

C# 프로그래밍

2019년 06월 27일

강진혁

● 목차

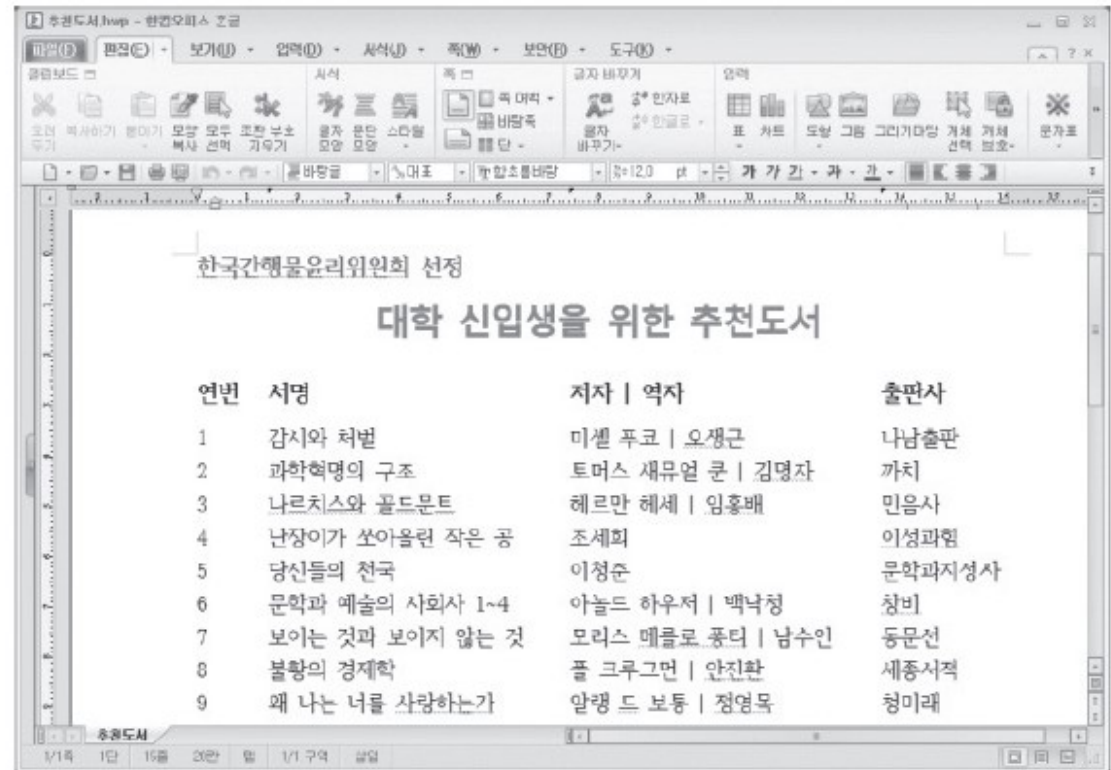
Robot Media Laboratory

1. C# 프로그래밍 언어
2. 자료형
3. 배열
4. 연산자
5. 조건문
6. 반복문
7. 예외처리
8. 네임스페이스
9. 구조체
10. 클래스
11. Nullable
12. 메서드
13. 이벤트
14. 전처리
15. Indexer
16. 접근 제한자
17. 클래스 상속
18. 정적
19. 제네릭
20. 인터페이스
21. Delegate
22. 람다식
23. 익명
24. Partial
25. Dynamic
26. await

C# 프로그래밍 언어

● C# 프로그래밍 언어

Robot Media Laboratory



● C# 프로그래밍 언어

Robot Media Laboratory

● 하드웨어

- 전자회로와 물리적인 장치로 이루어진 것

● 하드웨어의 5대 구성 요소

- 입력장치 : 문자나 기호 같은 데이터를 컴퓨터가 이해하도록 전기 신호로 변환
- 중앙처리장치 : 컴퓨터의 구성 요소를 제어하거나 연산을 수행
- 출력장치 : 중앙처리장치가 처리한 결과를 출력
- 주기억장치 : 컴퓨터 내에서 실행 중인 명령어와 데이터를 저장(RAM)
- 보조기억장치 : 주기억장치를 보조하는 장치로 데이터를 영구적으로 보존
하드디스크, 플로피디스크, CD-ROM 등

● C# 프로그래밍 언어

Robot Media Laboratory

● 소프트웨어

- 하드웨어에 명령을 내려 구동할 수 있도록 만들어진 프로그램(명령어들의 집합)
- 일반적으로 시스템 소프트웨어와 응용 소프트웨어로 나뉨.
- **시스템 소프트웨어** : 컴퓨터를 효율적으로 운영하고 제어하기 위한 프로그램
도스나 윈도우 같은 운영체제, 파일을 압축 · 정렬하는 유틸리티 등
- **응용 소프트웨어** : 특정 분야의 업무를 처리하기 위한 프로그램
문서 편집을 위한 한글이나 워드, 그림 편집을 위한 포토샵, 엑셀 등

● C# 프로그래밍 언어

Robot Media Laboratory

● 프로그래밍 언어

- 기계어(machine language)
 - 0, 1의 이진수로 구성된 언어
 - 컴퓨터의 CPU는 본질적으로 기계어만 처리 가능
- 어셈블리어
 - 기계어의 명령을 ADD, SUB, MOVE 등과 같은 상징적인 니모닉 기호(mnemonic symbol)로 일대일 대응시킨 언어
 - 어셈블러 : 어셈블리어 프로그램을 기계어 코드로 변환
- 고급언어
 - 사람이 이해하기 쉽고 복잡한 작업, 자료 구조, 알고리즘을 표현하기 위해 고안된 언어
 - Pascal, Basic, C/C++, Java, C#
 - 컴파일러 : 고급 언어로 작성된

