

중간고사

담당교수: 단국대학교 응용컴퓨터공학 박경신

- 답은 반드시 답안지에 기술할 것. 공간이 부족할 경우 반드시 답안지 몇 쪽의 뒤에 있다고 명기한 후 기술할 것. 그 외의 경우의 답안지 뒤쪽이나 연습지에 기술한 내용은 답안으로 인정 안 함. 답에는 반드시 네모를 쳐서 확실히 표시할 것.
- 답안지에 학과, 학번, 이름 외에 본인의 암호(4자리 숫자)를 기입하면 성적공고시 학번대신 암호를 사용할 것임.

1. 다음은 클래스 상속관계에서 다형성을 보여주고 있다. 아래의 질문에 답하라. (25점)

```
interface I {
    void method1(int v);
}
interface J {
    void method2();
}
abstract class A implements I, J {
    protected int m = 20;
    public void method3() {
        System.out.println(this);
    }
    public String toString() {
        return "a";
    }
}
class B extends A {
    public void method1(int v) {
        System.out.println("b1 " + v);
    }
    public void method2() {
        System.out.println("b2");
    }
}
class C extends A {
    protected int m = 10;
    public void method1(int v) {
        System.out.println("c1 " + v);
    }
    public void method2() {
        System.out.println("c2 " + m);
    }
    public String toString() {
        return "c";
    }
}
class D extends C {
    public void method2() {
        System.out.println("d2 " + this.m + " " + super.m);
    }
}
```

1.1 interface와 abstract class가 무엇인지 간단히 설명하라. (5점)

인터페이스는 클래스에서 구현해야 하는 동작을 지정하는데 사용하는 추상형. 자바에서 인터페이스는 다중 상속이 가능하다.

추상 클래스는 여러 기존의 클래스에서 공통된 부분을 추상화한 것. 추상 클래스를 상속하는 클래스에게 그 구현을 강제화하는 클래스. 하나 이상의 추상 메소드를 포함하는 클래스이다.

1.2 functional interface가 무엇인지 간단히 설명하라. 위의 코드에서 예시를 찾아라. (5점)

함수형 인터페이스는 추상 메소드를 단 하나만 포함하는 인터페이스로, 랴다식과 함께 사용할 수 있다.

위의 코드에서는 I과 J 인터페이스가 함수형 인터페이스이다.

1.3 다음 코드에서, 문제가 있는 부분을 찾아서 직접 표시하고 이유를 적어라. (5점)

```
A e1 = new A(); // compile error; cannot create A abstract class object
A e2 = new B();
D e3 = new C(); // compile error; cannot convert from C to D
I e4 = new I() {
    public void method1(int v) {
        System.out.println("b1 " + v);
    }
};
J e5 = new J(); // compile error; cannot create J interface object
```

1.4 다음 코드의 실행결과를 적고 이유를 자세히 설명하라 (dynamic binding에 주의). (10점)

```
A[] elements = {new C(), new B(), new D()};
for (int i = 0; i < elements.length; i++) {
    System.out.println(elements[i]);
    elements[i].method1(i);
    elements[i].method2();
    elements[i].method3();
}
```

```
c // this는 C => this.toString()은 c
c1 0 // C.method1(0)
c2 10 // C.method2()
c // C.method3() 없으니까 => A.method3() 호출하는데 => this는 C => C.toString()
a // this는 B => this.toString()은 B.toString() 없으니까 => A.toString() 호출해서 a
b1 1 // B.method1(1)
b2 20 20 // B.method2() this.m = super.m = 20
a // B.method3() 없으니까 => A.method3() 호출하는데 => this는 B => B.toString() 없으니까 => A.toString()
c // this는 D => this.toString()은 D.toString() 없으니까 => C.toString()
c1 2 // D.method1(2) 없으니까 => C.method1(2)
d2 10 10 // D.method2() this.m = super.m = 10
c // D.method3() 없으니까 => C.method3() 호출하는데 => this는 D => D.toString() 없으니까 => C.toString()
```

2. 다음 E 클래스의 start() 메소드 호출 시 실행결과를 적고 동작 원리를 설명하라. (10점)

```
class E {
    private int x, y;
    public E() {
        x = 3; y = 5; // (1) x = 3, y = 5
    }
    private void set(int s, int y) { // (5) s = 11, t = -1
        x = s; this.y = y; // (6) x = 11, y = -1
        first(y); // (8) x = -1, y = 0
    }
    private void first(int z) { // (2) z = 5 // (7) z = -1
        x = z; y++; // (3) x = 5, y = 6 // (8) x = -1, y = 0
    }
    private void second(int s, int t) { // (4) s = 6, t = 5
        set((y + t), (x - s)); // (8) x = -1, y = 0
        s = 1; t = 3; // (9) s = 1, t = 3
    }
    private void print() {
        System.out.println("x = " + x + " y = " + y);
    }
    public void start() {
        first(y); // (3) x = 5, y = 6
        second(y, x); // (8) x = -1, y = 0
        print(); // (10) x = -1, y = 0
    }
}
```

