

Language

1. 절차지향언어 / 객체지향언어

1-1. 개념

절차지향언어란 코드를 순차적으로 작업하는 언어이다. 예를들어 C언어가 있다.

객체지향언어란 코드를 객체 중심으로 구현하는것이다. 예를들어 JAVA, Python이 있다.

1-2. 절차지향언어의 장점/단점

- 빠르다
- 코드수정이 어렵다
- 코드 재사용이 어렵다

1-3. 객체지향언어의 장점/단점

- 절차지향언어보다 느리다
- 코드 재사용성이 높다
- 유지보수에 용이하다
- 코드의 중복제거

2. JAVA

2-1. 클래스 / 객체

객체가 사물이라면 사물의 행동, 상태 등을 정의해놓은것을 클래스라고 한다.

예를들어, 객체가 자동차면 클래스는 자동차 설계도라고 할 수 있다.

```
class Tv {  
    //속성(변수)  
    int channel; //채널  
    boolean power; //상태  
  
    //기능(메서드)  
    void power(){}  
    void channelUp() {}  
    void channelDown() {}  
}
```

2-2. 인스턴스

```
TV tv1;  
tv1=new TV();
```

tv1은 TV의 인스턴스이다.

```
TV tv1;  
tv1=new TV();  
tv1=tv2;
```

tv2는 tv1이랑 같은 주소값을 가리키고 있다.

```
TV tv1=new TV();  
TV tv2=new TV();
```

tv1과 tv2는 각기 다른 TV 인스턴스이다.

2-3. 구조체 / 클래스

- 변수 : 1

- 배열 : [1,2,3]
- 구조체 : struct{1, "abc", 10.123}
- 클래스 : class{1, "abc", 10.123, void(), int()}

2-4. 변수 종류

```
class TV{
    //클래스 영역
    //클래스 변수, 인스턴스 변수
    void power(){
        //메서드 영역
        //지역 변수
    }
}
```

인스턴스 변수는 인스턴스마다 고유의 값을 가질 수 있다. 생성시기는 인스턴스 생성 시기와 같다.

클래스 변수는 인스턴스 변수 앞에 static이 붙은 변수이다. 인스턴스 변수는 각 인스턴스가 독립적인 공간에 가지고 있지만 클래스 변수는 인스턴스들이 한곳을 공유하여 참조하고 있는 변수이다. 클래스가 생성되고 없어질때까지 메모리영역에 있지만 프로그램 전체적으로 공유하고 싶을 때는 앞에 public을 붙여 전역변수로 선언한다.

지역변수는 메서드 내에서 생성되는 변수이다. 메서드가 생성되고 없어질때까지만 유지된다.

2-5. Overloading (오버로딩)

메서드 이름은 같지만 매개변수 개수나 타입이 다르다. (리턴타입은 상관 X)

```
void println()
void println(int x)
void println(boolean x)
```

<오버로딩의 장점>

- 같은 역할을 하는 메서드가 이름을 달리 안하여도 된다.
- 메서드 이름을 절약할 수 있다. (고민 필요 X)

2-6. 생성자

인스턴스 초기화를 위한 것이다.

2-7. 상속

상속이란 클래스를 재사용하여 새로운 클래스를 만드는 것이다. 코드 재사용으로 개발 효율을 높이는 방법중에 하나이다.

```
class Parent{}
class Child extends Parent{}
```

2-8. Overriding(오버라이딩)

부모클래스에게 상속받은 메서드 내용을 변경하는 것이다. 오버라이딩은 이름이 같아야하며 매개변수와 리턴타입도 같아야한다.

2-9. package / import

- package란 ?

package 는 클래스를 묶어 놓은 것이다. 서로 관련된 클래스끼리 묶어서 관리할 수 있다는 것이 장점이다.

예를들어 java.lang.String은 String에 관련된 클래스들을 묶어 놓은 것이다.

- import란?

import는 다른 패키지의 클래스를 사용하고 싶을때 불러오는 방식이다.

2-10. 추상 클래스 / 추상 메서드

- 추상 클래스

추상 클래스는 미완성된 메서드(=추상 메서드)를 포함한 클래스이다. 추상 클래스는 인스턴스를 생성할 수 없다. 상속을 통해서 자손 클래스만 완성하여 사용할 수 있다.

- 추상 메서드

추상 메서드는 미완성 메서드이다. 상속받을 자손이 상황에 맞게 완성해야하는 메서드이다.

