

07 자바의 다양한 클래스

Sangwook Lee

07 자바의 다양한 클래스

1. 패키지 `java.lang`의 클래스

2. 패키지 `java.util`의 클래스

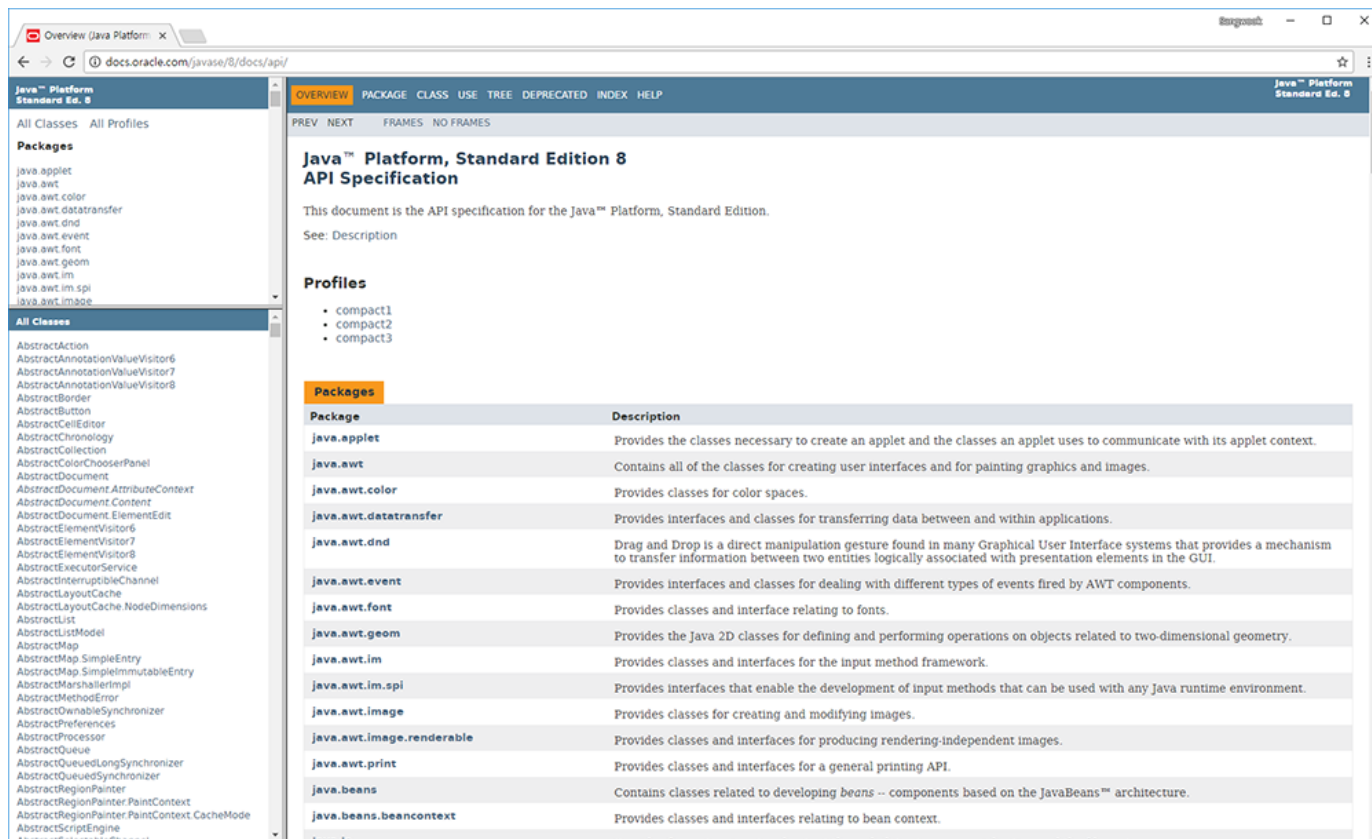
3. 컬렉션 프레임워크

4. 일반화 유형과 열거형

1. 패키지 java.lang의 클래스

- 자바 클래스 라이브러리

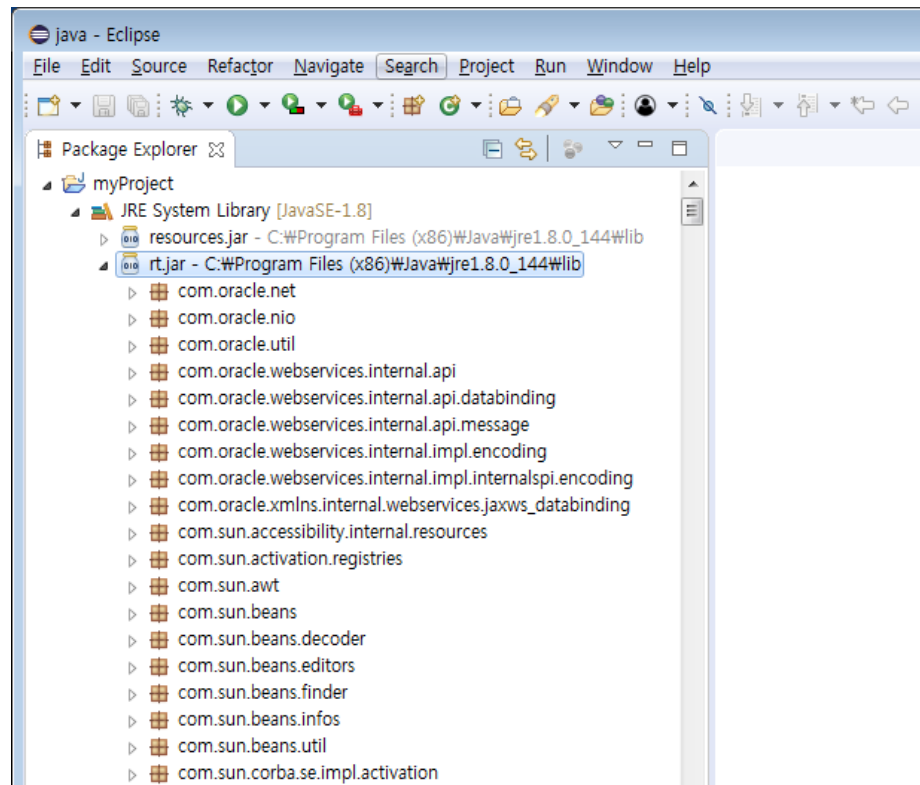
— 자바 언어를 활용하기 위해 이미 만들어 놓은 자바 클래스



<http://docs.oracle.com/javase/8/docs/api>

1. 패키지 java.lang의 클래스

- 자바 클래스 라이브러리
 - 이클립스에서 보기



Package Explorer > JRE System Library > rt.jar

1. 패키지 java.lang의 클래스

- 자바 클래스 라이브러리
 - 주요 패키지

패키지	설명
java.lang	자바 프로그램에서 자주 이용하는 기본 클래스 제공
java.io	파일, 프린터, 모니터, 키보드 등의 다양한 입출력 관련 클래스 제공
java.net	네트워크 관련 클래스 제공
java.util	날짜, 시간, 벡터, 난수 ^{random number} 등의 다양한 유틸리티 클래스 제공
java.sql	데이터베이스 프로그래밍에 필요한 다양한 클래스 제공
java.applet	애플릿을 구현하기 위한 관련 클래스 제공
java.awt	그래픽 윈도우 관련 다양한 클래스 제공
java.swing	AWT보다 간편한 스윙에 사용되는 윈도우 관련 다양한 클래스 제공

