

Part 2. 객체 지향 프로그래밍

Chapter 05. 참조 타입

5.1 데이터 타입 분류

- 기본 타입 (primitive type)
 - 정수 타입
 - byte
 - char
 - short
 - int long
 - 실수 타입
 - float
 - double
 - 논리 타입
 - boolean
- 참조 타입 (reference type)
 - 배열 타입
 - 열거 타입
 - 클래스
 - 인터페이스

기본 타입으로 선언된 변수는 *값 자체*를, 참조 타입으로 선언된 변수는 *객체가 생성된 메모리 번지*를 저장.

변수들은 모두 *스택(stack)*이라는 메모리 영역에 생성된다. 기본 타입 변수는 직접 값을 저장하고 있지만 참조 타입 변수는 *힙 메모리 영역*의 객체 번지를 저장하고 이 번지를 통해 객체를 참조한다.

5.2 메모리 사용 영역

JVM의 운영체제에서 할당받은 메모리 영역(Runtime Data Area) 구분 방법

- 메소드 영역(Method Area)
 - 바이트코드 파일 내용
 - 클래스 별 상수, 정적 필드, 메소드 코드, 생성자 코드 등 저장
- 힙 영역(Heap Area)
 - 객체가 생성
 - 객체의 번지는 메소드 영역과 스택 영역의 상수와 변수에서 참조
- 스택 영역(Stack)
 - 메소드를 호출할 때마다 생성되는 프레임(Frame)이 저장
 - 메소드 호출이 끝나면 프레임은 자동 제거
 - 프레임 내부에는 로컬 변수 스택이 있고, 여기서 기본 타입 변수와 참조 타입 변수가 생성되고 제거

5.3 참조 타입 변수의 ==, != 연산

```
package ch05.sec03;

public class ReferenceVariableCompareExample {
    public static void main(String[] args) {
        int[] arr1; //배열 변수 arr1 선언
        int[] arr2; //배열 변수 arr2 선언
        int[] arr3; //배열 변수 arr3 선언

        arr1 = new int[] { 1, 2, 3 }; //배열 { 1, 2, 3 }을 생성하고 arr1 변수에 대입
        arr2 = new int[] { 1, 2, 3 }; //배열 { 1, 2, 3 }을 생성하고 arr2 변수에 대입
        arr3 = arr2; //배열 변수 arr2의 값을 배열 변수 arr3에 대입

        System.out.println(arr1 == arr2); // arr1과 arr2 변수가 같은 배열을 참조하는
지 검사
        System.out.println(arr2 == arr3); // arr2와 arr3 변수가 같은 배열을 참조하는
지 검사
    }
}
```

결과

```
false
true
```

5.4 null과 NullPointerException

- null
 - 참조 타입 변수가 아직 번지를 저장하고 있지 않다는 뜻
 - null로 초기화된 참조 변수는 스택 영역에 생성
- NullPointerException
 - 변수가 null인 상태에서 객체의 데이터나 메소드를 사용하려 할 때 발생하는 예외

```
package ch05.sec04;

public class NullPointerExceptionExample {
    public static void main(String[] args) {
        int[] intArray = null;
        //intArray[0] = 10; //NullPointerException

        String str = null;
        //System.out.println("총 문자 수: " + str.length())
    };//NullPointerException
    }
}
```

경우에 따라서 참조 타입 변수에 일부러 null을 대입하기도 하는데, 그렇게 되면 번지를 잃게 되므로 더 이상 객체를 사용할 수 없게 된다. 힙 메모리에는 있지만, 위치 정보를 모르기 때문에 사용할 수 없게 되는데 자바는 이러한 객체를 쓰레기로 취급하고, **쓰레기 수집기(Garbage Collector)**를 실행시켜 자동으로 제거한다. 자바는 코드를 이용해서 객체를 직접 제거할 수 없기 때문에 객체를 제거하는 유일한 방법은 *객체의 모든 참조를 없애는 것*이다.

```
package ch05.sec04;

public class GarbageObjectExample {
    public static void main(String[] args) {
        String hobby = "여행";
        hobby = null; // "여행"에 해당하는 String 객체를 쓰레기로 만들

        String kind1 = "자동차";
        String kind2 = kind1; // kind1 변수에 저장되어 있는 번지를 kind2 변수에 대입
        kind1 = null; // "자동차"에 해당하는 String 객체는 쓰레기가 아님
        System.out.println("kind2: " + kind2);
    }
}
```

