

개념 정보처리실기

실무 알고리즘 응용

정보처리실기

합격을 위한
최소한의 JAVA

정보처리

PASS DNA

개념 정보처리실기
최소한의 JAVA

알고리즘 in 정보처리실기

PL 개념용어

APP 개발모델

필답형
주관식
실기

≠ 코딩

순서도

C언어

JAVA

두목넷 :: PASS DNA

정보
처리

필답형
주관식실기

알고리즘

≠ 코딩(Coding)

필기와 실기를 한번에 합격!

두목넷 :: PASS DNA

정보
처리

필답형
주관식실기

알고리즘

= 순서도 = C언어 = JAVA언어

필기와 실기를 한번에 합격!

두목넷 :: PASS DNA
정보 처리

필 답 형
주관식 실기

우리는 어떻게 합격 할 것인가?

정보처리실기
합격을 위한 **JAVA**

칼기와 실기를 한번에 합격!

개정 정보처리실기
최소한의 JAVA

:00:

**정보처리실기 합격을 위한
최소한의 JAVA**

개정 정보처리실기
최소한의 JAVA

:00:

**정보처리실기 합격을 위한
최소한의 JAVA**

with **C**

with 객체지향
프로그래밍

개정 정보처리실기
최소한의 JAVA

JAVA

vs C

- Java를 이용한 기본 프로그래밍 기법
- Java의 객체지향 특징을 활용한 프로그래밍

JAVA

- 1996년 미국 선 마이크로시스템즈(Sun Microsystems)가 발표한 객체지향 프로그래밍 언어이다.
- JAVA는 프로그래밍 언어와 함께 실행환경(플랫폼)을 포함하고 있다.
- 특징
 - > 플랫폼 독립적이다.
 - > 상속을 지원하는 객체지향 프로그래밍 언어이다.(Object-Oriented PL)
 - > 응용 프로그래밍과 웹 프로그래밍(애플릿)이 가능하다.
 - > 변수의 자료형의 불일치 오류의 소지를 많이 줄였다.
 - > 예외처리와 멀티 스레딩을 지원한다.

최대한의 JAVA

- | | |
|----------------------|-----------------------|
| 01. JAVA 언어 소개 | 09. AWT 컨트롤 클래스 |
| 02. JAVA 기본 문법 | 10. AWT 이벤트 |
| 03. 객체지향 프로그래밍 | 11. AWT Container 클래스 |
| 04. 패키지와 예외처리 | 12. Applet 프로그래밍 |
| 05. java.lang 패키지 | 13. JDBC 프로그래밍 |
| 06. 멀티 스레드 프로그래밍 | 14. 네트워크 프로그래밍 |
| 07. File 클래스와 IO 스트림 | |
| 08. AWT 프로그래밍 | |

최대한의 JAVA

- | | |
|----------------------|-----------------------|
| 01. JAVA 언어 소개 | 09. AWT 컨트롤 클래스 |
| 02. JAVA 기본 문법 | 10. AWT 이벤트 |
| 03. 객체지향 프로그래밍 | 11. AWT Container 클래스 |
| 04. 패키지와 예외처리 | 12. Applet 프로그래밍 |
| 05. java.lang 패키지 | 13. JDBC 프로그래밍 |
| 06. 멀티 스레드 프로그래밍 | 14. 네트워크 프로그래밍 |
| 07. File 클래스와 IO 스트림 | |
| 08. AWT 프로그래밍 | |

최소한의 JAVA

- | | |
|----------------------|-----------------------|
| 01. JAVA 언어 소개 | 09. AWT 컨트롤 클래스 |
| 02. JAVA 기본 문법 | 10. AWT 이벤트 |
| 03. 객체지향 프로그래밍 | 11. AWT Container 클래스 |
| 04. 패키지와 예외처리 | 12. Applet 프로그래밍 |
| 05. java.lang 패키지 | 13. JDBC 프로그래밍 |
| 06. 멀티 스레드 프로그래밍 | 14. 네트워크 프로그래밍 |
| 07. File 클래스와 IO 스트림 | |
| 08. AWT 프로그래밍 | |