1. 패키지 java.lang의 클래스

- 자바 클래스 라이브러리

- java.lang

- 자바 기본 클래스 소속
- 유일하게 import 문장이 필요 없이 사용
- 주요 클래스

클래스	설명
Object	모든 클래스의 최상위 클래스
Byte, Short, Integer, Long	정수형을 표현하는 래퍼 클래스
Float, Double	실수형을 표현하는 래퍼 클래스
Character, Boolean	문자와 논리형을 표현하는 래퍼 클래스
String, StringBuffer	문자열을 표현하는 클래스
Math	수학 관련 함수를 정적으로 제공하는 클래스

1. 패키지 java.lang의 클래스

- 자바 클래스 라이브러리

- object

- 모든 클래스의 가장 상위 클래스
 - 클래스 정의 시 상속 관계를 명시하지 않으면 자동으로 클래스 object의 하위 클래스가 됨
 - 주요 메소드
 - getClass(): 객체의 클래스 정보 반환
 - hashCode(): 객체의 고유 번호 반환
 - toString(): “클래스이름@해시코드” 반환

1. 패키지 java.lang의 클래스

- 자바 클래스 라이브러리

- math

- 지수, 로그, 제곱근과 같은 연산 메소드 제공
 - 생성자가 private 이므로 다른 클래스에서 객체 생성 불가
 - 모든 메소드가 static이므로 클래스 이름으로 멤버 참조
 - 주요 메소드
 - round(): 인자를 반올림한 정수 반환
 - abs(): 인자의 절대 값 반환
 - pow(): 인자의 제곱근 값 반환

