

정보처리기사 합격!
그 곳, 두목넷

2021년도

필답형 실기

정보처리기사

10. 프로그래밍 언어 활용

dumok.net

NCS **필답형 실기**

정보처리기사

10. 프로그래밍 언어 활용

수행
준거

응용 소프트웨어 개발에 사용되는 프로그래밍 언어의 기초 문법을 적용하고, 언어의 특징과 라이브러리를 활용하여 기본 응용 소프트웨어를 구현할 수 있다.

NCS

필답형 실기

정보처리기사

[필기] 4과목
프로그래밍 언어 활용

[실기] 08.
SQL 응용

[개편 전] 2017~2019
실기 "코드" 기출문제

10. 프로그래밍 언어 활용

C언어

C++언어

JAVA언어

Python

dumok.net

3

NCS

필답형 실기

정보처리기사

10. 프로그래밍 언어 활용

2020년 1회	2020년 2회	2020년 3회	2020년 4 · 5회
<ul style="list-style-type: none">• [C언어] 버블정렬 : 오름차순• [JAVA] 1차원 배열 : 요소 출력• [JAVA] switch ~ case	<ul style="list-style-type: none">• [JAVA] new• [JAVA] a=10• [Python] SET Type : 출력결과	<ul style="list-style-type: none">• [C언어] 누승 초기값 0• [C언어] 사용자함수 호출• [JAVA] 객수판별 합계• [JAVA] 다형성 : 상속• [C++] 생성자 (약술형)	<ul style="list-style-type: none">• [C언어] 포인터 변수와 포인터 연산자• [C언어] 2차원배열 크기• [JAVA] 진법변환 : 10진수 → 2진수• [JAVA] 오버라이드 및 재귀호출• [Python] LIST 중첩
3문제/20문제	3문제/20문제	5문제/20문제	5문제/20문제

C언어 JAVA Python C++

4



10. 프로그래밍 언어 활용



1. 기본문법 활용하기

Level 3



C언어 기본문법 활용하기



JAVA 기본문법 활용하기



Python 기본문법 활용하기

dumok.net



NCS

학습목표

산업인력공단

출제 기준(세세항목)

1. 기본문법 활용하기

기본문법 활용

- 응용 소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다.
- 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다.
- 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.

언어 특성 활용

- 프로그래밍 언어별 특성을 파악하고 설명할 수 있다.
- 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다.
- 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용할 수 있다.

라이브러리 활용

- 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다.
- 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성할 수 있다.
- 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.



NCS 산업인력공단
학습목표 출제 기준(세세항목)

1. 기본문법 활용하기

Section 01	C언어와 JAVA언어의 기본문법구조	Section 08	배열과 문자열
Section 02	데이터 타입	Section 09	C언어 포인터
Section 03	변수와 상수	Section 10	C언어 사용자 정의 함수
Section 04	연산자	Section 11	JAVA 클래스와 메소드
Section 05	데이터 입력 및 출력	Section 12	JAVA 상속
Section 06	제어문(1) - 선택문	Section 13	예외 처리
Section 07	제어문(2) - 반복문	Section 14	PYTHON에 대한 이해

C언어 JAVA Python C++

PYTHON 특강 강의 수강요망!

dumok.net



학습내용

Section 01

C언어와 JAVA언어의 기본문법 구조

프로그래밍 언어

소스(코드) 파일

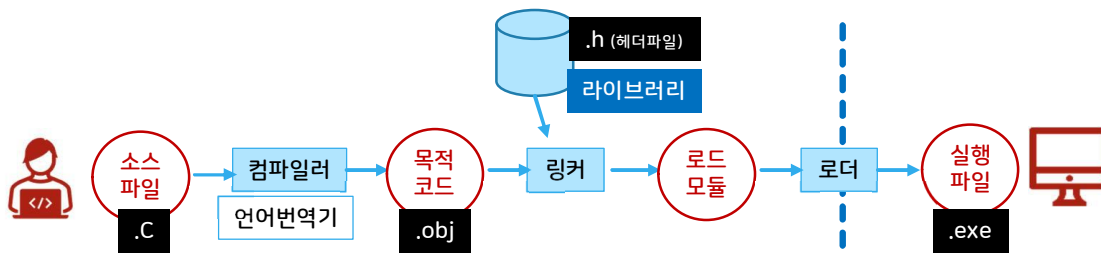
C언어 → 파일명.c
C++ → 파일명.cpp
JAVA → 파일명.java
Python → 파일명.py