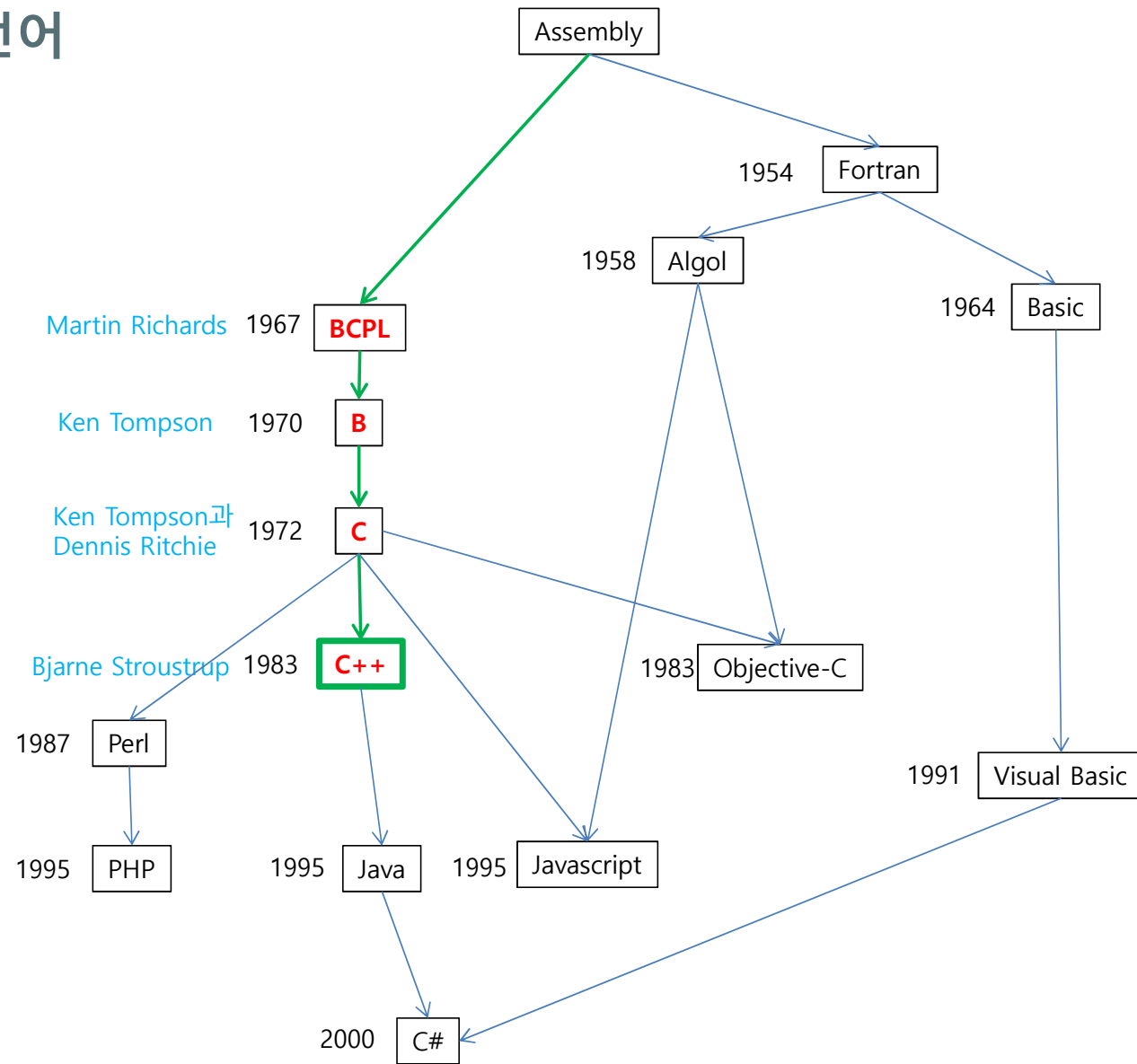
● C# 프로그래밍 언어

Robot Media Laboratory

- 2000년 7월 마이크로소프트에서 개발된 객체 지향 프로그래밍 언어.
- 1983년에 등장한 C++와 1995년에 등장한 Java에서 강한 영향을 받았다.

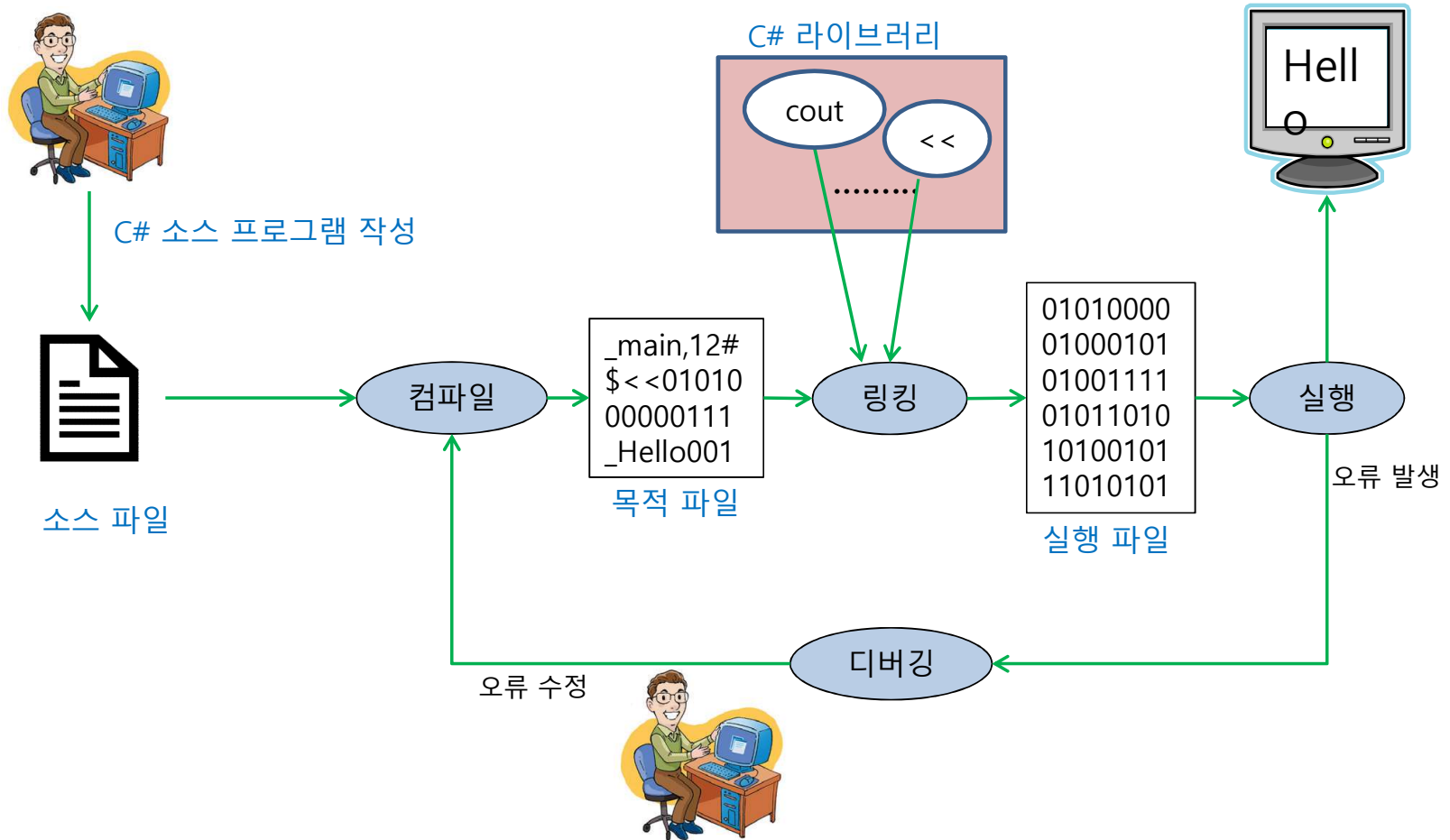
● C# 프로그래밍 언어

Robot Media Laboratory



● C# 프로그래밍 언어

Robot Media Laboratory



● C# 프로그래밍 언어

Robot Media Laboratory

```
1. /*주석*/  
2. //주석  
3. Using System;  
4. namespace ConsoleApplication1 {  
5.     class Program {  
6.         static void Main(string[] args) {  
7.             Console.WriteLine("Hello, world!");  
8.         }  
9.     }  
10.}
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 주석

- 프로그램의 내용을 설명하기 위해 사용
- 컴파일러는 주석을 만나면 기계어로 바꾸지 않고 그냥 넘어감.

● using

- 일반 작업공간을 설정하기 위해 사용
- 선언된 네임스페이스 내부개체들에 접근이 가능해진다.

● namespace

- 관련된 내용들의 그룹
- 특정공간을 지칭

● class

- 필드와 메소드를 포함하고있는 C#기본 단위

● Main

- 최초진입점
- 프로그램의 시작과 종료

● C# 프로그래밍 언어

Robot Media Laboratory

● 식별자(identifier)

- 사용자가 필요에 따라 이름을 만들어 사용하는 단어
- 대표적으로 변수명이나 함수명 등이 해당됨.

● 식별자 작성 규칙

- 영문 대문자와 소문자(A~Z, a~z), 숫자(0~9), 밑줄(_)을 포함한 63개의 문자로만 구성
- 첫 글자로 숫자를 사용할 수 없음.
- 대문자와 소문자를 구별하며, 키워드를 사용할 수 없음.

● C# 프로그래밍 언어

Robot Media Laboratory

● 상수(constant)

- 프로그램이 실행되어 끝날 때까지 값이 변하지 않는 자료
- 문자 1개는 ' ' 를 사용해서 표현하고 문자열은 " " 를 사용해서 표현
- 기호 상수는 전처리문인 #define을 이용하여 상수명과 값을 지정한 후 사용

● C# 프로그래밍 언어

Robot Media Laboratory

● 변수(variable)

- 프로그램에서 자료 값을 임시로 저장하는 장소
- 값이 계속 변할 수 있음.

● 변수 선언