최소한의 JAVA

- 01. JAVA 언어 소개
 - > Java 플랫폼
 - > Java 기초문법구조
- 02. JAVA 기본 문법
 - > 자료형, 변수와 상수
 - > 연산자와 제어문
 - > 배열과 문자열
- 03. 객체지향 프로그래밍
 - > 추상화와 클래스, 상속
 - > 객체 초기화
 - > 인터페이스
 - > 다형성과 오버로딩

최소한의 JAVA

애플리케이션 코드

C언어

```
#include <stdio.h>
int main(void)
{
    printf("Hello, C! Wn");
    return 0;
}
```

JAVA

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

두목넷 :: PASS DNA

정보
처리

필 답 형
주관식실기

정보처리실기JAVA
≠ 코딩 = 주석

우리는 어떻게 합격 할 것인가?

활가와 실가를 한번에 합격!

개편
정보처리실기

실무 알고리즘 응용

정보처리실기
합격을 위한
최소한의 JAVA

정보처리

PASS DNA

CH 01 #
JAVA 기초 문법구조

플랫폼 독립적 프로그램

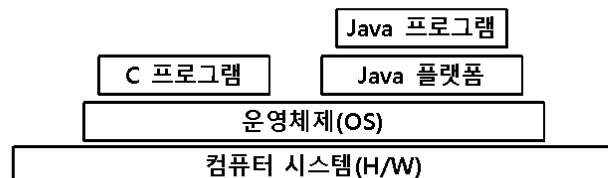
애플리케이션 코드

[Java 언어/ Java 프로그램은...
Java 플랫폼이라는 부르는 별도의 실행환경을 가지고 있다!
플랫폼(platform) : 프로그램이 동작할수 있는 실행환경]

플랫폼 독립적 프로그램

애플리케이션 코드

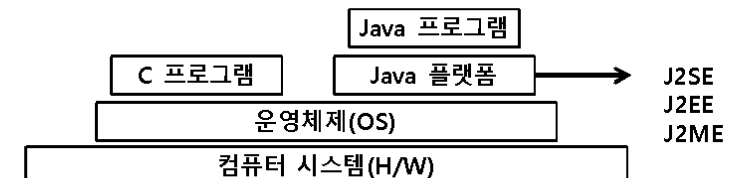
[Java 언어/ Java 프로그램은...
Java 플랫폼이라는 부르는 별도의 실행환경을 가지고 있다!
플랫폼(platform) : 프로그램이 동작할수 있는 실행환경]



플랫폼 독립적 프로그램

애플리케이션 코드

[Java 언어/ Java 프로그램은...
Java 플랫폼이라는 부르는 별도의 실행환경을 가지고 있다!
플랫폼(platform) : 프로그램이 동작할수 있는 실행환경]

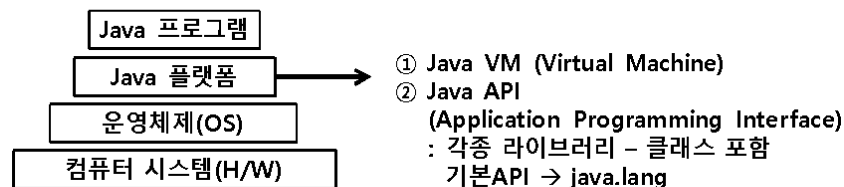


플랫폼 독립적 프로그램

애플리케이션 코드

Java 언어/ Java 프로그램은...
Java 플랫폼이라는 부르는 별도의 실행환경을 가지고 있다!

플랫폼(platform) : 프로그램이 동작할수 있는 실행환경



정보처리실기 합격을 위한
최소한의 JAVA

with C

with 객체지향
프로그래밍

:01: Java 플랫폼

주관식정보처리::PASS DNA

공개문제 05

(영역)알고리즘

배점

02

다음 Java(Java SE 7) 언어로 구현된 프로그램을 분석하여 그 실행 결과를 쓰시오.

```
public class Test {
    public static void main(String [] args) {
        int [] a = { 1, 2, 3, 4, 5, 6 };
        int i = a.length - 1;
        while ( i >= 0 ) {
            System.out.print(a[i]);
            i--;
        }
    }
}
```

○ 답 : 654321

정보처리실기 합격을 위한
최소한의 JAVA

C언어 + α

with C

with 객체지향
프로그래밍

최소한의 JAVA

애플리케이션 코드

C언어

```
#include <stdio.h>
int main(void)
{
    printf("Hello, C! Wn");
    return 0;
}
```