dumok.net

01 C언어와 JAVA언어의 개요

① C언어

- 1972년 미국 벨 연구소의 데니스 리치에 의해 개발된 시스템 기술용 언어이다.
- 유닉스(UNIX) 운영체제 개발에 사용할 목적으로 만들어졌다.
- 특징
 - 논리적이며 구조적인 시스템 프로그래밍 언어이다.
 - 하드웨어 제어가 가능하며 프로그램 이식성이 높다.
 - 간략한 문법 표현으로 함축적인 프로그램 작성이 용이하다.
 - 효율성과 유연성을 갖춘 저급 언어 특성을 가진 고급 언어이다.



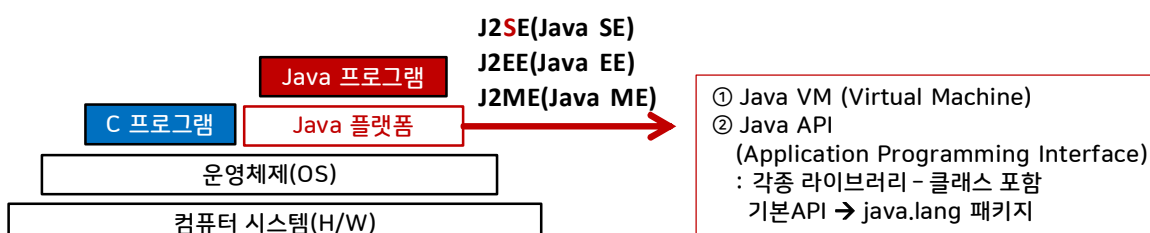
dumok.net

01 C언어와 JAVA언어의 개요

② JAVA언어

- 1996년 미국 썬 마이크로시스템즈가 발표한 객체 지향 프로그래밍 언어이다.
- JAVA는 프로그래밍 언어와 함께 실행 환경(플랫폼)을 포함하고 있다.
- 특징
 - 플랫폼 독립적이다.
 - 상속을 지원하는 객체 지향 프로그래밍 언어(Object-Oriented PL)이다.
 - 응용 프로그래밍과 웹 프로그래밍(애플릿)이 가능하다.
 - 예외 처리와 멀티 스래싱을 지원한다.

• 플랫폼(platform)
: 프로그램이 동작할수 있는
실행환경





② JAVA언어의 print()와 println() 메소드

출	Java 프로그램 소스 HelloEx002.java
1	public class HelloEx002 {
2	public static void main(String[] args) {
3	System.out.print(100);
4	System.out.println(100);
5	System.out.println("합격점수 : " + 100);
6	}
7	}
결과	100100 합격점수 : 100

JAVA의 출력 메소드

System.out.print("A"); System.out.print("B");	System.out.println("A"); System.out.println("B");	System.out.printf("%s\n%s", "A", "B");
AB	A B 	A B

dumok.net



③ JAVA언어의 println() 메소드의 예제

주의 System.out.println("x+y = " + x + y);

PassDNA 정보처리기사 필기

2021년 1회 : 4과목 프로그래밍 언어 활용

JAVA : 출력 메소드

프로그래밍 언어 활용
> 언어특성 활용

70. 다음 **JAVA** 코드 출력문의 결과는?

```

..생략..
System.out.println( " 5 + 2  " + 3 + 4);
System.out.println( " 5 + 2  " + (3 + 4));
..생략..
    
```

④ 5 + 2 = 34
5 + 2 = 7





05 프로그래밍 언어의 기본 데이터 타입(Primitive Type) 및 크기

- 자료형의 예약어는 모두 소문자이다.
- 문자열형(String)은 기본 데이터 타입이 아니다.
- C++언어의 논리형의 예약어는 bool이다.
- Python의 long형은 무한크기이다.
- C#언어의 실수형에는 decimal(16byte)이 있다.