2-11. Interface (인터페이스)

인터페이스는 추상 메서드와 상수만 멤버로 가질 수 있는 추상 클래스이다.

<인터페이스의 장점>

- 개발 시간을 단축시킨다. 그리고 협업하는데 있어서 동시 구현이 가능하다. (인터페이스를 구현하는 클래스 개발자와 클래스를 호출하여 사용하는 개발자가 동시 개발이 가능)
- 기본틀을 인터페이스로 작성하기 때문에 일관되게 프로그램 개발이 가능하다.
- 관계없는 클래스끼리 맺어줄 수 있다. 같은 조상이 아닌 클래스를 인터페이스로 공통된 역할관계를 맺어 줄 수 있다.
- 독립적인 프로그래밍이 가능하다. 클래스 하나가 변경되어도 다른 클래스 변경이 되지 않는다.

2-12. 자바개발환경

- JRE(자바 실행 환경): JVM + Java API
- JDK(자바 개발 도구): JRE + 자바 컴파일러

JVM은 자바 실행을 위한 가상 환경이다. 사용자 응용프로그램과 운영체제 사이에서 자바 실행환경을 조성해주는 것이다.

이로써 자바는 운영체제에 종속되지않고 독립적으로 실행될 수 있는 장점이있다. 하지만 JVM은 운영체제에 맞게 설치해주어야한다.

2-13. 자바 컴파일 과정

1. .java 파일 생성
2. Build를 하면 자바 컴파일러로 .class 파일 생성
(.class파일은 자바 바이트 코드이다)
3. 클래스 로더가 class파일을 JVM에 로드한다.
4. 클래스파일은 JVM에서 인터프리터에의해 실행된다.
5. JVM에 stack, method, heap 영역이 생성된다.

<자바는 왜 컴파일러랑 인터프리터를 같이 쓸까 ?>

컴파일러는 코드를 모두 번역한 후에 실행하기 때문에 빠르다라는 장점은 있지만 실행환경에 민감하다. 다른 운영체제에서는 다시 변환하여 컴파일 되어야하는 단점이 있다. 하지만 인터프리터는 프로세스가 바로 실행시킬 수 있어 운영체제에 종속되지 않는다는 장점이 있지만 컴파일러 보다는 실행시간이 느리다는 단점이 있다.

https://m.blog.naver.com/ehcibear314/221228200531

2-14. Garbage Collector

Garbage collector는 Heap 영역의 메모리를 관리한다. 즉, 참조되지 않는 객체들의 메모리를 회수하는 역할을 한다.

1. Heap 영역에 객체가 생성되면 Eden 영역에 할당된다.
2. Eden영역에 데이터가 어느정도 차면 참조정도에 따라 Survivor1, Survivor2 중 빈 공간으로 이동되거나 회수된다.
3. Eden, Survivor 영역이 어느정도 차면 Old 영역으로 이동되거나 회수된다.
4. Old 영역에 할당된 메모리가 허용범위를 넘으면 모든 객체들을 검사하여 참조되지 않은 객체를 한꺼번에 삭제한다. (Garbage Collector)
5. Garbage Collector 하는 때에는 실행 스레드를 제외한 모든 스레드의 작업을 멈춘다. (Stop-the-World)

https://aljjabaegi.tistory.com/387

2-15. 자바가 좋은 이유

C언어와 같은 언어는 미리 메모리를 할당하는 작업이 필요하며 필요없는 메모리는 다시 직접 회수해야하는 단점이 있다. 미리 예측받아서 할당하기때문에 빠르게 메모리 영역을 사용할 수는 있으나 메모리 공간을 할당하고서는 사용안할경우에는 낭비가 될 수 있는 단점이 있다.

이런면에서 자바엔진에서 Garbage Collector가 메모리 회수하는것이 자바가 좋은 이유라고 할 수 있다. 하지만 Garbage Collector를 하는 순간에 스레드 작업을 멈추기 때문에 성능 저하가 올 수 있는것이 단점이다.

또한, 자바로 개발된 응용프로그램은 운영체제 종속적이지 않을 수 있다. JVM위에서 동작되기 때문이다. 하지만 자바로 개발된 응용프로그램을 실행하기 위해서는 운영체제에 맞는 JVM을 반드시 설치해주어야하는 것이 중요하다. 이러한 장점때문에 사용자 응용프로그램은 자바로 개발을 많이 하는 반면 하드웨어적인 정교한 작업은 C 언어로 구현한다고한다.

https://wikidocs.net/22318