JAVA

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

:02: Java 기초 문법구조

C언어

test.c

```
#include <stdio.h>
void main(void)
{
    int a[6] = { 1, 2, 3, 4, 5, 6 };
    int l = sizeof(a)/sizeof(int)-1;
    while( i >= 0 ) {
        printf("%d", a[i]);
        i--;
    }
}
```

◦ 답 : 654321

JAVA

Test.java

```
public class Test {
    public static void main(String [ ] args) {
        int [ ] a = { 1, 2, 3, 4, 5, 6 };
        int i = a.length - 1;
        while ( i >= 0 ) {
            System.out.print(a[i]);
            i--;
        }
    }
}
```

◦ 답 : 654321

:02: Java 기초 문법구조

C언어

```
#include <stdio.h>
int main(void)
{
    printf("Hello, C! Wn");
    return 0;
}
```

JAVA

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

. C
함수중심

절차적 프로그래밍

. java
클래스중심 → 메소드

객체지향 프로그래밍

:03: System.out.print() 와 System.out.println()

pass001.c

```
#include <stdio.h>
void main( ) {
    printf( "%d", 100);
    printf( "%d\n", 100);
    printf( " 합격점수 : %d\n", 100);
}
```

Pass001.java

```
public class Pass001 {
    public static void main(String args[]) {
        System.out.print(100);
        System.out.println(100);
        System.out.println( " 합격점수 : " + 100);
    }
}
```

:03: System.out.print() 와 System.out.println()

Pass002.java

```
public class Pass002 {
    public static void main(String args[]) {

        int x = 10;
        int y = 20;

        System.out.println( " x + y = " + (x+y) );
        System.out.println( " x - y = " + (x-y) );
        System.out.println( " x * y = " + (x*y) );
        System.out.println( " x / y = " + (x/y) );

    }
}
```

실무 알고리즘 응용

정보처리실기
합격을 위한
최소한의 JAVA

정보처리

PASS DNA

실무 알고리즘 응용

최소한의 JAVA 특강

CH 02 #
JAVA 기본 문법(1)

정보처리

PASS DNA

실무 알고리즘 응용

최소한의 JAVA 특강

CH 02 #
JAVA 기본 문법(1)

c언어 + α

정보처리

PASS DNA

:00: C 프로그램의 구성요소

- (1) 예약어(reserved word) : 키워드
 - 자료형관련, 기억관련, 제어관련, 기타
 - int, char, static, if~else, for, while, include ---
- (2) 명칭(identifier) : 식별자
 - 변수명, 배열명, 함수명, 매크로명
- (3) 상수(constant)
 - 정수상수, 실수상수, 문자상수, 문자열상수
- (4) 연산자(operator)
- (5) 설명문(commnet) : 주석
 - 비실행문
 - 한줄주석 // C언어 공부방법
 - 여러줄주석 /* 다양한 코드를 살펴보며,
 코드에 주석을 적어보기!
 */

자료형, 변수와 상수

:00: Java 프로그램의 구성요소

- (1) 예약어(reserved word) : 키워드
 - 자료형관련, 기억관련, 제어관련, 기타
 - int, char, static, if~else, for, while ---
- (2) 명칭(identifier) : 식별자
 - 클래스명, 메소드명, 변수명, 배열명, 함수명
 - (관례) 클래스 이름의 경우, 대문자로 시작
 메소드나 필드, 변수의 이름은 소문자로 시작
- (3) 상수(constant)
 - 정수상수, 실수상수, 문자상수, 문자열상수
- (4) 연산자(operator)
- (5) 설명문(commnet) : 주석, 비실행문
 - 한줄주석 // Java언어 공부방법
 - 여러줄주석 /* 다양한 코드를 살펴보며,
 코드에 주석을 적어보기!
 */

:01: Java 의 자료형(data type)

- (1) 기본형 (Primitive Type)
 - 정해진 자료형의 값 자체

```
int a = 10;
double b = 3.14;
```

- (2) 참조형 (Reference Type)
 - (C언어) 포인터
 - 실제 값이 저장된 메모리 주소

```
int anArr[];
String rStr;
```

:01: Java 의 자료형(data type)