C++

```
#include <iostream>
#include <string>
using namespace std;
int main() {
    string a = "Hello";
    return 0;
}
```

JAVA

```
#import java.lang.String;
public class Test {
    public static void main(String[] args) {
        String a = "Hello";
    }
}
```

dumok.net



변수의 4 요소

- 변수 : 프로그램 실행 중, 변경할 수 있는 값이 저장되는 기억공간(=메모리)

선언문

① 주소(Address)

② 이름(Name)

③ 자료형(Data Type)

④ 값(Value)

대입문

바인딩(Binding)

When

Where

- 정적 바인딩 : 이온시간 바인딩, 번역시간, 데이터영역/스택영역
- 동적 바인딩 : 늦은시간 바인딩, 실행시간, 힙영역

★ 변수의 4요소

- 주소(address)
- 이름(name)
- 자료형(datatype)
- 값(value)

★ 바인딩(binding)

- 어떤 변수의 명칭과 그 메모리 주소, 데이터형 또는 실제 값을 연결하는 것이다.
- 변수들이 갖는 속성이 완전히 결정되는 시간을 바인딩 시간 (Binding Time)이라고 한다.
- 바인딩의 종류 : 정적 바인딩(번역시간), 동적 바인딩(실행시간)

주 기억장치

② C언어의 변수명 정의 규칙(Naming Rule)

- 변수명은 식별자이므로 영문 대소문자, 숫자를 혼합하여 명명한다.
- 대소문자 구별하며, 변수명으로 예약어를 사용하는 것을 금지한다.
- 밑줄(_)을 제외한 모든 특수문자의 사용을 금지한다.
 - \$a : C, C++, Python 변수명 불가능
 - \$a : JAVA 변수명 가능
- 숫자로 시작하는 것을 금지한다.
- 데이터의 의미나 역할을 표현할 수 있는 이름으로 명명하는 것이 프로그램의 가독성과 유지보수성 증가를 위해 바람직하다.



my_studnet

파일명



myStudnet

변수명, 필드명



MyStudnet

클래스명



헝가리안 표기법
nMystudnet

[기출]2020년 3회

- 프로그래밍 언어에서 변수 및 함수의 인자 이름 앞에 데이터 타입을 명시하는 코딩 규칙
- 소문자 변수타입 + 단어(첫글자만 대문자)

dumok.net

01 연산자(operator)와 우선순위

- 연산자는 자료에 대한 연산동작을 지정한 기호이다.
- 연산자 우선순위란 두 종류 이상의 연산자가 수식 내에 포함될 경우 연산의 순서를 의미한다.
- 연산자 결합 방향(결합성)이란 우선순위가 동일한 연산자들이 수식 내에 포함될 경우 어느 방향으로 결합하는가를 결정하는 것이다.





③ 예제 2

```
#include <stdio.h>
int main( )
{
    int x, y;
    x = 10;
    y = ++x;
    printf( " x : %d \n" , x);
    printf( " y : %d \n" , y);
    x = 10;
    y = x++;
    printf( " x : %d \n" , x);
    printf( " y : %d \n" , y);
    return 0;
}
```

결과

```
x : 11
y : 11
x : 11
y : 10
```

dumok.net



• 비트 이동(시프트)연산자

```
#include <stdio.h>
int main( )
{
    int a, b;
    a = 4;
    b = 2;
    printf( " 4 << 2 : %d\n" , a << b);
    printf( " 4 >> 2 : %d\n" , a >> b);
    return 0;
}
```