5.5 문자열(String) 타입

문자열 비교

- 자바는 문자열 리터럴이 동일하다면 String 객체를 공유
- new 연산자(객체 생성 연산자)로 String 객체를 생성하고 대입할 경우 변수는 서로 다른 String 객체의 번지 갖게 됨
- 동일한 String 객체든 다른 String 객체든 상관없이 내부 문자열만을 비교할 경우 String 객체의 **equals()** 메소드 사용

```
package ch05.sec05;

public class EqualsExample {
    public static void main(String[] args) {
        String strVar1 = "홍길동";
        String strVar2 = "홍길동";

        if(strVar1 == strVar2) {
            System.out.println("strVar1과 strVar2는 참조가 같음");
        } else {
            System.out.println("strVar1과 strVar2는 참조가 다름");
        }

        if(strVar1.equals(strVar2)) {
            System.out.println("strVar1과 strVar2는 문자열이 같음");
        }

        String strVar3 = new String("홍길동");
```

```
String strVar4 = new String("홍길동");

if(strVar3 == strVar4) {
    System.out.println("strVar3과 strVar4는 참조가 같음");
} else {
    System.out.println("strVar3과 strVar4는 참조가 다름");
}

if(strVar3.equals(strVar4)) {
    System.out.println("strVar3과 strVar4는 문자열이 같음");
}
}
```

결과

```
strVar1과 strVar2는 참조가 같음
strVar1과 strVar2는 문자열이 같음
strVar3과 strVar4는 참조가 다름
strVar3과 strVar4는 문자열이 같음
```

문자 추출

- charAt()

```
package ch05.sec05;

public class CharAtExample {
    public static void main(String[] args) {
        String ssn = "9506241230123";
        char sex = ssn.charAt(6);
        switch (sex) {
            case '1':
            case '3':
                System.out.println("남자입니다.");
                break;
            case '2':
            case '4':
                System.out.println("여자입니다.");
                break;
        }
    }
}
```

문자열 길이

- length()

```
package ch05.sec05;

public class LengthExample {
    public static void main(String[] args) {
        String ssn = "9506241230123";
        int length = ssn.length();
        if(length == 13) {
            System.out.println("주민등록번호 자릿수가 맞습니다.");
        } else {
            System.out.println("주민등록번호 자릿수가 틀립니다.");
        }
    }
}
```

문자열 대체

- replace()

```
package ch05.sec05;

public class ReplaceExample {
    public static void main(String[] args) {
        String oldStr = "자바 문자열은 불변입니다. 자바 문자열은 String입니다.";
        String newStr = oldStr.replace("자바", "JAVA");

        System.out.println(oldStr);
        System.out.println(newStr);
    }
}
```

결과

자바 문자열은 불변입니다. 자바 문자열은 String입니다.
JAVA 문자열은 불변입니다. JAVA 문자열은 String입니다.

