

기본타입과 참조타입의 차이

기본타입: 변수에 값 자체를 저장

참조타입: 메모리상 객체가 있는 위치를 저장

배열과 ArrayList의 차이

둘다 데이터 타입이 모두 같으면서 하나이상의 데이터를 저장할 수 있다.

배열은 초기에 배열 크기를 정해야 한다.(고정길이이기 때문)

ArrayList의 경우 동적으로 추가적인 데이터 할당이 가능하다.

클래스와 객체, 메소드

객체: 물리적으로 존재하거나(사물) 개념적으로 식별 가능한 것

클래스: 공통된 속성과 기능을 갖는 객체 집합(필드,

생성자, 메소드)

메소드: 객체가 실행해야 할 기능

객체 지향 언어의 특징

컴퓨터가 수행하는 작업을 객체들간의 상호작용으로 표현

1. 캡슐화 : 객체의 속성과 기능을 하나로 묶고 실제 구현 내용 일부를 외부에 감추어 은닉하는 것(접근제한자) -> 외부 사용으로 객체 손상되지 않기 위해서(보통 필드는 은닉화)
2. 상속 : 새로운 클래스가 기존 클래스의 필드, 생성자, 메소드 이용할 수 있게 하는 기능 -> 코드의 재사용성 높여주고 유지 보수 시간 최소화 시켜 준다.

3. 다형성 : 사용방법은 같지만 결과가 다양하게 나온다. - 즉 상위클래스에 있는 메소드를 오버라이딩 할 때, 하위클래스에서 메소드의 기능이 달라진다. -> 업/다운캐스팅

4. 추상화 : 공통의 속성이나 기능을 묶어 추상클래스를 만들어 사용

오버로딩 오버라이딩

오버로딩 : 함수의 재정의(함수의 이름만 동일하고 매개변수가 달라야한다(매개변수의 수, 배치 순서))

특정클래스 내에서 이름이 같은 메소드를 여러 개 중복해서 정의 한 것

오버라이딩 : 부모 클래스의 함수 재정의
(함수이름, 매개변수, 반환형 그대로)

접근제한자

Private : 같은 클래스 내에서만 접근 가능

Default: 같은 패키지 내에서만 접근 가능

Protected: 같은 패키지 내에서, 그리고 다른 패키지의 자식 클래스에서 접근 가능

Public: 접근 제한이 없다

Public > protected > default > private

Call by valule & call by reference

Call by valule : 인수로 기본형을 사용

call by reference : 매개변수의 주소에 값을 저장하는 방식, 클래스 객체를 인수로 전달한 경우

싱글톤

Abstract/interface 차이

Abstract: 추상클래스/추상메소드를 한 개 이상 포함하고 있는 클래스를 의미

둘다 객체 생성을 할 수 없다.

Abstract은 상속(extends) 개념으로 사용되고 interface는 구현(implement) 개념으로 사용됨

다중 상속의 목적이 되냐 안되냐

둘 다 상속받은 클래스에서 오버라이딩 하여 사용한다.

라이브러리 프레임워크 차이점

라이브러리: 자바에서 쓰는 기능을 모아 놓은 유틸 모음집

프레임워크 : 일정 수준 이상의 품질을 보증하는 코드를 비교적 빠르고 편하게 완성, 유지 보수 할 수 있는 솔루션

자바컬렉션의 대표 인터페이스

같은 타입 참조값을 여러 개 저장하기 위한 라이브러리

Set : 중복허용하지 않고 순서 가지지 않는다.

프로젝트에서는 어느부분에 ~~이용했다.

List : 중복허용하고 순서 유지된다.

Map : key, value 형태로 저장한다. key값은 중복을 허용하지 않는다.

스레드와 프로세스

스레드 : 하나의 프로그램 내에서의 실행 단위로 프로세스 내에 존재한다.

프로세스 : 실행 중인 프로그램

운영체제 내에 존재한다

보통 멀티스레드를 사용하는데 한 프로그램에서 여러 실행 흐름이 있어 동기화 방법을 이용해야 함

(wait/notify, synchronize 함수와 블록) – 프로젝트와 연결해서 설명