결과

```
4 << 2 : 16
4 >> 2 : 1
```



• 비트 논리 연산자

```
#include <stdio.h>
int main( )
{
    int a, b;
    a = 5;
    b = 2;
    printf( "논리곱 : %d\n", a & b);
    printf( "배타적 논리합 : %d\n", a ^ b);
    printf( "논리합 : %d\n", a | b);
    printf( "논리부정 : %d\n", ~a);
    return 0;
}
```

결과

```
논리곱 : 0
배타적 논리합 : 7
논리합 : 7
논리부정 : -6
```

dumok.net



• 형변환(캐스팅) 연산자

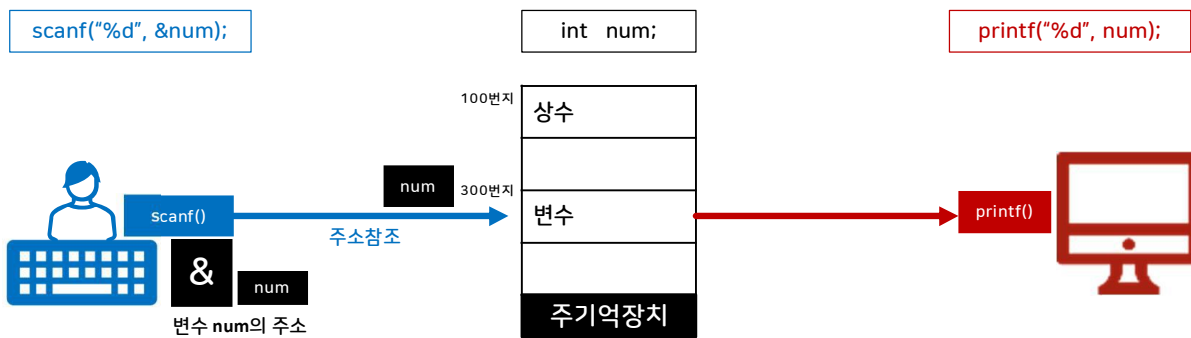
```
#include <stdio.h>
int main( )
{
    int x, y;
    x = 5;
    y = 2;
    printf( "x / y : %d\n", x / y);
    printf( "x / y : %f\n", (float)x / (float)y);
    return 0;
}
```

결과

```
x / y : 2
x / y : 2.500000
```

01 데이터 입 · 출력이란?

- 프로그램을 수행하는 과정에서 메모리로 데이터를 입력하고, 메모리에서 데이터를 출력하도록 하는 기법이다.
- 읽기(Read)와 쓰기(Write) 명령을 이용하여 데이터 전송 요청을 할 수 있다.



dumok.net

콘솔 화면으로 "문자열" 출력

C언어

```
#include <stdio.h>
void main( )
{
    printf("Hello, C!");
}
```

C++언어

```
#include <iostream>
int main( )
{
    std::cout << "Hello, C++!";
    return 0;
}
```

```
#include <iostream>
using namespace std;
int main( )
{
    cout << "Hello, C++!";
    return 0;
}
```

JAVA언어

```
public class Test {
    public static void main(String[] args) {
        System.out.print("Hello, JAVA!");
    }
}
```

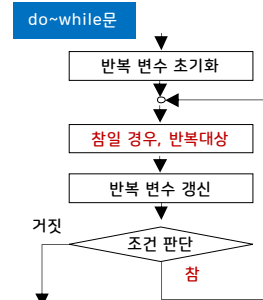
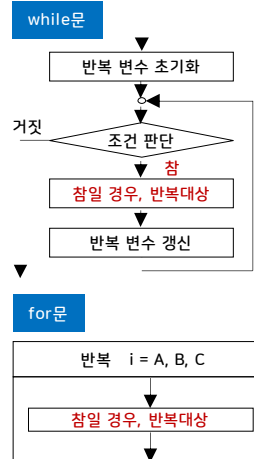
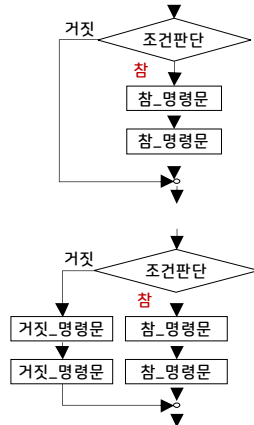
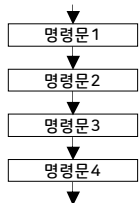
Python

```
>>> print("Hello, Python!")
```