1. 패키지 java.lang의 클래스

[예제 7-1] object와 math 사용

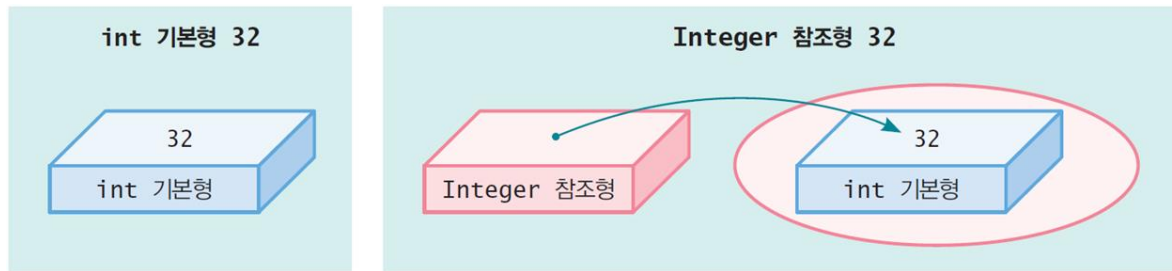
```
public class ObjectMath {  
    public static void main(String[] args) {  
        Object obj = new Object();  
        System.out.println(obj.getClass());  
        System.out.println(obj.hashCode());  
        System.out.println(obj.toString());  
  
        System.out.println(Math.PI + " " + Math.E);  
        System.out.println(Math.round(-3.5));  
        System.out.println(Math.abs(-3.4));  
        System.out.println(Math.pow(3, 4));  
    }  
}
```

```
class java.lang.Object  
31168322  
java.lang.Object@1db9742  
3.141592653589793 2.718281828459045  
-3  
3.4  
81.0
```

1. 패키지 java.lang의 클래스

- 랩퍼 클래스(wrapper class)

- 자바의 기본형을 참조형으로 만든 클래스



- 종류

구분	정수형				실수형		문자형	논리형
기본형	byte	short	int	long	float	double	char	boolean
참조형	Byte	Short	Integer	Long	Float	Double	Character	Boolean

1. 패키지 java.lang의 클래스

- 래퍼 클래스

- Integer

- 정수 또는 정수 형태의 문자열 인자로 생성
 - 객체에 저장된 값은 수정 불가
 - 주요 메소드
 - intValue(): 값을 int로 반환
 - parseInt(String s, int radix): s를 radix 진법으로 분석하여 int로 반환
 - toBinaryString(int i): i를 이진수 문자열로 반환
 - toString(int i, int radix): i를 radix 진법의 문자열로 반환

1. 패키지 java.lang의 클래스

[예제 7-2] Integer 사용

```
public class WrapperInteger {  
    public static void main(String[] args) {  
        Integer age = new Integer(20);  
  
        System.out.println("나이:" + age.intValue());  
  
        System.out.println(Integer.parseInt("ff", 16));  
        System.out.println(Integer.toBinaryString(255));  
        System.out.println(Integer.toString(255, 16));  
    }  
}
```

```
나이:20  
255  
11111111  
ff
```

1. 패키지 java.lang의 클래스

- 래퍼 클래스

- 박싱과 언박싱(boxing and unboxing)

- 실수 2.59를 저장하는 Double형 객체를 만든 후, 객체에 저장된 값을 double형 변수 r에 저장하는 방법

- 전통적인 방법

```
Double radius = new Double(2.59);  
double r = radius.doubleValue();
```

- 박싱과 언박싱

```
Double radius = 2.59;  
double r = radius;
```

1. 패키지 java.lang의 클래스

- 래퍼 클래스

- 박싱과 언박싱(boxing and unboxing)

- 박싱

- 기본형에서 래퍼 객체로 자동 변환

- 언박싱

- 래퍼 객체에서 기본형으로 자동 변환

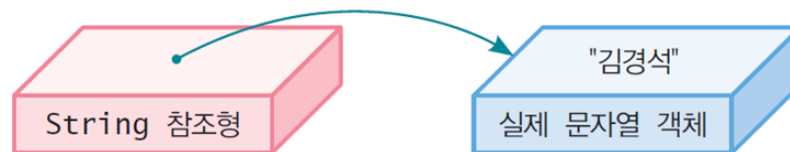
박싱과 언박싱은 모든 래퍼 클래스에서 이용 가능!

1. 패키지 java.lang의 클래스

- String과 StringBuffer

- String

- 문자열 클래스
- 생성 되면 문자열 수정 불가
- 생성 방법
 - 문자열 상수 직접 대입
 - String 생성자 호출



```
String snum = "20123458";  
String name = new String("김경석");  
String dept = new String(new char[] {'کم', '퓨', '터', '학', '과'});
```