문자열 잘라내기

- substring()

```
package ch05.sec05;

public class SubStringExample {
    public static void main(String[] args) {
        String ssn = "880815-1234567";
    }
}
```

```
String firstNum = ssn.substring(0, 6);
System.out.println(firstNum);

String secondNum = ssn.substring(7);
System.out.println(secondNum);
    }
}
```

결과

```
880815
1234567
```

문자열 찾기

- indexOf() -> 주어진 문자열이 시작되는 인덱스 리턴 (없을 경우 -1 리턴)
- contains() -> true/false 리턴

```
package ch05.sec05;

public class IndexOfContainsExample {
    public static void main(String[] args) {
        String subject = "자바 프로그래밍";

        int location = subject.indexOf("프로그래밍");
        System.out.println(location);
        String substring = subject.substring(location);
        System.out.println(substring);

        location = subject.indexOf("자바");
        if(location != -1) {
            System.out.println("자바와 관련된 책이군요.");
        } else {
            System.out.println("자바와 관련 없는 책이군요.");
        }

        boolean result = subject.contains("자바");
        if(result) {
            System.out.println("자바와 관련된 책이군요.");
        } else {
            System.out.println("자바와 관련 없는 책이군요.");
        }
    }
}
```

결과

3
프로그래밍
자바와 관련된 책이군요.
자바와 관련된 책이군요.

문자열 분리

- split()

```
package ch05.sec05;

public class SplitExample {
    public static void main(String[] args) {
        String board = "1,자바 학습,참조 타입 String을 학습합니다.,홍길동";

        //문자열 분리
        String[] tokens = board.split(",");

        //인덱스별로 읽기
        System.out.println("번호: " + tokens[0]);
        System.out.println("제목: " + tokens[1]);
        System.out.println("내용: " + tokens[2]);
        System.out.println("성명: " + tokens[3]);
        System.out.println();

        //for 문을 이용한 읽기
        for(int i=0; i<tokens.length; i++) {
            System.out.println(tokens[i]);
        }
    }
}
```

결과

번호: 1
제목: 자바 학습
내용: 참조 타입 String을 학습합니다.
성명: 홍길동

1
자바 학습
참조 타입 String을 학습합니다.
홍길동

5.6 배열(Array) 타입

- 배열은 같은 타입의 값만 관리한다.
- 배열의 길이는 늘리거나 줄일 수 없다.

값 목록으로 배열 생성

```
타입[] 변수 = { 값0, 값1, 값2, 값3, ... };
```

중괄호 {}는 나열된 값들을 항목으로 가지는 배열을 힙에 생성하고, 번지를 리턴한다. 배열 변수는 리턴된 번지를 저장함으로써 참조가 이루어진다.

배열 변수를 미리 선언한 후에는 중괄호{}로 감싼 값 목록을 변수에 대입할 수 없다. 배열 변수를 선언한 시점과 값 목록이 대입되는 시점이 다르다면 new 타입[]을 중괄호 앞에 붙여줘야 한다.

```
변수 = new 타입[] { 값0, 값1, 값2, 값3, ... };
```

이는 메소드의 매개변수가 배열 타입일 경우에도 마찬가지다.

```
printItem(new int[] {95, 85, 90});
```

```
package ch05.sec06;

public class ArrayCreateByValueListExample2 {
    public static void main(String[] args) {
        //배열 변수 선언
        int[] scores;
        //배열 변수에 배열을 대입
        scores = new int[] { 83, 90, 87 };
        //배열 항목의 총합을 구하고 출력
        int sum1 = 0;
        for(int i=0; i<3; i++) {
            sum1 += scores[i];
        }
        System.out.println("총합 : " + sum1);

        //배열을 매개값으로 주고, printItem() 메소드 호출
        printItem( new int[] { 83, 90, 87 } );
    }

    //printItem() 메소드 선언
    public static void printItem( int[] scores ) {
        //매개변수가 참조하는 배열의 항목을 출력
        for(int i=0; i<3; i++) {
            System.out.println("score[" + i + "]: " + scores[i]);
        }
    }
}
```



```
}
}
```

new 연산자로 배열 설정

- 값의 목록은 없지만 향후 값들을 저장할 목적으로 배열 미리 생성 가능
- 이미 배열 변수가 선언된 후에도 null을 대입한 후 new로 다른 길이의 새 배열 생성 가능
- new 연산자로 배열을 처음 생성하면 배열 항목은 기본값으로 초기화
 - 정수 배열 : 0
 - 실수 배열 : 0.0
 - 논리 배열 : false
 - 참조 배열 : null