(1) 기본형 (Primitive Type)

- 정해진 자료형의 값 자체

자료형	예약어(크기, byte)
정수형	byte(1), int(4), short(2), long(8)
실수형	float(4), double(8)
문자형	char(2) ← UNICODE
논리형	boolean(1) ← true, false

:02: Java 의 연산자(operator)

(2) 연산자 우선순위와 결합방향

	우선순위	연산자	결합방향
단항연산자	1	() [] -> .	→
	2	! ~ ++ -- & * sizeof() cast new delete	←
이항연산자	3	* / %	→
	4	+ -	→
	5	<< >> >>>	→
	6	> >= < <= instanceof	→
	7	== !=	→
	8	&	→
	9	^	→
	10		→
	11	&&	→
	12		→
삼항연산자	13	? :	←
	14	= += *= ... &=	←
	15	,	→

:02: Java 의 연산자(operator)

(1) 연산자(operator)

- 자료에 대한 연산동작을 지정한 기호

구분	연산자
산술 연산자	+ - * / % ++ --
관계 연산자	> < >= <= == != instanceof
논리 연산자	&& !
대입 연산자	+= -= *= /= %= <<= >>= = &=
조건 연산자	? :
비트 연산자	& ^ ~ << >> >>>
기타 연산자	sizeof() cast & * new delete

제어구조

[C언어 / Java언어는...
구조화된 프로그래밍, 블록 제어구조 { }, 순차/선택/반복 구조]

>> 순차(sequence) 제어구조

>> 선택(select) 제어구조

>> 반복(looping) 제어구조

:03: Java 의 제어문

제어구조	제어문
순차구조	int a; // 변수 선언문; a = 10 + 20; // 변수 대입문;
선택구조	(1) if문 (2) switch~case문
반복구조	(1) while문 (2) do~while문 (3) for문 → for-each문
제어명령문;	(1) break; (2) continue;

:03: Java 의 제어문

Pass004.java >> for-each문 : 객체내의(여러 개의)대상을 차례로 접근할 경우 유용

```
public class Pass004 {
    public static void main(String args[]) {

        int[] scoreOfSilgi = { 25, 25, 15, 25, 10 };

        for( int i : scoreOfSilgi )
        {
            System.out.print( i + " ");
        }
        System.out.println();
    }
}
```

:03: Java 의 제어문

Pass003.java

```
public class Pass003 {
    public static void main(String args[]) {

        int[] scoreOfSilgi = { 25, 25, 15, 25, 10 };

        for( int i = 0 ; i < 5 ; i++ )
        {
            // System.out.print( i + " ");
            System.out.print( scoreOfSilgi[i] + " ");
        }
        System.out.println();
    }
}
```

:04: Java 의 무한반복

C 언어

>> while문

while(1)

```
{
    반복대상;
}
```

>> for문

for(; ;)

```
{
    반복대상;
}
```

Java 언어

>> while문

while(true)

```
{
    반복대상;
}
```

>> for문

for(; ;)

```
{
    반복대상;
}
```

:04: Java 의 무한반복 + break;

Pass005.java

```
public class Pass005 {
    public static void main(String[] args) {

        int nSum = 0;
        int i = 1;
        while(true)
        {
            nSum += i;
            if( i == 10 ) break;
            i++;
        }
        System.out.print( "1~10의 합 : " + nSum);
        System.out.println();
    }
}
```

:04: Java 의 무한반복 + continue;

Pass006.java

```
public class Pass006 {
    public static void main(String[] args) {

        int nSumOfEven = 0;
        for(int i = 1; i<=10; i++)
        {
            if( i%2 != 0 ) continue;
            nSumOfEven += i;
        }
        System.out.print( "1~10 짝수의 합 : " + nSumOfEven);
        System.out.println();
    }
}
```

개념
정보처리실기

실무 알고리즘 응용

정보처리실기
합격을 위한
최소한의 JAVA

정보처리

PASS DNA

개념
정보처리실기

실무 알고리즘 응용

최소한의 JAVA 특강

CH 02 #
JAVA 기본 문법(2)

정보처리

PASS DNA

:00: Java 의 자료형(data type)

(1) 기본형 (Primitive Type)

- 정해진 자료형의 값 자체

```
int a = 10;
```

```
double b = 3.14;
```

(2) 참조형 (Reference Type)

- (C언어) 포인터
- 실제 값이 저장된 메모리 주소

```
int anArr[];
```

```
String rStr;
```

:01: Java 의 배열(Array)

(1) C언어 배열 VS Java 배열

- C언어 : int, char형과 같은 기본형 (배열변수)
- Java : 참조형

(2) Java 배열의 선언 규칙

- 배열은 선언한 뒤 초기화나 배열객체 생성 후, 사용 가능하다.
- 배열의 크기를 지정할 수 없다.
- 다차원 배열을 255차원까지 가능하다.