제어문과 순서도

- 순서도(Flowchart)는 알고리즘을 표현한 것으로 어떠한 일을 처리하는 과정을 순서대로 정해진 기호와 도형으로 도식화하여 보여주는 다이어그램이다.

순차구조	선택구조	반복구조
<pre>int a; // 선언문 a = 10 + 20; // 대입문 printf("%d", a);</pre>	① if문 / if ~ else문 ② switch~case문	① while문 ② do~while문 ③ for문



dumok.net

개정 NCS 필기 기출문제

PassDNA 정보처리기사 필기

2021년 1회 : 4과목 프로그래밍 언어 활용

파이썬 : 반복 제어문

프로그래밍 언어 활용 활용 > 기본문법 활용

79. 다음은 파이썬으로 만들어진 반복문 코드이다. 이 코드의 결과는?

```
>> while(True) :
    print( ' A ' )
    print( ' B ' )
    print( ' C ' )
    continue
    print( ' D ' )
```

① A, B, C 출력이 반복된다.



• goto문 (C언어)

```
#include <stdio.h>
void main( )
{
    int num = 0;
START:
    num++;
    printf( "%d\n" , num);
    if( num == 5 )
        goto END;
    else
        goto START;
END:
    printf( "종료\n" );
}
```

결과

```
1
2
3
4
5
종료
```



dumok.net



② String 클래스의 주요 메소드

char charAt(int index) char charAt(int index)	인덱스 위치의 문자 하나 리턴
boolean equals(Object obj)	다른 문자열 객체와 비교 boolean equals(Object obj)
String replace(char oldChar, char newChar)	특정 문자를 새로운 문자로 치환
static String valueOf(para)	숫자값을 문자형으로 처리 static String valueOf(para)
int length()	문자열의 길이(널문자 제외)



String 클래스의 주요 메서드

- 객체지향언어의 장점 : 클래스가 제공하는 메서드(연관된 기능)이용

또는

```
import java.lang.*;
```

```
import java.lang.String;
```

주요 메서드	기능 설명
char charAt (intdex)	인덱스 위치의 문자 하나 리턴
boolean equals (Oobject obj)	다른 문자열 객체와 비교
String replace (char oldChar, char newChar)	특정 문자를 새로운 문자로 치환
static String valueOf (para)	숫자값을 문자형으로 처리
int length ()	문자열의 길이 (널문자 제외)

주의 `int[] num = { 1, 2, 3, 4, 5 }`
`System.out.println("num배열의 크기 : " + num.length);`

dumok.net



[개정 NCS 기출문제] 2020년 4 · 5회 정보처리기사 실기

기출문제 08

10. 프로그래밍 언어 활용

배점 05

다음은 C언어로 작성된 프로그램이다. 이를 실행한 출력 결과를 쓰시오.

```
#include <stdio.h>

void main()
{
    char *p = "KOREA";

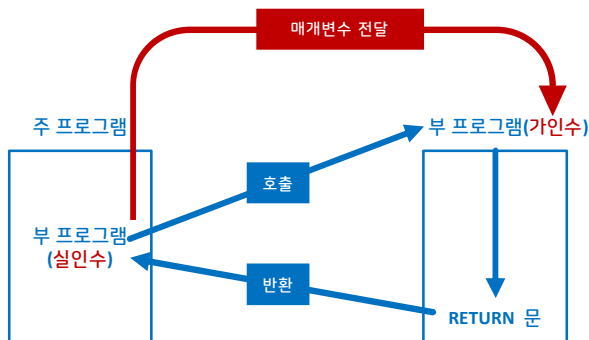
    printf("%s\n", p);
    printf("%s\n", p+3);
    printf("%c\n", *p);
    printf("%c\n", *(p+3));
    printf("%c\n", *p+2);
}
```



