

예제 7-1 : 정수만 다루는 Vector<Integer> 컬렉션 활용

1

정수만 다루는 벡터를 생성하고, 활용하는 기본 사례를 보인다.

➔ 다음 프로그램에서 Vector를 3개로 생성하도록 변경하고 4번 정수를 입력 받도록 수정 하시오. 단, 4번째 입력 받은 정수는 3번째에 삽입하도록 하시오 (나머지는 동일하게 수행)

```
import java.util.Vector;

public class VectorEx {
    public static void main(String[] args) {
        // 정수 값만 다루는 제네릭 벡터 생성
        Vector<Integer> v = new Vector<Integer>();

        v.add(5); // 5 삽입
        v.add(4); // 4 삽입
        v.add(-1); // -1 삽입

        // 벡터 중간에 삽입하기
        v.add(2, 100); // 4와 -1 사이에 정수 100 삽입

        System.out.println("벡터 내의 요소 객체 수 : " + v.size());
        System.out.println("벡터의 현재 용량 : " + v.capacity());

        // 모든 요소 정수 출력하기
        for(int i=0; i<v.size(); i++) {
            int n = v.get(i);
            System.out.println(n);
        }
    }
}
```

```
// 벡터 속의 모든 정수 더하기
int sum = 0;
for(int i=0; i<v.size(); i++) {
    int n = v.elementAt(i);
    sum += n;
}
System.out.println("벡터에 있는 정수 합 : "
                    + sum);
}
```

예제 7-2 : Point 클래스만 다루는 Vector<Point> 컬렉션 활용

2

점 (x, y)를 표현하는 Point 클래스를 만들고, Point의 객체만 다루는 벡터를 작성하라.

➔ Point 클래스에서 Point(x, y) 형태로 출력하도록 수정하시오. 또한, x, y 값을 3개 입력 받아 Point 객체를 생성하도록 수정하시오

➔ 모든 Point 객체를 Vector에 입력한 후 Point 객체 정보를 출력하고 마지막 요소 삭제 후 다시 모든 Point 객체 정보를 출력하도록 수정하시오

```
import java.util.Vector;
```

```
class Point {
```

```
    private int x, y;
```

```
    public Point(int x, int y) {
```

```
        this.x = x;
```

```
        this.y = y;
```

```
    }
```

```
    public String toString() {
```

```
        return "(" + x + "," + y + ");"
```

```
    }
```

```
}
```

```
public class PointVectorEx {
```

```
    public static void main(String[] args) {
```

```
        // Point 객체를 요소로만 가지는 벡터 생성
```

```
        Vector<Point> v = new Vector<Point>();
```

```
        // 3 개의 Point 객체 삽입
```

```
        v.add(new Point(2, 3));
```

```
        v.add(new Point(-5, 20));
```

```
        v.add(new Point(30, -8));
```

```
        v.remove(1); // 인덱스 1의 Point(-5, 20) 객체 삭제
```

```
        // 벡터에 있는 Point 객체 모두 검색하여 출력
```

```
        for(int i=0; i<v.size(); i++) {
```

```
            Point p = v.get(i); // 벡터에서 i 번째 Point 객체 얻어내기
```

```
            System.out.println(p); // p.toString()을 이용하여 객체 p 출력
```

```
        }
```

```
    }
```

```
}
```

예제 7-3 : 문자열 입력받아 ArrayList에 저장

3

이름을 4개 입력받아 ArrayList에 저장하고 모두 출력한 후 제일 긴 이름을 출력하라.

➔ 제일 긴 이름과 제일 짧은 이름을 출력하고, 몇 번째 입력된 것인지도 같이 출력하도록 수정하시오

```
import java.util.*;

public class ArrayListEx {
    public static void main(String[] args) {
        // 문자열만 삽입가능한 ArrayList 컬렉션 생성
        ArrayList<String> a = new ArrayList<String>();

        // 키보드로부터 4개의 이름 입력받아 ArrayList에 삽입
        Scanner scanner = new Scanner(System.in);
        for(int i=0; i<4; i++) {
            System.out.print("이름을 입력하세요>>");
            String s = scanner.next(); // 키보드로부터 이름 입력
            a.add(s); // ArrayList 컬렉션에 삽입
        }

        // ArrayList에 들어 있는 모든 이름 출력
        for(int i=0; i<a.size(); i++) {
            // ArrayList의 i 번째 문자열 얻어오기
            String name = a.get(i);
            System.out.print(name + " ");
        }
    }
}
```

```
// 가장 긴 이름 출력
int longestIndex = 0;
for(int i=1; i<a.size(); i++) {
    if(a.get(longestIndex).length() < a.get(i).length())
        longestIndex = i;
}
System.out.println("\n가장 긴 이름은 : " +
    a.get(longestIndex));
}
scanner.close();
}
```