```
int age;  
자료형 변수명;
```

- 이름이 age인 정수형 변수 1개를 선언한다는 의미
- 주기억장치에 age라는 정수형 크기의 메모리가 할당



● C# 프로그래밍 언어

Robot Media Laboratory

● 변수의 초기화

- 선언된 변수에 처음으로 값을 저장하는 것

변수 초기화 방법	사용 예
선언과 동시에 초기화하기	<code>int age = 20;</code>
선언 후에 초기화하기	<code>int age;</code> <code>age = 20;</code>

- 초기값이 저장되기 전에는 사용자가 알 수 없는 무의미한 값(dummy)이 저장됨.
- 변수값을 바꾸는 방법

```
int age = 20;  
age = 22;
```


● C# 프로그래밍 언어

Robot Media Laboratory

● 변수의 동시 선언

- 동일한 자료형의 변수를 여러 개 사용해야 할 때 동시에 선언

변수 선언 방법	사용 예
변수 2개를 따로 선언하기	<pre>int a = 10; int b = 20;</pre>
변수 2개를 동시에 선언하기	<pre>int a = 10, b = 20;</pre>

● C# 프로그래밍 언어

Robot Media Laboratory

1. using System;

2. namespace ConsoleApplication1

3. {

4. class Program

5. {

6. static void Main(string[] args)

7. {

8. int a;

9. a = 1000;

10. Console.WriteLine(a);

11. }

12. }

13.}

● C# 프로그래밍 언어

Robot Media Laboratory

1. using System;

2. namespace ConsoleApplication1

3. {

4. class Program

5. {

6. static void Main(string[] args)

7. {

8. int a = 1000;

9. int b = 2000;

10. int c = 3000;

11. Console.WriteLine(a);

12. }

13. }

14.}

● C# 프로그래밍 언어

Robot Media Laboratory

1. using System;

2. namespace ConsoleApplication1

3. {

4. class Program

5. {

6. static void Main(string[] args)

7. {

8. int a = 1000, b = 2000,c = 3000;

9. Console.WriteLine(a);

10. }

11. }

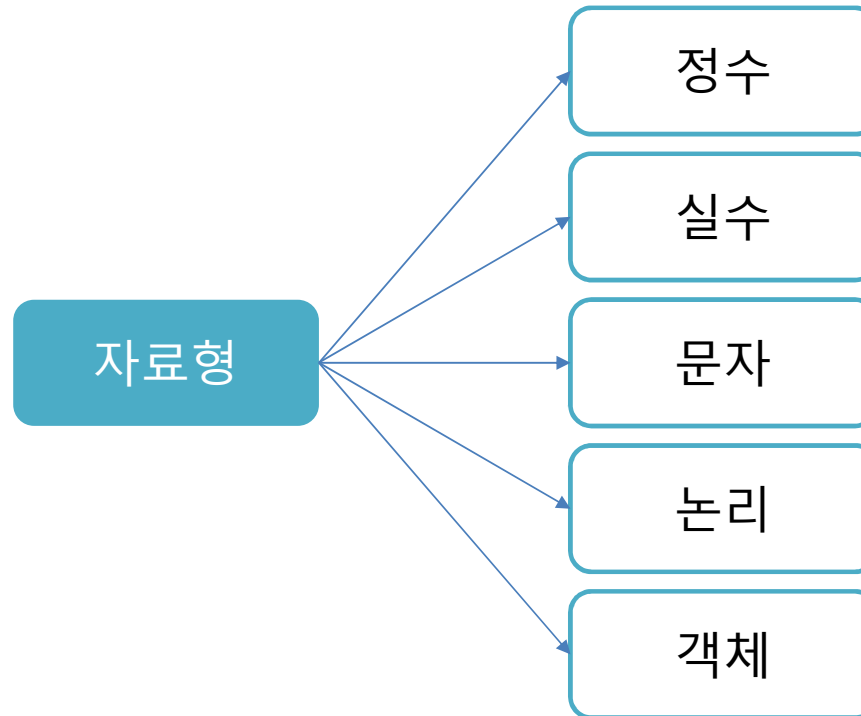
12.}

● C# 프로그래밍 언어

Robot Media Laboratory

● 자료형

- 프로그램에서 선언된 변수들이 기억 공간에서 어떻게 저장되고 처리되어야 할지를 컴파일러에 알려줌.



● C# 프로그래밍 언어

Robot Media Laboratory

구분	데이터 형식	크기(Byte)	값의 범위
정수	byte	1	0~255
정수	sbyte	1	-128~127
정수	short	2	-32,768~32,767
정수	ushort	2	0~65535
정수	int	4	-2,147,483,648~2,147,483,647
정수	uint	4	0~4,294,967,295
정수	long	8	-922,337,203,685,477,508~922,337,203,685,477,507
정수	ulong	8	0~18,446,744,073,709,551,615
문자	char	2	
실수	float	4	-3.402823e38~3.402823e38
실수	double	8	-1.79769313486232e308~1.79769313486232e308
실수	decimal	16	$\pm 1.0 \times 10^{-28} \sim \pm 7.9 \times 10^{28}$
문자열	string		
논리	bool	1	true, false
객체	object		

● C# 프로그래밍 언어

Robot Media Laboratory

● 정수형

- 소수점이 없는 숫자
- 정수형 상수에는 양수, 0, 음수가 있으며 양수, 음수에 따라 +, - 부호 사용 가능



- 오버플로(overflow) : 허용 범위를 넘는 값을 변수에 저장할 경우 사용자가 의도한 값이 아닌 전혀 다른 값이 저장되는 현상

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. using System;
2. namespace ConsoleApplication1
3. {
4.     class Program
5.     {
6.         static void Main(string[] args)
7.         {
8.             byte a = 200;
9.             sbyte b = -126;
10.            short c = 30000;
11.            ushort d = 40000;
12.            int e = 1000000000;
13.            uint f = 10000000000;
14.            long g = 10000000000000000000;
15.            ulong h = 100000000000000000000;
16.            Console.WriteLine("a={0}, b={1}, c={2}, d={3}, e={4}, f={5}, g={6}, h={7}", a, b, c, d, e, f, g, h);
17.        }
18.    }
19.}
```


● C# 프로그래밍 언어

Robot Media Laboratory

● 실수형

- 소수점이나 지수가 있는 수를 의미
- 기본형은 double

● C# 프로그래밍 언어

Robot Media Laboratory

1. using System;

2. namespace ConsoleApplication1

3. {

4. class Program

5. {

6. static void Main(string[] args)

7. {

8. float a = 123.567891011f;

9. double b = 1234.567878910111213;

10. decimal c = 1234.56789101112131415161718m;

11. Console.WriteLine("a={0}, b={1}, c={2}", a, b, c);

12. }

13. }

14.}

● C# 프로그래밍 언어

Robot Media Laboratory

● 문자형

- 문자 1개를 작은따옴표(' ') 안에 넣어서 사용하는 자료형
- 작은따옴표 안에 있는 문자를 데이터 1개로 취급하며, 8비트(1바이트)로 처리
- 내부적인 문자 코드(아스키ASCII나 EBCDIC 코드)에 상응하는 숫자로 바꿔 기억
- 자료형은 char이고, 0~127의 부호 없는 정수에 문자를 정의

● 문자열

- char형 문자를 여러 개 모아 놓은 문자 집합
- 문자는 작은따옴표(' ')로, 문자열은 큰따옴표(" ")로 나타냄.
- 문자열 끝에 항상 문자 코드 '\0'을 덧붙여야 함.
- 자료형은 string

● C# 프로그래밍 언어

Robot Media Laboratory

1. using System;

2. namespace ConsoleApplication1

3. {

4. class Program

5. {

6. static void Main(string[] args)

7. {

8. char a = '안';

9. string b = "녕하세요";

10. Console.WriteLine("{0}{1}", a, b);

11. }

12. }

13.}

● C# 프로그래밍 언어

Robot Media Laboratory

● bool 자료형의 추가

- C언어에서는 참, 거짓의 판별을 위하여 0과 1을 사용하였으나, 가독성이 떨어지기 때문에 `#define TRUE 1` 과 같이 명시하여 사용하였음.
- 참과 거짓을 저장할 수 있는 자료형 `bool`의 추가로 더 이상 `define`할 필요가 없음
- 참 : `true` 거짓 : `false` 으로 사용 함.