배열 길이

- 배열변수.length
- 읽기만 가능
- 인덱스 초과해서 사용할 시 *ArrayIndexOutOfBoundsException* 발생

5.7 다차원 배열

- 배열 항목에 또 다른 배열이 대입

값 목록으로 다차원 배열 생성

```
package ch05.sec07;

public class MultidimensionalArrayByValueListExample {
    public static void main(String[] args) {
        //2차원 배열 생성
        int[][] scores = {
            { 80, 90, 96 },
            { 76, 88 }
        };

        //배열의 길이
        System.out.println("1차원 배열 길이(반의 수): " + scores.length);
        System.out.println("2차원 배열 길이(첫 번째 반의 학생 수): " +
            scores[0].length);
        System.out.println("2차원 배열 길이(두 번째 반의 학생 수): " +
            scores[1].length);

        //첫 번째 반의 세 번째 학생의 점수 읽기
        System.out.println("scores[0][2]: " + scores[0][2]);

        //두 번째 반의 두 번째 학생의 점수 읽기
        System.out.println("scores[1][1]: " + scores[1][1]);
    }
}
```

```

//첫 번째 반의 평균 점수 구하기
int class1Sum = 0;
for(int i=0; i<scores[0].length; i++) {
    class1Sum += scores[0][i];
}
double class1Avg = (double) class1Sum / scores[0].length;
System.out.println("첫 번째 반의 평균 점수: " + class1Avg);

//두 번째 반의 평균 점수 구하기
int class2Sum = 0;
for(int i=0; i<scores[1].length; i++) {
    class2Sum += scores[1][i];
}
double class2Avg = (double) class2Sum / scores[1].length;
System.out.println("두 번째 반의 평균 점수: " + class2Avg);

//전체 학생의 평균 점수 구하기
int totalStudent = 0;
int totalSum = 0;
for(int i=0; i<scores.length; i++) {
    totalStudent += scores[i].length;
    for(int k=0; k<scores[i].length; k++) {
        totalSum += scores[i][k];
    }
}
double totalAvg = (double) totalSum / totalStudent;
System.out.println("전체 학생의 평균 점수: " + totalAvg);
}
}

```

결과

```

1차원 배열 길이(반의 수): 2
2차원 배열 길이(첫 번째 반의 학생 수): 3
2차원 배열 길이(두 번째 반의 학생 수): 2
scores[0][2]: 96
scores[1][1]: 88
첫 번째 반의 평균 점수: 88.66666666666667
두 번째 반의 평균 점수: 82.0
전체 학생의 평균 점수: 86.0

```

new 연산자로 다차원 배열 생성

- 2차원 배열의 길이 다르게 주기

```

int[][] scores = new int[2][];
scores[0] = new int[3]; // 첫 번째 반의 학생 수가 3명
scores[1] = new int[2]; // 두 번째 반의 학생 수가 2명