(3) Java 배열의 크기

- 배열이름.length

:01: Java 의 배열(Array)

Pass007.java

```
public class Pass007 {
    public static void main(String[] args) {

        // 1차원 배열의 초기화
        // int intArr[ ] = { 1,2,3 };
        int[ ] intArr = { 1,2,3 };
        for(int i =0; i<intArr.length; i++)
        {
            System.out.print(intArr[i] + " ");
        }
        System.out.println();
    }
}
```

:01: Java 의 배열(Array)

Pass007.java

```
public class Pass007 {
    public static void main(String[] args) {

        // 1차원 배열의 생성
        // int intArr[ ] = new int[3];
        int[ ] intArr = new int[3];
        for(int i = 0; i < intArr.length; i++)
            intArr[i] = i+1;
        for(int i : intArr)
        {
            System.out.print(i + " ");
        }
        System.out.println();
    }
}
```

:02: Java 의 문자열

(2) String 클래스

- java.lang 패키지의 주요 클래스 중의 하나인 String 클래스!
- String 클래스는 주로 문자열을 출력하거나 결합하는데 사용한다.
- 자바는 편리하게 문자열을 사용 가능하다! < String 클래스
String str = new String("정보처리 한방 합격!");
- "문자열"간 결합이 용이하다. < String 클래스
System.out.println(name + "님 합격을 축하합니다!");

:02: Java 의 문자열

(1) Java의 문자열

- String : 참조형

// 1. 문자열형의 생성 (대입형)

String strArr1 = "Java";

// 2. String 클래스의 생성자를 이용하여 초기화

String strArr2 = new String("JAVA");

:02: Java 의 문자열

(3) String 클래스의 주요 메서드

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

주요 메서드	기능 설명
char charAt(intdex)	인덱스 위치의 문자 하나 리턴
boolean equals(Object 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)

:03: Java 의 문자열과 + 연산자

(1) '+' 연산자

- 문자열형 변수나 리터럴에 대하여 연결 (문자열 연결)
- +연산자를 사용할 경우, 기본형이나 참조형 데이터를 문자열로 자동 변환시켜줌

:03: Java 의 문자열과 + 연산자

Pass009.java

```
public class Pass009 {
    public static void main(String[] args) {

        String str = "SSAP";
        int length = str.length();
        for(int i = length-1; i >= 0; i--)
            System.out.printf("%c", str.charAt(i));
        System.out.println();

    }
}
```

:03: Java 의 문자열과 + 연산자

Pass008.java

```
public class Pass008 {
    public static void main(String[] args) {

        String strS1 = "Gisa";
        String strS2 = "One Pass!";
        String strS3 = strS1 + strS2;
        System.out.println(strS3);
        System.out.println(100 + "점 합격~!");

    }
}
```

:03: Java 의 문자열과 + 연산자

Pass009.java

```
public class Pass009 {
    public static void main(String[] args) {

        String str = "SSAP";
        int length = str.length();
        for(int i = length-1; i >= 0; i--)
            System.out.printf("%c", str.charAt(i));
        System.out.println();

    }
}
```

:04: Java 의 Scanner 클래스

Pass010.java

>> System.out : 콘솔 출력 객체
>> System.in : 키보드 입력 객체

```
import java.util.Scanner;
public class Pass010 {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);
        System.out.print(" = 덧셈(1) 뺄셈(2) 선택 ? ");
        String sel = input.next();
        if(sel.equals(" 1 "))
            System.out.println(" 10 + 20 = " + (10+20));
        if(sel.equals(" 2 "))
            System.out.println(" 10 - 20 = " + (10-20));

    }
}
```