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. using System;
2. namespace ConsoleApplication1
3. {
4.     class Program
5.     {
6.         static void Main(string[] args)
7.         {
8.             bool a = true;
9.             bool b = false;
10.            Console.WriteLine("a={0}, b={1}", a, b);
11.        }
12.    }
13.}
```

Robot Media Laboratory

31

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. using System;
2. namespace MySystem
3. {
4.     class Program
5.     {
6.         static void Main(string[] args)
7.         {
8.             string s = "C# Studies";
9.             // 문자열을 배열인덱스로 한문자 액세스
10.            for (int i = 0; i < s.Length; i++)
11.            {
12.                Console.WriteLine("{0}: {1}", i, s[i]);
13.            }
14.            // 문자열을 문자배열로 변환
15.            string str = "Hello";
16.            char[] charArray = str.ToCharArray();
17.            for (int i = 0; i < charArray.Length;
18.                i++)
19.            {
20.                Console.WriteLine(charArray[i]);
21.            }
22.            // 문자배열을 문자열로 변환
23.            char[] charArray2 = { 'A', 'B', 'C', 'D' };
24.            s = new string(charArray2);
25.            Console.WriteLine(s);
26.
27.            // 문자 연산
28.            char c1 = 'A';
29.            char c2 = (char)(c1 + 3);
30.            Console.WriteLine(c2); // D 출력
31.        }
32.    }
33. }
```


- **C# 프로그래밍 언어**

Robot Media Laboratory

- **object 자료형**

- 모든 자료형을 표현 할 수 있는 자료형

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. namespace ConsoleApplication1
6. {
7.     class Program
8.     {
9.         static void Main(string[] args)
10.        {
11.            object a = 12345678910;
12.            object b = 12345.67891011;
13.            object c = true;
14.            object d = "안녕하세요";
15.            Console.WriteLine("a={0}, b={1}, c={2}, d={3}", a, b, c, d);
16.        }
17.    }
18.}
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 형 변환

- 자료형을 바꾸는 것
- 묵시적 형 변환과 명시적 형 변환이 있음.

① 묵시적 형 변환

- 컴파일러가 자동으로 수행
- 데이터의 값을 잃지 않는 방향으로 이루어짐.

char → short → int → unsigned → long → float → double

- 묵시적 형 변환의 예

```
int a = 10;  
float b = 20.05, c;  
c = a + b; // a = 10.00이 되는데,  
이는 정수형이 실수형으로 자동 변환된 것이다.
```

● C# 프로그래밍 언어

Robot Media Laboratory

② 명시적 형 변환

- 사용자가 강제로 지정
- 캐스팅 연산자를 사용하여 형 변환
- 연산식 앞에 ()를 붙이고 () 안에는 변환하는 자료형을 넣음.
- 명시적 형 변환의 예

```
float x;  
float y = 14.66;  
x = (int) y / 2; // y = 14가 되는데, 이는 실수형이 정수형으로 강제 변환된 것.
```

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. namespace ConsoleApplication1
6. {
7.     class Program
8.     {
9.         static void Main(string[] args)
10.        {
11.            int a = 10, b = 20, c;
12.            double d = 5.6, e = 8.4, f1,f2;
13.            c = (int)(d + e);
14.            f1 = d + e;
15.            f2 = d + (int)e;
16.            Console.WriteLine("c=d+b:{0} Wnf=d+e:{1} Wnf=d(int)e:{2}",c,f1,f2);
17.        }
18.    }
19.}
```

- C# 프로그래밍 언어

Robot Media Laboratory

- 연산자(operator)

- 산술 연산자인 +, -, *, /와 같이 이미 정의된 연산을 수행하는 기호

- 피연산자(operand)

- 연산에 참여하는 변수나 값

- 연산자와 피연산자의 구분 예

```
int x = 3, y = 4, z;  
z = x + y;
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 연산자의 종류

종류	예
산술 연산자	+, -, *, /, %
관계 연산자	>, <, ==, !=, >=, <=
증감 연산자	++, --
논리 연산자	&&, , !
조건 연산자	?:
비트 논리 연산자	&, , ^, ~
비트 이동 연산자	<<, >>
대입 연산자	=

● C# 프로그래밍 언어

Robot Media Laboratory

● 대입(할당) 연산자

- 일반적인 수학에서 사용하는 = 기호
- 프로그램에서 가장 기본이 되는 연산자
- 다른 연산자를 이용하여 얻은 의미 있는 문장을 실행할 때 사용

● 연산 진행 순서

```
x = 5 + 6;
```

- 연산자 : +, =
 - 피연산자 : 5, 6, x
- 대입 연산자가 맨 마지막에 수행됨.

● C# 프로그래밍 언어

Robot Media Laboratory

● 수학과와의 차이점

```
x = x + 2;
```

- 연산자 : +, =
- 피연산자 : x, 2

→ 대입 연산자를 기준으로 오른쪽에 있는 수식을 먼저 실행 후 결과를 왼쪽 값에 대입

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. using System;
2. using System.Collections.Generic;
3. using System.Linq;
4. using System.Text;
5. namespace ConsoleApplication1
6. {
7.     class Program
8.     {
9.         static void Main(string[] args)
10.        {
11.            int a, b = 0;

12.            a = 4;
13.            Console.WriteLine("1:{0}", a);
```

```
14.            a = a + 3;
15.            Console.WriteLine("2:{0}", a);

16.            b = a + 2;
17.            Console.WriteLine("3:{0}", b);
18.        }
19.    }
20.}
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 산술 연산자

- 산술 연산을 수행하는 연산자
- 연산자가 다루는 피연산자의 개수에 따라 단항 연산자와 이항 연산자로 나뉨

● 단항 연산자

- 피연산자를 1개만 사용하는 산술 연산자

● 증감 연산자

- 오른쪽에서 왼쪽으로 적용
- 각각 변수값을 1씩 증가 또는 감소시키는 ++와 --가 있음.
- 피연산자의 앞에 붙으면 전치 증가, 피연산자의 뒤에 붙으면 후치 증가라고 함.
- 증감 연산자를 사용하면 연산의 처리 속도가 좀 더 빨라짐.

● C# 프로그래밍 언어

Robot Media Laboratory

● 증감 연산자의 종류

연산자	연산식	설명
++	a ++	변숫값을 수식에 먼저 적용한 후 최종 변숫값을 1만큼 증가시킨다.
	++ a	변숫값을 1만큼 먼저 증가시킨 후 최종 변숫값을 수식에 적용한다.
--	a --	변숫값을 수식에 먼저 적용한 후 최종 변숫값을 1만큼 감소시킨다.
	-- a	변숫값을 1만큼 먼저 감소시킨 후 최종 변숫값을 수식에 적용한다.

● 증감 연산자의 사용 예

연산식	실행 절차	실행 결과
b = ++a	a에 1을 더한 후 그 값을 b에 대입한다.	a = 11, b = 11
b = --a	a에서 1을 뺀 후 그 값을 b에 대입한다.	a = 9, b = 9
b = a++	a를 b에 대입한 후 a에 1을 더한다.	a = 11, b = 10
b = a--	a를 b에 대입한 후 a에서 1을 뺀다.	a = 9, b = 10

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2. {
3.     int a, b, c;
4.         a = 10; b = 20; c = 30;

5.         a = ++b;
6.         c = b++;
7.         Console.WriteLine("a = {0}, b = {1}, c = {2}\n\n", a, b, c);

8.         a = ++b + ++c;
9.         Console.WriteLine("a = ++b + ++c 문장 실행 후\na = {0}, b = {1}, c = {2}\n\n", a, b,
10.            c);
11.         a = b++ + c++;
12.         Console.WriteLine("a = b++ + c++ 문장 실행 후\na = {0}, b = {1}, c = {2}\n\n", a, b,
13.            c);
14.     }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 이항 연산자

- 피연산자가 2개 필요한 산술 연산자
- 사칙연산과 나머지 연산자를 포함

연산자	의미	기능	연산자	의미	기능
+	덧셈	$a = b + c$	-	뺄셈	$a = b - c$
*	곱셈	$a = b * c$	/	나눗셈	$a = b / c$
%	나머지	$a = b \% c$			