1. 패키지 java.lang의 클래스

- String과 StringBuffer
 - String 주요 메소드

메소드	설명
int compareTo(String str)	인자 str과 이 문자열과 비교하여 같으면 0을 리턴, 다르면 다른 첫 문자를 비교하여 유니코드 값에 따라 양수(인자의 유니코드가 작을 때), 음수(인자의 유니코드가 클 때)를 반환
boolean equals(Object anObject)	인자 문자열과 비교하여 동일하면 true, 아니면 false
boolean equalsIgnoreCase(String str)	대소문자 무시하고 인자 문자열과 비교하여 동일하면 true, 아니면 false
int indexOf(int ch)	문자 ch가 나타나는 처음 인덱스를 반환
int length()	문자열의 길이를 반환
String concat(String str)	인자 str를 연결한 새로운 문자열 생성하여 반환
String replace(char oldChar, char newChar)	oldChar를 newChar로 대체한 문자열 반환
String substring(int beginIndex)	첨자 beginIndex에서부터 끝까지의 문자열 반환
String substring(int start, int end)	첨자 start에서 end-1까지의 문자열을 반환
String toLowerCase()	모든 문자를 소문자로 만든 문자열을 생성하여 반환
String toUpperCase()	모든 문자를 대문자로 만든 문자열을 생성하여 반환
boolean endsWith(String suffix)	인자 suffix로 종료되는지 검사
public int indexOf(int ch)	문자 ch가 나타나는 처음 첨자를 반환
public int lastIndexOf(int ch)	문자 ch가 나타나는 마지막 첨자를 반환

1. 패키지 java.lang의 클래스

[예제 7-5] String 활용

```
public class StringInfo {  
    public static void main(String[] args) {  
        String snum = "20123458";  
        String name = new String("김경석");  
        String dept = new String(new char[] {'컴', '퓨', '터', '학', '과'});  
        String java = new String("java");  
  
        System.out.printf("%d ", snum.compareTo("20123458"));  
        System.out.printf("%d %n", snum.compareTo("20123456"));  
        System.out.printf("%s ", name);  
        System.out.printf("%s ", name.replace('김', '강'));  
        System.out.printf("%s %n", name);  
        System.out.printf("%s ", dept);  
        System.out.printf("%s ", dept.substring(0, 3));  
        System.out.printf("%d %n", dept.length());  
        System.out.printf("%s ", java.indexOf('a'));  
        System.out.printf("%s ", java.concat(" language"));  
        System.out.printf("%s ", java.replace('j', 'J'));  
        System.out.printf("%s ", java.toUpperCase());  
        System.out.printf("%b ", java.equals("JAVA"));  
        System.out.printf("%s ", java.equalsIgnoreCase("JAVA"));  
        System.out.printf("%s %n", java.substring(1));  
    }  
}
```

0 2

김경석 강경석 김경석

컴퓨터학과 컴퓨터 5

1 java language Java JAVA false true ava

1. 패키지 java.lang의 클래스

- String과 StringBuffer

- StringBuffer

- 문자열 클래스
 - 생성 후 문자열 수정 가능
 - 생성 방법
 - StringBuffer 생성자 호출
 - 생성자 인자로 용량이나 문자열 가능
 - » 기본 생성자로 생성된 객체의 용량은 16

```
StringBuffer sb1 = new StringBuffer();  
StringBuffer sb2 = new StringBuffer(32);  
StringBuffer sb3 = new StringBuffer("java");
```

1. 패키지 java.lang의 클래스

- String과 StringBuffer
 - StringBuffer 주요 메소드

메소드	설명
StringBuffer append(String str)	문자열 str을 문자열 마지막에 추가
StringBuffer delete(int start, int end)	첨자 start에서 end까지 삭제
StringBuffer insert(int offset, String str)	첨자 위치 Offset에 문자열 str을 삽입
StringBuffer reverse()	문자열을 이루는 문자의 순서를 역으로 변환
StringBuffer replace(int start, int end, String str)	첨자 start에서 end-1까지 문자열 str로 대체
String substring(int start)	첨자 start에서 마지막까지의 문자열을 반환
String substring(int start, int end)	첨자 start에서 end-1까지의 문자열을 반환
int length()	문자열 길이를 반환
void setLength(int newLength)	지정된 길이 만큼의 문자열로 재지정
String toString()	문자열 String으로 반환
int capacity()	문자열의 용량을 반환
void setCharAt(int index, char ch)	지정된 index 위치에 문자 ch를 지정
char charAt(int index)	지정된 index 위치의 문자를 참조

1. 패키지 java.lang의 클래스

[예제 7-6] StringBuffer 활용

```
public class StringBufferInfo {  
    public static void main(String[] args) {  
        StringBuffer pg = new StringBuffer("Java");  
  
        System.out.println(pg.capacity());  
        System.out.println(pg.append(" language"));  
        System.out.println(pg.insert(5, "programming "));  
        System.out.println(pg.capacity());  
        System.out.println(pg);  
        System.out.println(pg.replace(0, 4, "Objective-C"));  
        System.out.println(pg.substring(0, 9));  
        System.out.println(pg);  
        System.out.println(pg.charAt(10));  
        pg.setCharAt(10, 'D');  
        System.out.println(pg);  
    }  
}
```

```
20  
Java language  
Java programming language  
42  
Java programming language  
Objective-C programming language  
Objective  
Objective-C programming language  
C  
Objective-D programming language
```