--	--	--	--	--	--	--

결과

KOREA
EA
K
E
M



함수의 유형	실인수 = 가인수	반환값	주프로그램 > 함수호출문; 호출	부프로그램 > 함수선언문;
유형1	없음	없음	add();	void add();
유형2	없음	있음	result = add();	int add();
유형3	있음	없음	add(10, 20);	void add(int x, int y);
유형4	있음	있음	result = add(10, 20);	int add(int x, int y);

dumok.net



학습내용

Section 11

JAVA 클래스와 메소드

C언어

구조체

```
struct person {
    char* name;
    int age;
};
```

C++

클래스

```
class Person {
private:
    char* name;
    int age;
public:
    Person();
    void show();
};
```

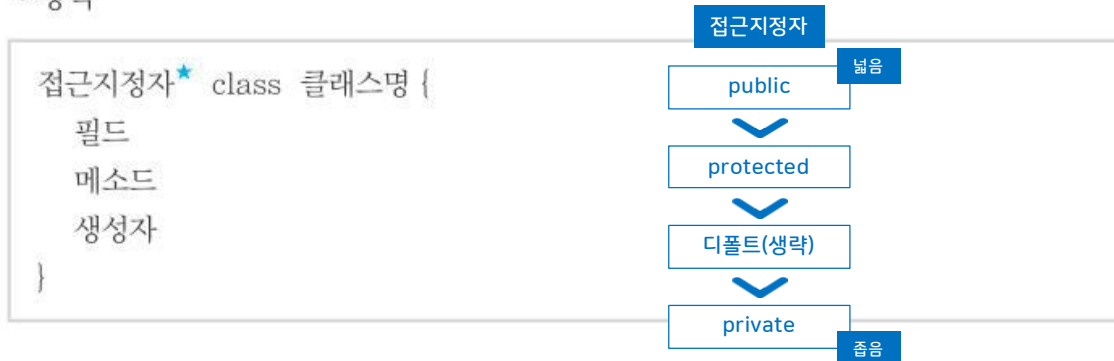
JAVA

클래스

```
class Person {
    private String name;
    private int age;
    Person() { }
    public void show() {
        System.out.println(name+ age);
    }
}
```

- JAVA 소스파일(.java)에는 여러 개의 class 작성이 가능하지만 public class는 한 개만 작성 가능하다. public class가 있을 경우에는 반드시 클래스명을 파일명으로 지정한다.
- main method는 실행의 시작을 위해 반드시 필요하며 실행을 시작하는 public class 내에 작성을 하고 main method가 있는 class명으로 파일명을 지정해야만 실행이 가능하다.
- 형식

정보은닉 (= 접근제어자, Access Modifier)



★ 접근지정자(Access Modifier)

- JAVA언어에서는 클래스와 클래스 멤버에 접근 지정자를 두어 다른 클래스에서 접근하여 사용할 수 있는지 여부를 지정한다. 4가지 접근지정자는 공개 범위에 따라 private < 디폴트(생략) < protected < public 순으로 공개범위가 넓다.
- class 앞에는 접근지정자 중 public 또는 디폴트(생략) 두 가지 중 한 가지만 사용 가능하다.

dumok.net

JAVA의 상속



부모클래스

```
class A {
    public int a;
    void show() {
        System.out.println("a=" + a);
    }
}
```



자식클래스

```
class B extends A {
}
```

실행클래스

```
public class Main{
    public static void main(String[] args) {
        A obj1 = new A();
        obj1.show();
        B obj2 = new B();
        obj2.show();
    }
}
```