● C# 프로그래밍 언어

Robot Media Laboratory

● 나눗셈 연산자

- 정수형 변수에 사용하면 몫만 값으로 저장되고 나머지는 없어짐.

```
int x, y = 9;  
x = y / 2;      // 결과 x에 4가 저장된다.  
x = y / 4;      // 결과 x에 2가 저장된다.
```

● 나머지 연산자

- 나머지만 저장(프로그래밍에서 유용하게 사용)

```
int x, y = 9;  
x = y % 4;      // 결과 x에 1이 저장된다.  
x = y % 6;      // 결과 x에 3이 저장된다.
```

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2. {
3.     int a = 20, b = 3, c = 7, d, e;
4.     double f = 15.0, g;
5.     d = a / b;

6.     Console.WriteLine("d = a / b를 실행한 후 d의 값은 {0}\n", d);

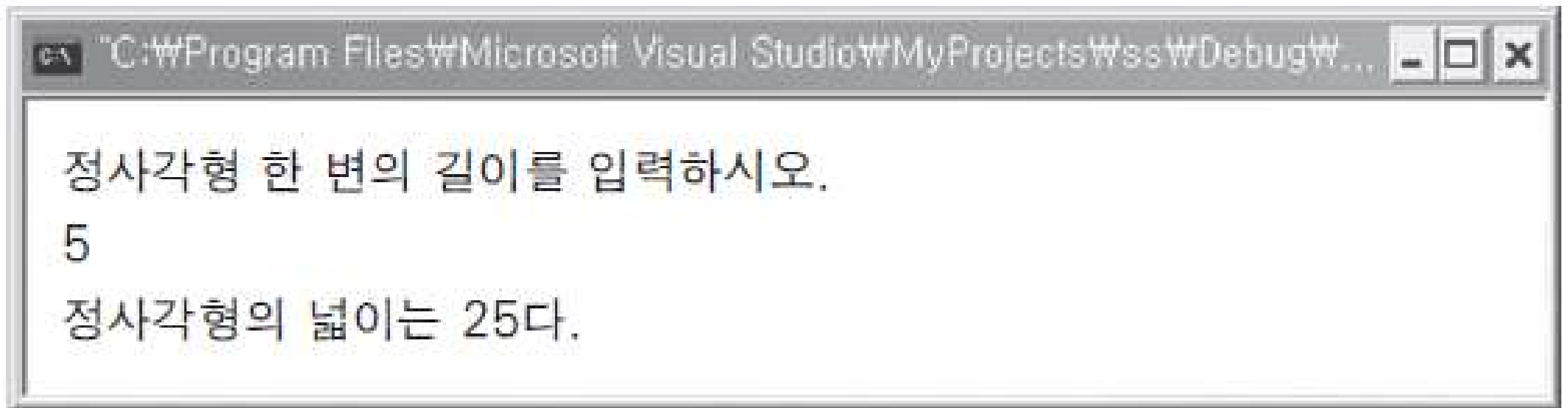
7.     e = a % c;