Backup Slides

Levels of Program Code

- High-level language program (in C)

```
swap (int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

one-to-many

C Compiler

- Assembly language program (for MIPS)

```
swap:    sll    $2, $5, 2
         add    $2, $4, $2
         lw     $15, 0($2)
         lw     $16, 4($2)
         sw     $16, 0($2)
         sw     $15, 4($2)
         jr     $31
```

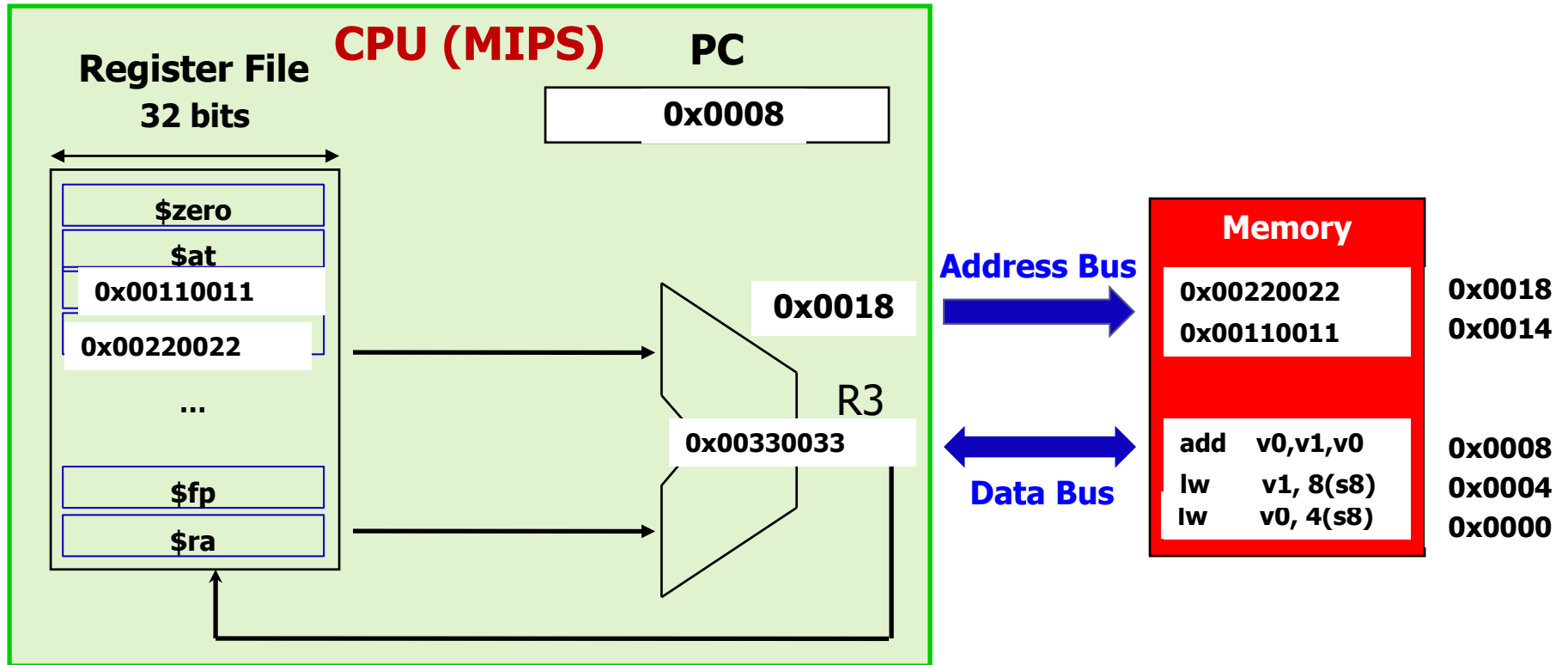
one-to-one

Assembler

- Machine (object, binary) code (for MIPS)