```

```

package ch05.sec07;

public class MultidimensionalArrayByNewExample {
    public static void main(String[] args) {
        //각 반의 학생 수가 3명으로 동일할 경우 점수 저장을 위한 2차원 배열 생성
        int[][] mathScores = new int[2][3];
        //배열 항목 초기값 출력
        for (int i = 0; i < mathScores.length; i++) {                //반의 수만큼
반복
            for (int k = 0; k < mathScores[i].length; k++) {          // 해당 반의 학
생 수만큼 반복
                System.out.println("mathScores[" + i + "][" + k + "]: " +
mathScores[i][k]);
            }
        }
        System.out.println();
        //배열 항목 값 변경
        mathScores[0][0] = 80;
        mathScores[0][1] = 83;
        mathScores[0][2] = 85;
        mathScores[1][0] = 86;
        mathScores[1][1] = 90;
        mathScores[1][2] = 92;
        //전체 학생의 수학 평균 구하기
        int totalStudent = 0;
        int totalMathSum = 0;
        for (int i = 0; i < mathScores.length; i++) {                //반의 학생 수
합산
            for (int k = 0; k < mathScores[i].length; k++) {          //해당 반의 학
생 수만큼 반복
                totalMathSum += mathScores[i][k];                    //학생 점수 합
산
            }
        }
        double totalMathAvg = (double) totalMathSum / totalStudent;
        System.out.println("전체 학생의 수학 평균 점수: " + totalMathAvg);
        System.out.println();

        //각 반의 학생 수가 다를 경우 점수 저장을 위한 2차원 배열 생성
        int[][] englishScores = new int[2][];
        englishScores[0] = new int[2];
        englishScores[1] = new int[3];
        //배열 항목 초기값 출력
        for (int i = 0; i < englishScores.length; i++) {              //반의 수만큼
반복
            for (int k = 0; k < englishScores[i].length; k++) {        // 해당 반의 학
생 수만큼 반복
                System.out.println("englishScores[" + i + "][" + k + "]: " +
englishScores[i][k]);
            }
        }
    }
}

```

```

        System.out.println();
        //배열 항목 값 변경
        englishScores[0][0] = 90;
        englishScores[0][1] = 91;
        englishScores[1][0] = 92;
        englishScores[1][1] = 93;
        englishScores[1][2] = 94;
        //전체 학생의 영어 평균 구하기
        totalStudent = 0;
        int totalEnglishSum = 0;
        for (int i = 0; i < englishScores.length; i++) {
반복
            totalStudent += englishScores[i].length;
합산
            for (int k = 0; k < englishScores[i].length; k++) {
생 수만큼 반복
                totalEnglishSum += englishScores[i][k];
산
            }
        }
        double totalEnglishAvg = (double) totalEnglishSum / totalStudent;
        System.out.println("전체 학생의 영어 평균 점수: " + totalEnglishAvg);
    }
}

```

결과

```

mathScores[0][0]: 0
mathScores[0][1]: 0
mathScores[0][2]: 0
mathScores[1][0]: 0
mathScores[1][1]: 0
mathScores[1][2]: 0

전체 학생의 수학 평균 점수: 86.0

englishScores[0][0]: 0
englishScores[0][1]: 0
englishScores[1][0]: 0
englishScores[1][1]: 0
englishScores[1][2]: 0

전체 학생의 영어 평균 점수: 92.0

```

5.8 객체를 참조하는 배열

```
package ch05.sec08;
```

```

public class ArrayReferenceObjectExample {
    public static void main(String[] args) {
        String[] strArray = new String[3];
        strArray[0] = "Java";
        strArray[1] = "Java";
        strArray[2] = new String("Java");

        System.out.println( strArray[0] == strArray[1] );           //true: 같은 객체
        System.out.println( strArray[0] == strArray[2] );           //false: 다른 객체
        System.out.println( strArray[0].equals(strArray[2]) );     //true: 문자열이 동

참조
를 참조
일
    }
}

```

5.9 배열 복사

- `for` 문을 이용한 가장 기본적인 복사 방법

```

package ch05.sec09;

public class ArrayCopyByForExample {
    public static void main(String[] args) {
        //길이 3인 배열
        int[] oldIntArray = { 1, 2, 3 };
        //길이 5인 배열을 새로 생성
        int[] newIntArray = new int[5];
        //배열 항목 복사
        for(int i=0; i<oldIntArray.length; i++) {
            newIntArray[i] = oldIntArray[i];
        }
        //배열 항목 출력
        for(int i=0; i<newIntArray.length; i++) {
            System.out.print(newIntArray[i] + ", ");
        }
    }
}