최종 실행 결과 : x = -1 y = 0

3. 다음 질문에 답하라 (70점)

```
public class MainTest {
    static <T> void print(T[] arr) { // arr = {1, 2, 3}
        System.out.println(Arrays.toString(arr)); // [1, 2, 3]
    }
    static <T> void print(List<T> list) { // list = {1, 2, 3}
        System.out.println(Arrays.toString(list.toArray())); // [1, 2, 3]
    }
    static void print(String filename) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader(filename));
        String line = "";
        while ((line = br.readLine()) != null) {
            String[] items = line.split(",");
            print(items);
        }
        br.close();
    }
}
```

```

public static void main(String[] args) {
    // (3.1) Integer Array
    Integer[] intArray = new Integer[5];
    intArray[0] = 3;
    intArray[4] = -18;
    intArray[2] = 5;
    intArray[1] = intArray[2];
    int x = intArray[0];
    intArray[2] = 6;
    intArray[x] = intArray[1] * intArray[4];
    print(intArray);
    // (3.2) Why UnsupportedOperationException?
    //List<Integer> intList= Arrays.asList(intArray);
    List<Integer> intList= new ArrayList<Integer>();
    for (Integer v : intArray)
        intList.add(v);
    intList.add(5);
    intList.add(7);
    print(intList);
    // (3.3) Write a code to remove 5, 7 in the intList
    Iterator<Integer> iter = intList.iterator();
    while (iter.hasNext()) {
        Integer v = iter.next();
        if (v == 5 || v == 7) iter.remove();
    }
    // (3.4) lambda & stream & method reference
    List<Character> charList= Arrays.asList('a', 'b', '2', '3', 'c', '1');
    List<Character> a = charList.stream().filter(Character::isAlphabetic)
                                .map(Character::toUpperCase)
                                .collect(Collectors.toList());
    print(a);
    List<Integer> b = charList.stream().filter(v -> Character.isDigit(v))
                                .map(v -> Character.getNumericValue(v)
                                    * Character.getNumericValue(v))
                                .sorted();
                                .collect(Collectors.toList());
    print(b);
}