8.     Console.WriteLine("e = a % c를 실행한 후 e의 값은 {0}\n", e);
9.     g = f / c;

10.    Console.WriteLine("g = f / c를 실행한 후 g의 값은 {0}\n", g);
11.}
```


● C# 프로그래밍 언어

Robot Media Laboratory



```

C:\Program Files\Microsoft Visual Studio\MyProjects\ss\Debug\...
정사각형 한 변의 길이를 입력하시오.
5
정사각형의 넓이는 25다.

```

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2. {
3.     int a = 0;
4.     Console.WriteLine("정사각형의 한변의 길이를 입력하시오:");
5.     a=int.Parse(Console.ReadLine());
6.     Console.WriteLine("정사각형의 넓이:{0}",a*a);
7. }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 관계 연산자

- 두 수 사이의 대소 관계와 특정 조건을 검사할 때 사용하는 연산자
- 관계가 성립되면 참(true 또는 1), 성립되지 않으면 거짓(false 또는 0)으로 표시

연산자	의미	사용법	설명
>	~보다 크다.	$a = (b > c)$	b가 c보다 크면 $a = 1$, 그렇지 않으면 $a = 0$ 이다.
<	~보다 작다.	$a = (b < c)$	b가 c보다 작으면 $a = 1$, 그렇지 않으면 $a = 0$ 이다.
>=	~보다 크거나 같다.	$a = (b >= c)$	b가 c보다 크거나 같으면 $a = 1$, 그렇지 않으면 $a = 0$ 이다.
<=	~보다 작거나 같다.	$a = (b <= c)$	b가 c보다 작거나 같으면 $a = 1$, 그렇지 않으면 $a = 0$ 이다.
==	같다.	$a = (b == c)$	b와 c가 같으면 $a = 1$, 같지 않으면 $a = 0$ 이다.
!=	같지 않다.	$a = (b != c)$	b와 c가 같지 않으면 $a = 1$, 같으면 $a = 0$ 이다.

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int score;
4.         bool grade;

5.         Console.WriteLine("점수를 입력하시오");
6.         score = int.Parse(Console.ReadLine());

7.         grade = score >= 90;
8.         if (grade == true)
9.             Console.WriteLine("합격");
10.        else
11.            Console.WriteLine("불합격");
12.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 논리 연산자

- 조건 여러 개를 결합하여 판정하는 연산자
- AND, OR, NOT의 논리 연산을 수행
- 참(true)일 때는 1로, 거짓(false)일 때는 0으로 결과를 표시

X	Y	AND(&&)	OR()	NOT X(!X)
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

● C# 프로그래밍 언어

Robot Media Laboratory

● 논리 연산자의 종류와 사용법

연산자	의미	사용법	설명
논리곱(&&)	AND	<code>a = b&& c</code>	b와 c가 모두 참이면 <code>a = 1</code> , 아니면 <code>a = 0</code> 이다.
논리합()	OR	<code>a = b c</code>	b와 c가 모두 거짓이면 <code>a = 0</code> , 아니면 <code>a = 1</code> 이다.
논리부정(!)	NOT	<code>a = !b</code>	b가 참이면 <code>a = 0</code> , b가 거짓이면 <code>a = 1</code> 이다.

● 논리 연산자의 우선순위

- 단항 연산자인 `!`이 가장 높고 다음이 `&&`이며 `||`가 가장 낮음.
- `&&`과 `||`이 같이 쓰이면 `&&`이 먼저 수행됨.

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int score;
4.         bool grade;

5.         Console.WriteLine("점수를 입력하시오");
6.         score = int.Parse(Console.ReadLine());

7.         grade = score >= 90 && 100>score;
8.         if (grade == true)
9.             Console.WriteLine("합격");
10.        else
11.            Console.WriteLine("불합격");
12.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 축약 연산자

- 연산 2개를 동시에 수행하여 값을 할당하는 연산자

연산자	사용법	설명
<code>+=</code>	<code>a += b</code>	a와 b를 더하여 a에 대입한다.
<code>-=</code>	<code>a -= b</code>	a에서 b를 뺀 결과 값을 a에 대입한다.
<code>*=</code>	<code>a *= b</code>	a와 b를 곱하여 a에 대입한다.
<code>/=</code>	<code>a /= b</code>	a를 b로 나눈 몫을 a에 대입한다.
<code>%=</code>	<code>a %= b</code>	a를 b로 나눈 나머지를 a에 대입한다.
<code><<=</code>	<code>a <<= b</code>	a를 b비트만큼 왼쪽으로 이동하여 a에 대입한다.
<code>>>=</code>	<code>a >>= b</code>	a를 b비트만큼 오른쪽으로 이동하여 a에 대입한다.
<code>&=</code>	<code>a &= b</code>	a와 b를 비트별로 AND 연산하여 a에 대입한다.
<code> =</code>	<code>a = b</code>	a와 b를 비트별로 OR 연산하여 a에 대입한다.
<code>^=</code>	<code>a ^= b</code>	a와 b를 비트별로 XOR 연산하여 a에 대입한다.

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int a = 10;
4.         Console.WriteLine("초깃값 : a = 10");

5.         a += 5;
6.         Console.WriteLine("a += 5 : {0}", a);

7.         a = 10; a -= 5 + 2;
8.         Console.WriteLine("a -= 5 + 2 : {0}", a);

9.         a = 10; a *= 2 + 5;
10.        Console.WriteLine("a *= 2 + 5 : {0}", a);
11.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 연산자 우선순위

– 단항 > 산술 > 이동 > 관계 > 비트 > 논리 > 조건 > 대입

● 연산자 우선순위의 원칙

- () 안의 내용을 먼저 처리
- 단항 연산자를 이항 연산자보다 먼저 처리
- 증감 연산자는 위치에 따라 처리되는 순서가 다르지만 대개 왼쪽에서 오른쪽으로 수행하는 것이 원칙

● C# 프로그래밍 언어

Robot Media Laboratory

우선순위	연산자의 종류		연산자	결합성
고 ↑ ↓ 저	식, 구조체, 공용체		(), [], .., ->	좌 → 우
	단항 연산자		!, ~, -, +, ++, --, &, *	좌 → 우
	이항 연산자	승제	*, /, %	좌 → 우
		가감	+, -	
		이동	<<, >>	
		비교	<, <=, >, >=	
		등가	==, !=	
		비트 AND	&	
		비트 XOR	^	
		비트 OR		
		논리 AND	&&	
		논리 OR		
	조건 연산자		?:	좌 ← 우
	대입 연산자		=, +=, -=, *=, /=, <<=, >>=, &=	좌 ← 우
	콤마 연산자		,	좌 → 우

● C# 프로그래밍 언어

Robot Media Laboratory

- $-x + 3$ 단항 > 이항
- $x + 3 * y$
- $x + 3 > y - 2$
- $x > 3 \ \&\& \ x < 5$
- `int result = x + y * 3;`
- $x = 2 + 1$
- $x < -1 \ || \ x > 3 \ \&\& \ x < 5$

● C# 프로그래밍 언어

Robot Media Laboratory

- $-x + 3$
- $x + 3 * y$
- $x + 3 > y - 2$
- $x > 3 \ \&\& \ x < 5$
- `int result = x + y * 3;`
- $x = 2 + 1$
- $x < -1 \ || \ x > 3 \ \&\& \ x < 5$

단항 > 이항

곱셈, 나눗셈 > 덧셈, 뺄셈

산술 > 비교

비교 > 논리

산술 > 대입

산술 > 대입

비교 > `&&` > `>||`

● C# 프로그래밍 언어

Robot Media Laboratory

&&, ||, +, -, *, ?:, =, |, <, >, <<, >>, ==, !=, &, ^, >>=, <<=, ^=, %

우선순위	연산자의 종류	연산자	결합성
고 ↑ ↓ 저	식, 구조체, 공용체	(), [], .. ->	좌 → 우
	단항 연산자	!, ~, -, +, ++, --, &, *	좌 → 우
	이항 연산자	승제 *, /, %	좌 → 우
		가감 +, -	
		이동 <<, >>	
		비교 <, <=, >, >=	
		등가 ==, !=	
		비트 AND &	
		비트 XOR ^	
		비트 OR 	
		논리 AND &&	
		논리 OR 	
	조건 연산자	?:	좌 ← 우
	대입 연산자	=, +=, -=, *=, /=, <<=, >>=, &=	좌 ← 우
	콤마 연산자	,	좌 → 우

● C# 프로그래밍 언어

Robot Media Laboratory

`&&, ||, +, -, *, ?:, =, |, <, >, <<, >>, ==, !=, &, ^, >>=, <<=, ^=, %`

1.	산술연산:	* %
2.	산술연산:	+ -
3.	이동연산:	>> <<
4.	비교연산:	< >
5.	비교연산:	!= ==
6.	비트연산:	&
7.	비트연산:	^
8.	비트연산:	
9.	논리연산:	&&
10.	논리연산:	
11.	조건연산:	?:
12.	대입연산:	= >>= <<= ^=

● C# 프로그래밍 언어

Robot Media Laboratory

- `Int x=10`

- `Int y=5`

- `(x>7)&&(y<=5) = ?`

- `(x%3 == 2) || (y%2 != 1) = ?`

- **C# 프로그래밍 언어**

Robot Media Laboratory

- **Int x=10**
- **Int y=20**
- **Int z=(++x)+(y--)**

- **X=?, Y=?, Z=?**

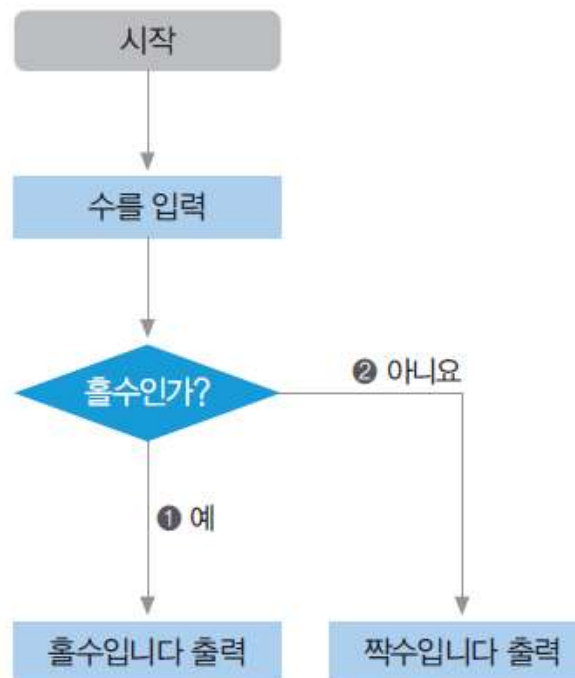
● C# 프로그래밍 언어

Robot Media Laboratory

● 제어문

- 프로그램의 실행을 인위적으로 조절
- 프로그램의 흐름을 제어(이동 또는 반복시킴)

● 제어문의 흐름



● C# 프로그래밍 언어

Robot Media Laboratory

● 제어문의 종류

종류	제어문	보조 제어문
선택문	if if~else if~else if 다중 if문 switch~case	break return
반복문	while for do~while	
무조건 분기문	goto	return

● C# 프로그래밍 언어

Robot Media Laboratory

● if문

- 조건에 따라 그다음 명령 수행을 달리하고자 할 때 사용하는 명령문

● If문 형식

```
if(조건식) 문장;
```

```
if(조건식)  
문장 1;  
else  
문장 2;
```

```
if(조건식 1)  
문장 1;  
else if(조건식 2)  
문장 2;  
else if(조건식 3)  
문장 3;  
:  
else  
문장 4;
```

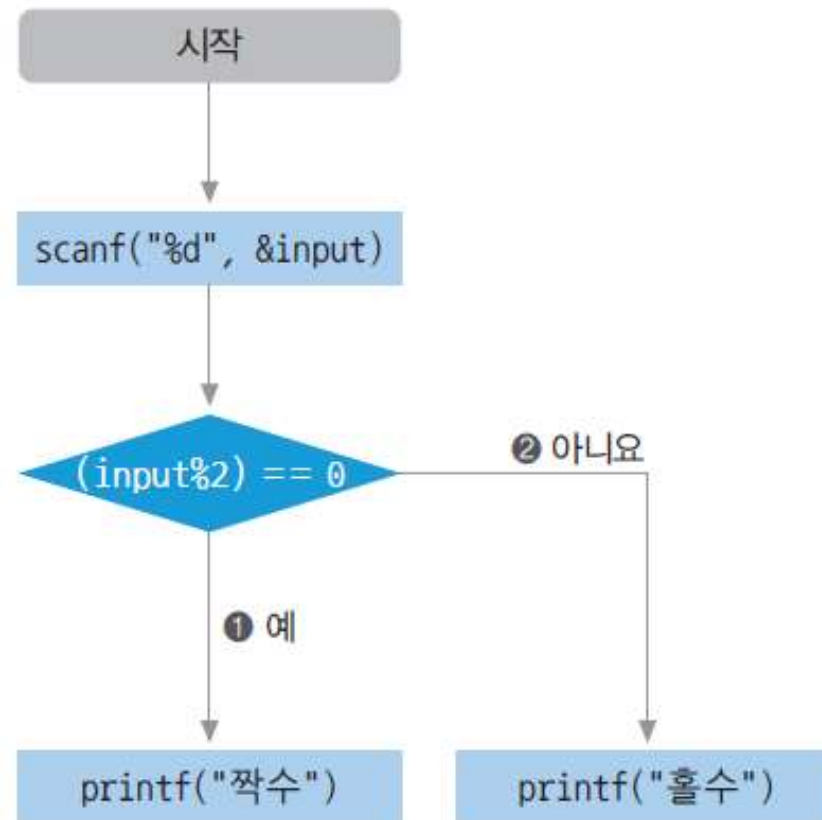
```
if(조건식 1)  
if(조건식 2)  
문장 1;
```

● C# 프로그래밍 언어

Robot Media Laboratory

● if~else문

- 조건식이 참이면 if문 아래 문장을, 거짓이면 else문 아래 문장을 수행



● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int input;
4.         Console.WriteLine("양수를 입력하세요");
5.         input = int.Parse(Console.ReadLine());

6.         if (input % 2 == 0)
7.             Console.WriteLine("입력한 수 {0}는 짝수입니다.", input);
8.         else
9.             Console.WriteLine("입력한 수 {0}는 홀수입니다.", input);
10.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 괄호 사용법

