 16진수 (앞에 0x)
    int oct = 010;    // 8진수 (앞에 0)

    // 각각을 10진수 형태로 출력
    printf("10진수 dec = %d\n", dec);
    printf("16진수 hex = %d\n", hex);
    printf("8진수 oct = %d\n", oct);

    // 상수끼리의 연산
    printf("hex + oct = %d\n", 0x10 + 010);
    printf("dec + hex + oct = %d\n", 10 + 0x10 + 010);

    return 0;
}
```

1. 정수 리터럴 상수 응용

- 예제 1-2: 2진수 직접 변환

```
#include <stdio.h>

int main(void)
{
    // 2진수를 직접 표현하는 표준 문법은 없으나, 16진수 / 8진수를 통해 변환 가능
    // 예: 2진수 1010(10진수 10)을 16진수 0xA로 표현
    // 아래와 같이 다양한 2진수를 16진수로 변환해두고, 출력해 보는 응용

    int bin1 = 0b1010;    // C99/gcc 일부 확장으로 0b 지원 (컴파일러마다 다름)
    int bin2 = 0b1111;

    printf("0b1010 = %d\n", bin1);
    printf("0b1111 = %d\n", bin2);

    return 0;
}
```

2. 실수 리터럴 상수 응용

- 예제 2-1: 지수 표기법(Scientific Notation)

```
#include <stdio.h>

int main(void)
{
    double d1 = 3.14;
    double d2 = 3.14e1;    // 3.14 × 10^1 = 31.4
    double d3 = 3.14e-2;   // 3.14 × 10^-2 = 0.0314

    printf("d1 = %f\n", d1);
    printf("d2 = %f\n", d2);
    printf("d3 = %f\n", d3);

    return 0;
}
```

2. 실수 리터럴 상수 응용

- 예제 2-2: float vs. double

```
#include <stdio.h>

int main(void)
{
    float fval = 3.1415926535f;    // float 접미사 f
    double dval = 3.1415926535;    // double

    printf("float fval = %.10f\n", fval);
    printf("double dval = %.10f\n", dval);

    return 0;
}
```

3. 문자 리터럴 상수 응용

- 예제 3-1: ASCII 코드 확인

```
#include <stdio.h>

int main(void)
{
    char c1 = 'A';
    char c2 = 'a';

    printf("c1 = %c, ASCII code(10진수) = %d, (16진수) = %x\n", c1, c1, c1);
    printf("c2 = %c, ASCII code(10진수) = %d, (16진수) = %x\n", c2, c2, c2);

    // 여러 문자 상수에 대한 ASCII 코드를 반복문으로 표시해보는 응용도 가능
    return 0;
}
```

5. 심볼릭 상수(const & #define) 응용

- 예제 5-1: const와 #define 비교

```
#include <stdio.h>

#define SIZE 10

int main(void)
{
    const int length = 5;

    // length = 10;    // 컴파일 에러 (const 변경 불가)
    // SIZE = 20;      // 매크로는 이런 식으로 바꾸지 않음. (실제로는 치환됨)

    printf("const로 선언한 length: %d\n", length);
    printf("#define으로 정의한 SIZE: %d\n", SIZE);

    // 둘 다 변하지 않는 값처럼 쓸 수 있지만, 작동 방식이 다름
    return 0;
}
```

5. 심볼릭 상수(const & #define) 응용

- 예제 5-2: #define을 사용한 간단한 매크로

```
#include <stdio.h>

#define PI 3.14159
#define CIRCLE_AREA(r) (PI * (r) * (r))           // 매크로 함수
#define CIRCLE_CIRCUM(r) (2.0 * PI * (r))          // 둘레 매크로

int main(void)
{
    double radius = 2.0;

    printf("반지름 %.2f인 원의 넓이: %.2f\n", radius, CIRCLE_AREA(radius));
    printf("반지름 %.2f인 원의 둘레: %.2f\n", radius, CIRCLE_CIRCUM(radius));

    return 0;
}
```

1. 다양한 진법 정수 리터럴

1. 기능

- 10진수, 8진수, 16진수 형태의 상수를 각각 선언하고, 세 상수를 더한 결과를 10진수로 출력한다.
- 추가로, 더한 결과를 16진수, 8진수 형태로도 출력해볼 것.

2. 요구사항

- 예시: 10진수 10, 16진수 0x10, 8진수 010을 변수로 선언 후 출력.
- printf 형식 지정자를 사용하여 10진수(%d), 16진수(%x), 8진수(%o)로 결과를 표시.

10진수: 10 (dec), 8진수: 010 (oct), 16진수: 0x10 (hex)
합계 (10진수) : 42
합계 (16진수) : 2a
합계 (8진수) : 52

2. 지수 표기법과 정밀도 비교

1. 기능

- float 변수를 하나, double 변수를 하나 선언한다.
- 실수 리터럴을 지수 표기법(e, E)을 포함하여 여러 가지 형태(예: 3.14, 3.14e1, 2.71828e-2)로 입력하고 각각을 출력해 본다.

2. 요구사항

- 같은 실수 값을 float와 double에 저장할 때, 소수점 이하 자리수가 어떻게 달라지는지 출력해볼 것.
- %.4f, %.6f, %e 등 다양한 형식 지정자를 사용해보자.

```
float fval1 = 3.14e1 -> fval1 = 31.4000  
double dval1 = 3.14e1 -> dval1 = 31.400000  
float fval2 = 2.71828e-2 -> fval2 = 0.027182  
double dval2 = 2.71828e-2 -> dval2 = 0.027182
```

3. 심볼릭 상수를 이용한 도형 계산

1. 기능

- 원의 넓이와 둘레를 구하는 프로그램을 작성한다.
- 심볼릭 상수로 원주율(파이, π)을 정의하고(const나 #define 중 하나 필수), 반지름을 입력받아 그에 맞는 넓이와 둘레를 계산하여 출력한다.

2. 요구사항

- const double PI = 3.14159; 또는 #define PI 3.14159 중 적어도 하나 사용.
- 반지름(radius)은 double로 선언하거나, scanf로 사용자 입력을 받아서 저장.
- 원의 넓이 = $\text{PI} \times (\text{radius})^2$
- 원의 둘레 = $2 \times \text{PI} \times \text{radius}$

```
반지름을 입력하세요: 5.0  
원의 넓이: 78.54  
원의 둘레: 31.42
```