2023-03-20

데이터 타입 분류

- 참조타입
  - 객체의 번지를 참조하는 타입
  - 배열, 열거, 클래스, 인터페이스 타입
  - 기본타입으로 선언된 변수는 값 자체를 저장하지만, 참조타입으로 선언된 변수는 객체가 생성된 메모리 번지를 저장

메모리 사용영역

- 메서드, 힙, 스택
  - JVM은 운영체제에서 할당받은 메모리 영역을 메서드, 힙영역, 스택 영역으로 구분해서 사용
  - 메서드영역 : 바이트코드 파일을 읽은 내용이 저장되는 영역
  - 힙 영역 : 객체가 생성되는 영역. 객체의 번지는 메서드 영역과 스택 영역의 상수와 변수에 참조
  - 스텍 영역 : 메서드를 호출할 때마다 생성되는 프레임이 저장되는 영역

참조 타입 변수의 == != 연산

- ==, != 연산자는 객체의 번지를 비교해 변수의 값이 같은지, 아닌지를 조사
- 번지가 같다면 동일한 객체를 참조하는 것이고, 다르다면 다른 객체를 참조한다.

null 과 NullPointerException

null 값

- null(널)값 : 참조 타입 변수는 아직 번지를 저장하고 있지 않다는 뜻
- null도 초기값으로 사용할 수 있기 때문에 null로 최기화된 참조변수는 스택 영역에 생성

NullPointerException

- 변수가 null인 상태에서 객체의 데이터나 메서드를 사용하려 할 때 발생하는 예외
- 참조 변수가 객체를 정확히 참조하도록 번지를 대입해 해결

문자열(String)타입

문자열은 String 객체로 생성

```
// 문자열 리터럴이 동일하다면 String 객체를 공유
String name1 = "홍길동";
String name2 = "홍길동";
```

new 연산자(객체 생성자 연산자)로 직접 String 객체를 생성/대입 가능

```
charAt()메서드는 문자열에서 매개값으로 주어진 인덱스의 문자를 리턴해 특정 위치의 문자를 얻을 수 있다
JAVA // 문자열 위치는 0부터 시작
0123

replace()메서드는 기존 문자열은 그대로 두고, 대체한 새로운 문자열을 return
String oldStr = "자바 프로그래밍"; // 자바 프로그래밍 출력
String newStr = oldStr.replace("자바", "JAVA"); //JAVA 프로그래밍 출력

String board = " 번호, 제목, 내용, 글쓴이";
String[] arr = board.split(","); //csv 파일에서 많이 사용함.
// split : , 를 기준으로 배열을 생성하여 return 합니다.
```

배열 복사

- 배열은 한 번 생성하면 길이를 변경할 수 없기 때문에 더 많은 저장 공간이 필요하다면 더 큰 길이의 배열을 새로 만들고 이전 배열로부터 항목들을 복사해야한다.
- System의 arraycopy()메서드를 이용해 배열 복사가 가능하다.

```
System.arraycopy(Object src, int srcPos, Object dest, int destPos, int length);
src : 원본배열
srcPos: 원본배열 복사 시작 인덱스
dest : 새 배열
destPos : 새 배열 붙여넣기 시작 인덱스
length : 복사항목수
```

Enum(열거)

- 요일, 계절처럼 한정된 값을 갖는 타입
- 우선 열거타입 이름으로 소스파일(.java)을 생성하고 한정된 값으로 코드를 정의해야한다.
- camel표기법으로 이름을 지어준다.

패키지(package)

- 클래스들을 묶는 단위, 실제로는 폴더가 되며 자바소스의 제일 위에 패키지 선언이 되어야한다.
- 패키지화는 필수가 아니다.
- 클래스의 정식 명칭 : PackageName.ClassName, PackageName.SubPackageName.ClassName
- 특정 패키지의 클래스나 인터페이스 사용시 import문을 선언하여 사용하며 java.lang 패키지는 자동 import된다.

Wrapper Class

포장클래스 : 기본형 데이터를 객체로 만들 때

```
type        Class
----------------------
int         Integer
char        Character
double      Double
boolean     Boolean
```

StringBuffer

```
String str="";
while(true) {
  str += br.readLine();
}
```

제네릭(generic)구문

프로그램에서 처리할 데이터의 타입을 일반화시킨다.

재사용성을 높이는 결과가 된다.

클래스에서 처리할 데이터 타입을 클래스를 만들 때 정하는 것이 아닌 클래스를 객체 생성할 때 정하는 것이다.

```
Value2 v2 = new Value2();

Value3<String> v3 = new Value3<>(); // 문자열타입을 처리하는 Value3
Value4<Date> v4 = new Value4<>(); // Date 타입을 처리하는 Value4
```

자료구조

데이터를 효율적으로 사용할 수 있도록 구조를 만들어서 저장.

▼ 자료구조의 종류

- 리스트(list): ArrayList, LinkedList, (Vector)
- 스텍(stack) : LinkedList, (Stack)
- 큐(queue) : LinkedList
- 해쉬 테이블(hashtable) : HashMap, (HashTable)
- 집합(set) : HashSet

  * 엄밀히 말하면 자료구조가 아님

HashMap은 먼저 들어온 데이터가 있으면 나중에 들어오는 데이터로 오버라이딩된다.

HashSet은 중복데이터를 허용하지 않는다. 꺼내는 메서드를 제공하지 않는다.

```
Iterator<String> iterator = set.iterator(); - iterator메서드를 호출하여 객체를 가져온다
while (iterator.hasNext()) {
  String str = iterator.next();
  // 데이터 처리 부분
}
// while {} - 객체의 데이터를 순서대로 가져와서 처리.
```