```
000000 00000 00101 0001000010000000
000000 00100 00010 0001000000100000
. . .
```

CPU Execution Example



중앙 처리 장치의 종류와 특성

- 사용 목적에 따른 컴퓨터 분류

- 범용 컴퓨터

- 사무처리, 과학 계산 등의 다양한 업무를 처리할 수 있도록 만들어진 컴퓨터
 - 우리 주변에서 일반적으로 볼 수 있는 컴퓨터

- 임베디드 시스템

- 특정 응용 분야에 적합하도록 만들어진 컴퓨터
 - 휴대폰, 게임기, 자동차 등에 내장된 컴퓨터



중앙 처리 장치의 종류와

범용 컴퓨터용
중앙 처리 장치

* 코어

여러 개의 중앙 처리 장치가 하나의 칩으로 구현될 때 각각의 중앙 처리 장치를 코어라고 부른다.

범용 컴퓨터에 사용하는 중앙 처리 장치는 다양한 분야에서 좋은 성능을 발휘할 수 있도록 설계된다.

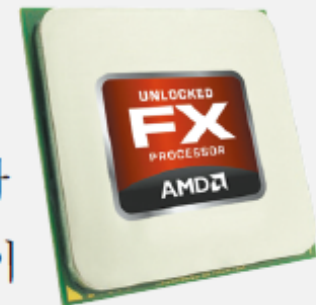
① 코어 i3/i5/i7

인텔사에서 개발한 중앙 처리 장치로, 2개 또는 4개의 코어*를 가진 모델이 많이 사용되고 있으며, 그래픽 프로세서가 포함되어 있다. 우수한 성능과 호환성 때문에 높은 시장 점유율을 확보하고 있다.



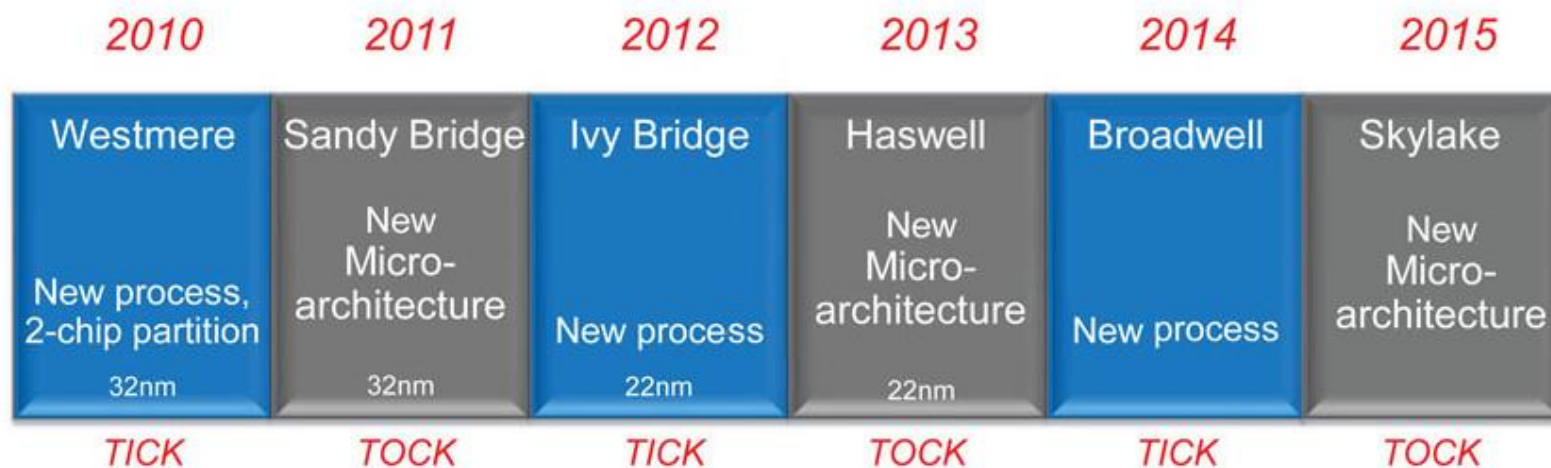
② FX 시리즈