결과

```
a=0
a=0
```

C++의 상속

```
#include <iostream>
using namespace std;
```

```
class A {
    public:
        int a;
        void show() {
            cout << "a=" << a << endl;
        }
};
```

```
class B :public A {
};
```

```
int main()
{
    A obj1;
    obj1.show();
    B obj2;
    obj2.show();
    return 0;
}
```

```
a=0
a=0
```



[개정 기출 문제] 2020년 3회 실기

PassDNA 정보처리기사 실기

2020년 3회 정보처리기사

기출 문제 13

10. 프로그래밍 언어 활용

배점

05

약술형

C++ 언어의 생성자(Constructor)에 대해 간략히 설명하시오.

- 답: 객체 생성시 자동으로 호출되는 메소드로
멤버를 초기화하는 목적으로 주로 사용된다.

C++ 언어의 생성자(Constructor)

C++ 언어의 생성자(Constructor)는 클래스의 객체가 인스턴스화 될 때 자동으로 호출되는 멤버함수이다. 일반적으로 클래스의 멤버 변수를 초기화하거나 클래스를 사용하는데 필요한 사전 설정하는 기능을 한다.

- 생성자의 특징
 - 객체 생성시 최초 한 번만 자동호출 된다.
 - 생성자의 이름은 클래스 이름과 같다.
 - 반환형(Return type)은 선언되지 않는다.
 - 오버로딩(Overloading) 가능하다.
 - 매개변수의 디폴트 값을 설정 가능하다.
 - 정의하지 않으면 기본 생성자를 제공한다.

C++ 언어의 소멸자(Destructor)

C++ 언어의 소멸자(Destructor)는 객체가 소멸될 때 자동으로 호출되는 멤버함수이다. 소멸자는 동적 메모리(Heap Memory)에 생성된 객체를 제거하기 위한 작업 수행 시 활용한다.

- 소멸자의 특징
 - 객체 소멸시 최초 한 번만 자동호출 된다.
 - 소멸자의 이름은 ~클래스 이름과 같다.
 - 반환형(Return type)은 선언되지 않는다.
 - 오버로딩(Overloading) 불가능하며 클래스에 단 하나 존재한다.
 - 정의하지 않으면 기본 소멸자를 제공한다.

연습란

※ 다음 여백은 연습란으로 사용하시기 바랍니다.

Pass DNA 정보처리기사

프로그래밍 언어 활용

33

dumok.net



JAVA의 생성자

- 패키지 선언문: 자바의 클래스는 반드시 하나의 패키지에 포함되어야 한다.
- 패키지 선언문이 생략되는 경우 unnamed package에 포함되어 번역된다.
- 디폴트(기본) 생성자는 생성자내에 명령문이 없을 경우 생략될 수 있다.

클래스

```
package net.dumok;
class AAA {
    private String name;    // 필드
    private int age;        // 필드
    public void show() {    // 메서드
        System.out.println(age + "세 " + name);
    }
    AAA() {                // 생성자
        name = "강희영";
        age = 20;
    }
}
```

실행클래스

```
public class Main {
    public static void main(String[] args) {
        AAA obj = new AAA();
        obj.show();
    }
}
```

객체 생성과 동시에 생성자 자동호출

결과

20세 강희영



JAVA의 생성자

자식객체 생성과 동시에 생성자(부모생성자→자식생성자) 자동호출



부모클래스

```
package net.dumok;
class AAA {
    AAA() {                // 부모 생성자
        System.out.println("AAA생성자");
    }
}
```



자식클래스

```
class BBB extends AAA {
    BBB() {                // 자식 생성자
        System.out.println("BBB생성자");
    }
}
```

실행클래스

```
public class Main {
    public static void main(String[] args) {
        BBB obj = new BBB();    // 자식객체 생성
    }
}
```

결과

```
AAA생성자
BBB생성자
```