```

- `System`의 `arraycopy()` 메소드 이용

```

package ch05.sec09;

public class ArrayCopyExample {
    public static void main(String[] args) {
        //길이 3인 배열
        String[] oldStrArray = { "java", "array", "copy" };
        //길이 5인 배열을 새로 생성
    }
}

```

```
String[] newStrArray = new String[5];
//배열 항목 복사
System.arraycopy( oldStrArray, 0, newStrArray, 0, oldStrArray.length);
//배열 항목 출력
for(int i=0; i<newStrArray.length; i++) {
    System.out.print(newStrArray[i] + ", ");
}
}
```

System.arraycopy(원본배열, 원본배열 복사 시작인덱스, 새배열, 새배열 붙여넣기 시작인덱스, 복사 항목 수)

5.10 배열 항목 반복을 위한 향상된 for 문

```
package ch05.sec10;

public class AdvancedForExample {
    public static void main(String[] args) {
        //배열 변수 선언과 배열 생성
        int[] scores = { 95, 71, 84, 93, 87 };
        //배열 항목 전체 합 구하기
        int sum = 0;
        for (int score : scores) {
            sum = sum + score;
        }
        System.out.println("점수 총합 = " + sum);
        //배열 항목 전체 평균 구하기
        double avg = (double) sum / scores.length;
        System.out.println("점수 평균 = " + avg);
    }
}
```

5.11 main() 메소드의 String[] 매개변수 용도

```
package ch05.sec11;

public class MainStringArrayArgument {
    public static void main(String[] args) {
        if(args.length != 2) {
            System.out.println("프로그램 입력값이 부족");
            System.exit(0);
        }

        String strNum1 = args[0];
        String strNum2 = args[1];

        int num1 = Integer.parseInt(strNum1);
```

```

        int num2 = Integer.parseInt(strNum2);

        int result = num1 + num2;
        System.out.println(num1 + " + " + num2 + " = " + result);
    }
}

```

이클립스에서 입력값을 주고 실행하거나 명령 프롬프트/터미널에서 입력값 주기

5.12 열거(Enum) 타입

- enumeration type
- 한정된 값을 갖는 타입
- 열거 타입 이름으로 소스 파일(.java) 생성 후 한정된 값을 코드로 정의
- 첫 문자는 대문자로, 캐멀 스타일이 관례

```

package ch05.sec12;

public enum Week {
    MONDAY,
    TUESDAY,
    WEDNESDAY,
    THURSDAY,
    FRIDAY,
    SATURDAY,
    SUNDAY
}

```

- 열거 상수 : 열거 타입으로 사용할 수 있는 한정된 값.
- 알파벳으로 정의, 모두 대문자로 작성
- 여러 단어일 경우 언더바(_)로 연결

```

package ch05.sec12;

import java.util.Calendar;

public class WeekExample {
    public static void main(String[] args) {
        //Week 열거 타입 변수 선언
        Week today = null;

        //Calendar 얻기
        Calendar cal = Calendar.getInstance();

        //오늘의 요일 얻기(1~7)
        int week = cal.get(Calendar.DAY_OF_WEEK);

        //숫자를 열거 상수로 변환해서 변수에 대입
    }
}

```

```
switch(week) {  
    case 1: today = Week.SUNDAY ; break;  
    case 2: today = Week.MONDAY ; break;  
    case 3: today = Week.TUESDAY ; break;  
    case 4: today = Week.WEDNESDAY ; break;  
    case 5: today = Week.THURSDAY ; break;  
    case 6: today = Week.FRIDAY ; break;  
    case 7: today = Week.SATURDAY ; break;  
}  
  
//열거 타입 변수를 사용  
if(today == Week.SUNDAY) {  
    System.out.println("일요일에는 축구를 합니다.");  
} else {  
    System.out.println("열심히 자바를 공부합니다.");  
}  
}
```

결과

열심히 자바를 공부합니다.