AMD사에서 개발한 중앙 처리 장치로, 개인용 컴퓨터에 사용되는 중앙 처리 장치로는 최초로 8개의 코어를 가진 모델이 있다. 비슷한 성능을 가진 인텔사의 제품과 비교했을 때 상대적으로 저렴한 가격이 장점이다.



Tick Tock

Intel Tick-Tock Model

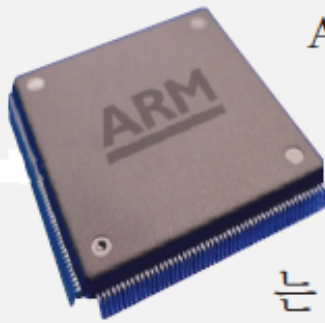


Tick-tock delivers leadership through technology innovation on a reliable and predictable timeline

중앙 처리 장치의 종류와 특성

임베디드 시스템에 사용하는 중앙 처리 장치는 특정 분야에 적합한 성능을 발휘할 수 있도록 설계된다.

① ARM 시리즈



ARM사에서 개발한 중앙 처리 장치로, ARM은 현재 스마트폰을 포함한 대부분의 모바일 기기 시장에서 높은 점유율을 확보하고 있다. ARM의 가장 큰 특징은 저전력으로 동작한다는 것이다.

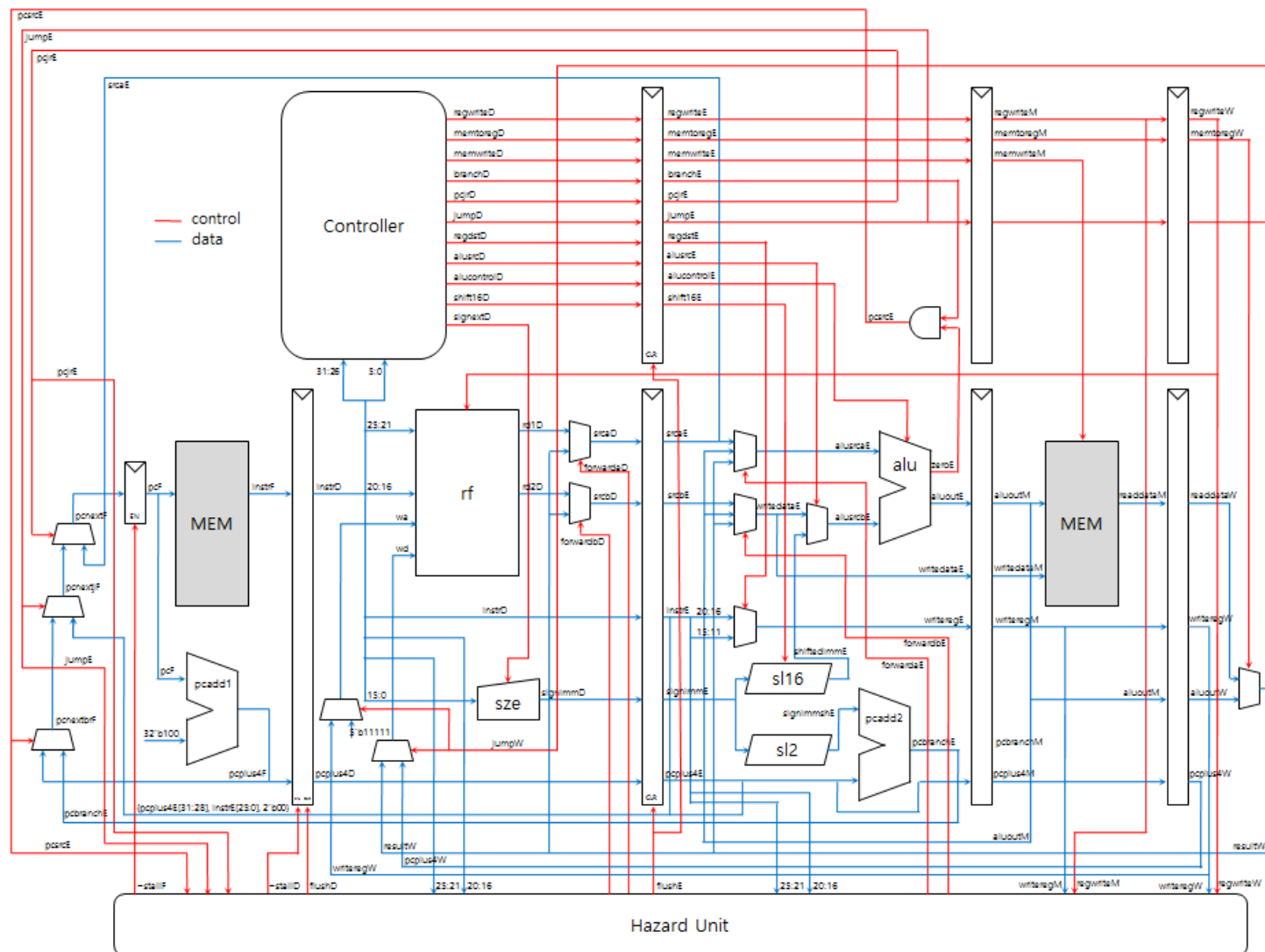
ARM은 스마트폰과 디지털카메라뿐만 아니라 컴퓨터 제품군까지 영역을 확장하고 있다.

② MIPS 시리즈

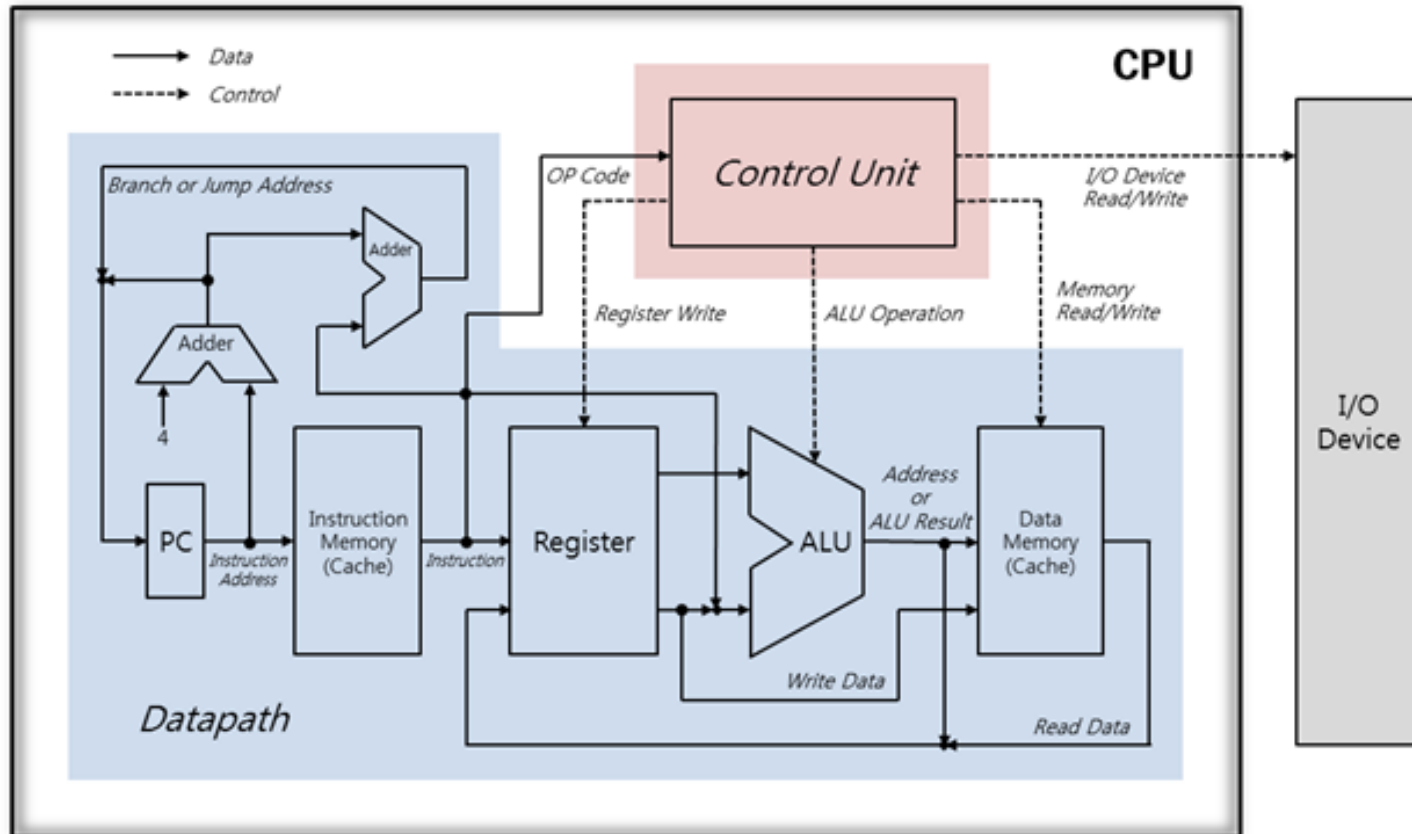


MIPS사에서 개발한 중앙 처리 장치로, 네트워크 장비나 공장 자동화 시스템 등의 분야에 많이 사용되고 있다.