```
if(조건식)
    문장 1;
if(조건식){
    문장 1;
    문장 2;
}
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 괄호를 바르게 사용한 예

```
if(조건식){  
    문장 1;  
    문장 2;  
}
```

- 결과 : 의도대로 조건식이 참이면 문장 1과 문장 2를 수행
참이 아니면 두 문장 모두 수행하지 않음.

● 괄호를 잘못 사용한 예

```
if(조건식)  
    문장 1;  
    문장 2;
```

- 결과 : 조건식이 참이면 문장 1과 2를 수행하지만, 참이 아니어도 문장 2를 수행

- **C# 프로그래밍 언어**

Robot Media Laboratory

- **if~else if문**

- 대등한 조건이 하나가 아니라 여러 개 있는 경우에 사용

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2. {
3.     int jumsu;

4.     Console.WriteLine("100~0점을 입력하세요");
5.     jumsu = int.Parse(Console.ReadLine());

6.     if (jumsu >= 0 && jumsu <= 100)
7.         Console.WriteLine("입력한 점수는 {0}점입니다.", jumsu);
8.     else
9.     {
10.        Console.WriteLine("0점에서 100점 사이를 입력하셔야 합니다.");
11.        return;
12.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

```
1.      if ( jumsu < 60)
2.          Console.WriteLine("{0}점은 F학점입니다.", jumsu);

3.      else if (jumsu < 70)
4.          Console.WriteLine("{0}점은 D학점입니다.", jumsu);

5.      else if (jumsu < 80)
6.          Console.WriteLine("{0}점은 C학점입니다.", jumsu);

7.      else if (jumsu < 90)
8.          Console.WriteLine("{0}점은 B학점입니다.", jumsu);
9.      else
10.         Console.WriteLine("{0}점은 A학점입니다.", jumsu);
11.}
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 다중 if문

- if문 안에서 또 다른 조건식을 검사하고자 할 때 사용

● 다중 if문을 사용한 예와 사용하지 않은 예

```
if (input > = 0){  
    if(input % 2 == 0){  
        Console.WriteLine ("입력한 수 {0}는 양수이며 짝수입니다.", input);  
    }  
    else  
        Console.WriteLine("입력한 수 {0}는 양수이며 홀수입니다.", input);  
}
```

```
if(input > = 0 && input % 2 == 0)  
    Console.WriteLine("입력한 수 {0}는 양수이며 짝수입니다.", input);  
  
else if(input > = 0 && input % 2 != 0)  
    Console.WriteLine("입력한 수 {0}는 양수이며 홀수입니다.", input);  
  
else  
    Console.WriteLine("입력한 수 {0}는 음수입니다.",input);
```

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int input;
4.         Console.WriteLine("100~0점을 입력하세요");
5.         input = int.Parse(Console.ReadLine());
6.         if (input >= 0)
7.         {
8.             if (input % 2 == 0)
9.                 Console.WriteLine("입력한 수 {0}는 짝수입니다.", input);
10.            else
11.                Console.WriteLine("입력한 수 {0}는 홀수입니다.", input);
12.        }
13.        else
14.        {
15.            Console.WriteLine("입력한 수 {0}는 음수입니다.", input);
16.        }
17.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

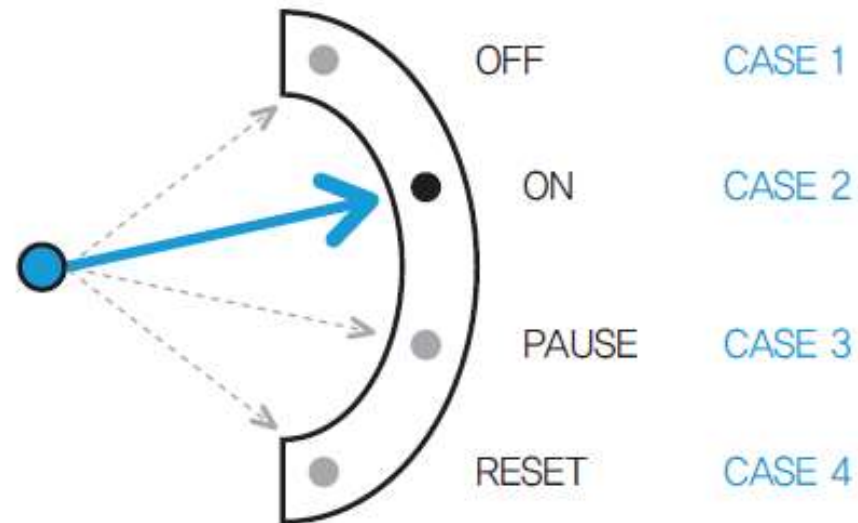
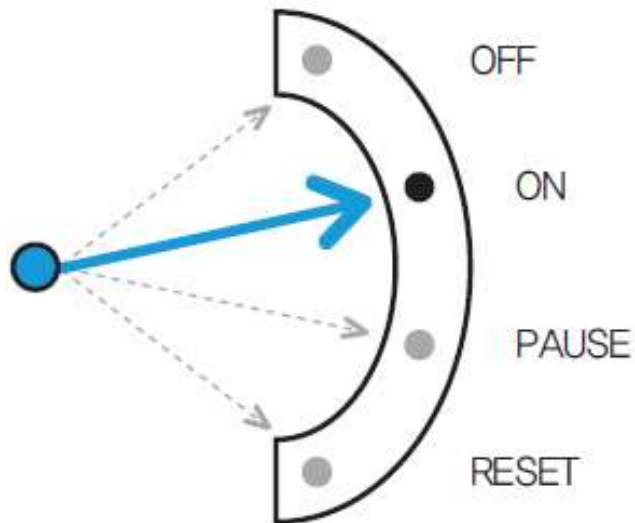
```
1. static void Main(string[] args)
2.     {
3.         int input;
4.         Console.WriteLine("100~0점을 입력하세요");
5.         input = int.Parse(Console.ReadLine());
6.         if (input >= 0 && input % 2 == 0)
7.             Console.WriteLine("입력한 수 {0}는 짝수입니다.", input);
8.         else if (input >= 0 && input % 2 != 0)
9.             Console.WriteLine("입력한 수 {0}는 홀수입니다.", input);
10.        else
11.            Console.WriteLine("입력한 수 {0}는 음수입니다.", input);
12.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● switch~case문

- 조건식을 먼저 평가한 뒤 그 식의 값이 case 상수와 일치하는 쪽으로 분기하여 해당 명령문을 수행



● C# 프로그래밍 언어

Robot Media Laboratory

● Switch문 형식

```
switch(조건식)
{
    case 상숫값 1 :
        명령문 블록 1
        break;
    case 상숫값 2 :
        명령문 블록 2
        break;
    :
    default :
        명령문 블록 n
        break;
}
```


● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int input;
4.         Console.WriteLine("영어로 인사하는 법을 배우겠습니다.\n아침 인사는 1번을 누르세요.\n점심 인사는 2번을 누르\n세요.\n저녁 인사는 3번을 누르세요.\n");
5.         input = int.Parse(Console.ReadLine());
6.         switch (input)
7.         {
8.             case 1:
9.                 Console.WriteLine("Good Morning!!");
10.                break;
11.            case 2:
12.                Console.WriteLine("Good Afternoon!!");
13.                break;
14.            case 3:
15.                Console.WriteLine("Good Night!!");
16.                break;
17.        }
18.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● Switch문 형식

[1을 입력했을 때]

```
"C:\Program Files\Microsoft Visual Studio\MyProje"
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
1
Good Morning!!
```

[2를 입력했을 때]

```
"C:\Program Files\Microsoft Visual Studio\MyProje"
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
2
Good Afternoon!!
```

[3을 입력했을 때]

```
"C:\Program Files\Microsoft Visual Studio\MyProje"
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
3
Good Night!!
```

[4를 입력했을 때]

```
"C:\Program Files\Microsoft Visual Studio\MyProje"
영어로 인사하는 법을 배우겠습니다.
아침 인사는 1번을 누르세요.
점심 인사는 2번을 누르세요.
저녁 인사는 3번을 누르세요.
4
```

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int input;
4.         Console.WriteLine("영어로 인사하는 법을 배우겠습니다.아침 인사는 1번을 누르세요.\n점심 인사는 2번을 누르세요.\n저녁 인
   사는 3번을 누르세요.\n");
5.         input = int.Parse(Console.ReadLine());
6.         switch (input)
7.         {
8.             case 1:
9.                 Console.WriteLine("Good Morning!!");
10.                break;
11.             case 2:
12.                 Console.WriteLine("Good Afternoon!!");
13.                break;
14.             case 3:
15.                 Console.WriteLine("Good Night!!");
16.                break;
17.             default:
18.                 Console.WriteLine("입력 불가능");
19.                break;
20.         }
21.     }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● switch~case문의 조건식

- 반드시 수식이나 값을 넣어야 함.

● 조건식을 잘못 사용한 예

```
switch(input > 0)
{
    :
}
```

● 조건식을 바르게 사용한 예

```
switch(input)
{
    :
}
```

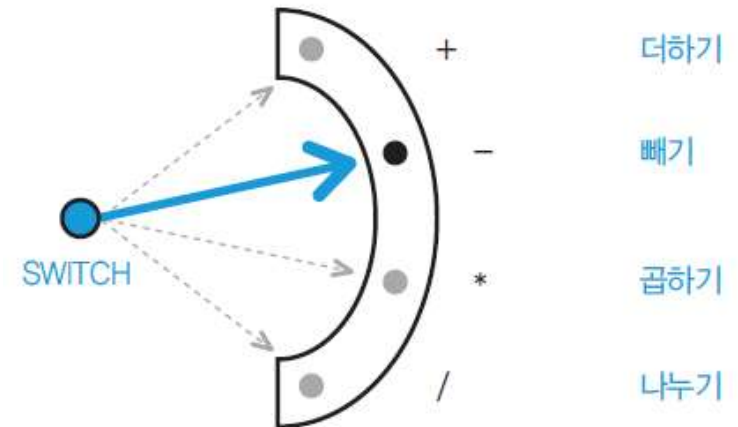
● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int score;
4.         Console.WriteLine("topcit예상점수를 적으시오");
5.         score = int.Parse(Console.ReadLine());
6.         switch (score / 10)
7.         {
8.             case 10:
9.             case 9:
10.                Console.WriteLine("점수는 {0}점이고 성적은 {1}입니다.", score, 'A'); break;
11.
12.            case 8:
13.                Console.WriteLine("점수는 {0}점이고 성적은 {1}입니다.", score, 'B'); break;
14.
15.            case 7:
16.                Console.WriteLine("점수는 {0}점이고 성적은 {1}입니다.", score, 'C'); break;
17.
18.            case 6:
19.                Console.WriteLine("점수는 {0}점이고 성적은 {1}입니다.", score, 'D'); break;
20.
21.            default: Console.WriteLine("점수는 {0}점이고 성적은 {1}입니다.", score, 'F'); break;
22.        }
23.     }
```

● C# 프로그래밍 언어

Robot Media Laboratory



● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int num1, num2;
4.         Char arithmetic;
5.         Console.WriteLine("숫자 2개와 연산자를 입력하세요.");
6.
7.         Console.WriteLine("연산자는 아래와 같습니다.");
8.         Console.WriteLine("더하기 : +, 빼기 : -, 곱하기 : *, 나누기 : /");
9.         Console.WriteLine("입력 순서 : 정수1 Wn연산자Wn 정수2");
10.        num1 = int.Parse(Console.ReadLine());
11.        arithmetic = Char.Parse(Console.ReadLine());
12.        num2 = int.Parse(Console.ReadLine());
13.
14.        switch (arithmetic)
15.        {
16.            case '+':
17.                Console.WriteLine("{0} {1} {2} = {3}", num1, arithmetic, num2, num1 + num2);
18.                break;
19.            case '-':
20.                Console.WriteLine("{0} {1} {2} = {3}", num1, arithmetic, num2, num1 - num2);
21.                break;
22.            case '*':
23.                Console.WriteLine("{0} {1} {2} = {3}", num1, arithmetic, num2, num1 * num2);
24.                break;
25.            case '/':
26.                Console.WriteLine("{0} {1} {2} = {3}", num1, arithmetic, num2, num1 / num2);
27.                break;
28.            default:
29.                Console.WriteLine("연산자 입력이 잘못 되었습니다. Wn다시입력해주세요. ");
30.                break;
31.        }
32.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 조건 연산자

- C 언어에서 유일하게 피연산자가 3개인 삼항 연산자
- ?와 : 연산자를 이용하여 프로그램을 제어

● 조건 연산자의 작성 형식

(문장 1) ? (문장 2) : (문장 3);

- 문장 1의 내용이 참이면 문장 2를 수행하고, 거짓이면 문장 3을 수행

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {

3.         int x=10, y=20;
4.         int max, min;
5.         max = (x > y) ? x : y;
6.         min = (x > y) ? y : x;
7.         Console.WriteLine("두 수 {0}과 {1} 중에 큰 수는 {2}이다.",x,y,max);
8.         Console.WriteLine("두 수 {0}과 {1} 중에 작은 수는 {2}이다.", x, y, min);
9.
10.    }
```

● C# 프로그래밍 언어

Robot Media Laboratory

● 조건 연산자의 예

- x 값이 y 값보다 크면 결과 값이 1이 되고, 그렇지 않으면 결과 값이 2가 됨.

```
( x > y ) ? 1 : 2;
```

- 두 수 중 큰 값을 알아내는 프로그램

```
max = ( x > y ) ? x : y;
```

- 두 수 중 작은 값을 찾는 예

```
min = ( x > y ) ? y : x;
```

- x 값이 0이 아니면 value는 1이 되고, x 값이 0이면 value는 0이 됨.

```
value = ( x == 0 ) ? 0 : 1;
```

● C# 프로그래밍 언어

Robot Media Laboratory

```
1. static void Main(string[] args)
2.     {
3.         int x,y;
4.         int max,min;
5.         x = int.Parse(Console.ReadLine());
6.         y = int.Parse(Console.ReadLine());
7.         if (x > y)
8.         {
9.             max = x;
10.            min = y;
11.        }
12.        else
13.        {
14.            max = y;
15.            min = x;
16.        }
17.        Console.WriteLine("두 수 {0}와 {1} 중 큰 수는 {2}이다.", x, y, max);
18.        Console.WriteLine("두 수 {0}와 {1} 중 작은 수는 {2}이다.", x, y, min);
19.    }
20. }
```