```

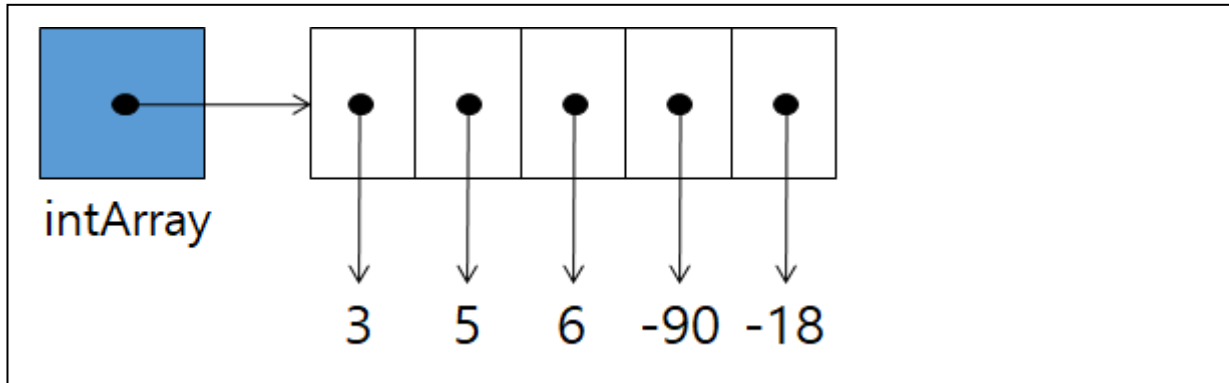
학과 _____

학번 _____

이름 _____

```
// (3.5) == & equals & hashCode
Integer oi = 1000;
Integer oj = 1000;
Integer ok = oi;
System.out.println("oi == oj " + (oi == oj));
System.out.println("oi == ok " + (oi == ok));
System.out.println("oi equals oj " + oi.equals(oj));
System.out.println("oi equals ok " + oi.equals(ok));
System.out.println("oi oj hashCode" + (oi.hashCode() == oj.hashCode()));
System.out.println("oi ok hashCode" + (oi.hashCode() == ok.hashCode()));
E ei = new E(); // 문제 2의 class E 사용
E ej = new E();
E ek = ei;
System.out.println("ei == ej " + (ei == ej));
System.out.println("ei == ek " + (ei == ek));
System.out.println("ei equals ej " + ei.equals(ej));
System.out.println("ei equals ek " + ei.equals(ek));
System.out.println("ei ej hashCode" + (ei.hashCode() == ej.hashCode()));
System.out.println("ei ek hashCode" + (ei.hashCode() == ek.hashCode()));
// (3.6) HashSet & HasMap
Set<Integer> intSet = new HashSet<Integer>();
intSet.add(oi);
intSet.add(oj);
intSet.add(ok);
intSet.forEach(v -> System.out.println(v));
Map<Integer, String> isMap = new HashMap<Integer, String>();
isMap.put(oi, "oi");
isMap.put(oj, "oj");
isMap.put(ok, "ok");
isMap.forEach((key, value) -> System.out.println(key + " " + value));
// (3.7) FileIO
try {
    print("test.csv");
} catch (IOException e) {
    e.printStackTrace();
}
}
}
```

3.1 main()에 (3.1) 코드가 실행되었을 때 실행결과 배열을 그려라. (10점)



3.2 main()에 (3.2) 코드는 `UnsupportedOperationException`이란 Runtime Exception이 발생한다. 그 이유를 적고, 이 예외상황 발생하지 않도록 밑줄 친 코드를 수정하라. (10점)

`Arrays.asList(intArray)`를 사용하면 immutable한 collection이 생성되므로 add를 할 수 없어서 `UnsupportedOperationException`이 발생한다.

밑줄친 부분을 아래와 같이 mutable한 collection으로 생성해야 한다.

```
List<Integer> intList= new ArrayList<Integer>();
for (Integer v : intArray)
    intList.add(v);
```

3.3 main()에 (3.3)에서는 수정한 (3.2) 코드 `intList`에서 5, 7 요소를 지우는 코드를 작성하라. (10점)

```
Iterator<Integer> iter = intList.iterator();
while (iter.hasNext()) {
    Integer v = iter.next();
    if (v == 5 || v == 7) iter.remove();
}
```

3.4 main()에 (3.4)는 람다와 스트림을 활용한 코드이다. 그 실행 결과를 적고 이유를 설명하라. (10점)

`[A, B, C]` // `charList`로부터 alphabet를 filter해서 UpperCase로 map해서 새로운 Character 리스트 생성

`[1, 4, 9]` // `charList`로부터 digit을 filter해서 Integer로 변환하여 제곱을 한 후 정렬하여 새로운 Integer 리스트 생성

3.5 main()에 (3.5)는 == 과 equals 와 hashCode를 활용한 코드이다. 그 실행 결과를 적고 이유를 설명하라. (10점)

```
oi == oj false // 다른 reference
oi == ok true // 동일한 reference
oi equals oj true // 내용은 같다
oi equals ok true // 동일한 reference에 동일한 내용
oi oj hashCode true // 같은 hashCode 값
oi ok hashCode true // 동일한 reference에 동일한 내용, 동일한 hashCode 값
ei == ej false // 다른 reference
ei == ek true // 동일한 reference
ei equals ej false // equals override가 없으므로 내용 다르다
ei equals ek true // 동일한 reference에 동일한 내용
ei ej hashCode false // hashCode override가 없으므로 hashCode 값이 다르다
ei ek hashCode true // 동일한 reference에 동일한 내용, 동일한 hashCode 값
```

3.6 main()에 (3.6)는 HashSet과 HashMap을 활용한 코드이다. 그 실행 결과를 적고 이유를 설명하라. (10점)

```
1000 // HashSet은 동일한 hashCode 값을 갖는 element를 추가할 수 없다. 추가 시 마지막 것으로 대체됨.
1000 ok // HashMap은 동일한 key를 사용할 수 없다. 추가 시 마지막 것으로 대체됨.
```

3.7 main()에 (3.7)는 파일 입출력 코드이다. try/catch 구문을 사용해야 하는 이유를 설명하라. 그리고 본인이 "test.csv" 파일을 만들어서 실행 결과를 적고 이유를 설명하라. (10점)

```
try/catch 구문을 사용해야 하는 이유는 print(String filename) 메소드가 IOException을 throw하고 있기 때문에, 컴파일 하는 과정에서 예외 처리 코드가 필요한지 검사하는 try/catch 구문이 필요하다. 만약 try/catch 구문이 없다면 컴파일 오류를 발생한다.
print(String filename) 메소드는 파일을 읽어들여 ","로 tokenize하여 String[]로 만들어준 것을 출력

"test.csv"가 아래와 같을 시
Dooly,1000,40,126.4,MALE
A,B,C
1,2,3,4

실행결과는 아래와 같다.
[Dooly, 1000, 40, 126.4, MALE]
[A, B, C]
[1, 2, 3, 4]
```