MIPS는 많은 대학에서 컴퓨터 구조 분야 수업에 활용하고 있다.

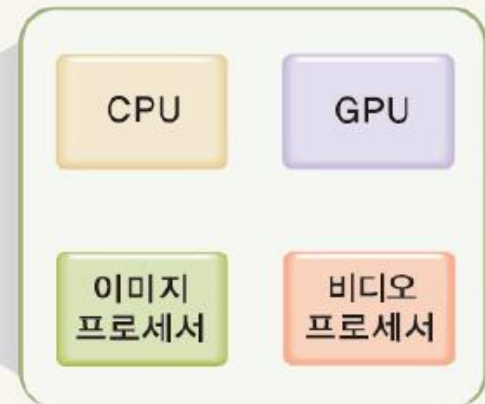
The logo for MIPS CPU, featuring the letters 'MIPS' and 'CPU' in a stylized, rounded font, each letter contained within a brown hexagonal shape.

MIPS CPU



Mobile AP(Application Processor)

스마트폰과 태블릿 PC 사용자들이 늘어나면서 모바일 AP에 관한 관심이 높아지고 있다. AP는 모바일 기기의 두뇌와 같은 역할을 한다는 점에서 컴퓨터의 중앙 처리 장치와 유사하다. AP란 특정한 용도에 적합한 고성능 저 전력 모바일 기기를 만들기 위해 중앙 처리 장치를 비롯한 다양한 장치들을 하나의 칩으로 만들어 놓은 것을 말한다.



▲ AP 구조

Mobile AP(Application Processor)

- AP 종류

- 모바일 AP 주요 제조사: 퀄컴, 엔비디아, 삼성, 애플

제조사	애플	삼성	퀄컴	엔비디아
AP	 A series	 Exynos	 Snapdragon	 Tegra

이들 제조사에서 만드는 AP에 내장된 CPU는 모두 **ARM** (이)다