**단국대학교 응용컴퓨터공학 JAVA 프로그래밍 II 기말고사 (2017년 봄학기) 2017년 6월 14일**

학과 _____ 학번 _____ 이름 _____

# 기말고사

**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. Value 클래스이다. 빈칸을 채워라. (10점)

```
import java.util.Comparator;
import java.util.Objects;

public class Value implements Comparable<Value> {
	private int a;
	private int b;

	public Value(int a, int b) {
		this.a = a;
		this.b = b;
	}
	public int getB() {
		return this.b; // b 를 반환
	}
	@Override
	public String toString() {
		return "(" + this.a + "," + this.b + ")"; // (a,b)로 출력
	}
	@Override
	public boolean equals(Object other) {
		if (this == other) return true;
		if (other instanceof Value) { // 만약 other 가 Value 객체타입이라면
			Value that = (Value)other;
			return Objects.equals(this.a, that.a)
				&& Objects.equals(this.b, that.b);
		}
		return false;
	}
	@Override
	public int hashCode() {
		return Objects.hash(this.a, this.b); // hash 값 생성
	}
	@Override
	public int compareTo(Value other) {
		return this.a - other.a; // a 로 비교
	}
}
```

학과 _____ 학번 _____ 이름 _____

## 2. Lab 클래스의 main() 메소드의 실행결과를 적어라. (10점)

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

		List<Value> vList = new ArrayList<Value>();
		vList.add(new Value(1, 2));
		vList.add(new Value(1, 2));
		vList.add(new Value(-3, 4));
		vList.add(new Value(-3, 4));
		vList.add(new Value(6, -5));
		vList.add(new Value(-7, -8));
		vList.add(new Value(-7, -8));
		vList.sort((Value v1, Value v2)-> v1.getB() - v2.getB());
		vList.forEach((v) -> System.out.println(v));
		vList.sort(null);
		vList.forEach((v) -> System.out.println(v));

		Map<Value, Integer> vMap = new HashMap<Value, Integer>();
		vMap.put(new Value(1, 2), 1);
		vMap.put(new Value(1, 2), 2);
		vMap.put(new Value(-3, 4), 3);
		vMap.put(new Value(-3, 4), 4);
		vMap.put(new Value(6, -5), 5);
		vMap.put(new Value(-7, -8), 6);
		vMap.put(new Value(-7, -8), 7);
		vMap.forEach((v, i) -> System.out.println(v + " => " + i));
	}
}
```

// B 로 정렬
(-7,-8)
(-7,-8)
(6,-5)
(1,2)
(1,2)
(-3,4)
(-3,4)

// vList.sort(null);은 Comparable<Value> compareTo를 이용한 A로 정렬
(-7,-8)
(-7,-8)
(-3,4)
(-3,4)
(1,2)
(1,2)
(6,-5)

// HashSet이나 HashMap은 동일한 key값을 허용하지 않는다. Value 클래스에 hashCode()가 재정의되어 있기 때문에 같은 내용을 갖는 Value는 동일한 hashCode를 가지므로 중복될 수 없다.
(-7,-8) => 7
(1,2) => 2
(6,-5) => 5
(-3,4) => 4

단국대학교 응용컴퓨터공학 JAVA 프로그래밍 II 기말고사 (2017년 봄학기) 2017년 6월 14일

학과 _____ 학번 _____ 이름 _____

**3. Thread를 생성하는 방법 2가지 extends Thread와 implements Runnable를 설명하고, 자바 예시코드를 보여라. (10점)**

1. extends Thread – Thread 클래스를 상속받아서 추상메소드인 public void run() { … } 내부를 구현하는 방법

```
public class Multithread extends Thread {
    // 중간 생략
    @Override
    public void run() {
        while (true) {
            count++;
        }
    }
}
Multithread th = new Multithread();
th.start();
```

2. implements Runnable – Runnable 인터페이스를 상속받아서 추상메소드인 public void run() { … } 내부를 구현하는 방법

```
public class MultithreadRunnable implements Runnable {
    // 중간 생략
    @Override
    public void run() {
        while (true) {
            count++;
        }
    }
}
Thread r = new Thread(new MultithreadRunnable());
r.start();
```

**4. Producer-Consumer Problem을 모니터(Monitor)로 구현한 코드를 보여준다. 빈칸을 채워라. (10점)**

```
public class BoundedBuffer {

        private volatile String[] buffer;
        private volatile int tail = 0;
        private volatile int head = 0;
        private volatile int count = 0;

        public BoundedBuffer(int size) {
                this.buffer = new String[size];
        }

        // Producer put
        public synchronized void put(String data) throws InterruptedException {
                while (count >= buffer.length) {
                        System.out.println("Full buffer: count=" + count);
                        wait(); // wait
                }
                System.out.println("producing " + data);
                buffer[tail] = data;
                tail = (tail + 1) % buffer.length;
                count++;
                notifyAll(); // signal
        }
```

학과 _____ 학번 _____ 이름 _____

```
        // Consumer take
        public synchronized String take() throws InterruptedException {
                while (count <= 0) {
                        System.out.println("\t\tEmpty buffer: count=" + count);
                        wait(); // wait
                }
                String data = buffer[head];
                head = (head + 1) % buffer.length;
                count--;
                System.out.println("\t\t consuming " + data);
                notifyAll(); // signal
                return data;
        }
}
```

**5. TCP(Transmission Control Protocol) 와 UDP(User Datagram Protocol)의 특징과 차이점을 설명하라. 자바 소켓 프로그래밍에 사용하는 ServerSocket 클래스, Socket 클래스, DatagramSocket 클래스에 대해 설명하라. (10점)**

TCP is a connection-oriented reliable stream transport protocol. 연결형 (전송 전에 연결을 맺어야 함) 신뢰형 (메시지 전송을 신뢰할 수 있음. 모든 데이터에 대한 승인이 있음. 패킷이 안정적으로 갈 수 있게 전송을 제어해주는 프로토콜.)

UDP is a connectionless unreliable datagram transport protocol. 비연결형 (연결 수립이 없이 데이터를 송신함) 비신뢰형 (메시지 전송을 신뢰할 수 없음. 승인이 없는 best-effort 전송 방식 프로토콜. )

ServerSocket 클래스는 서버 소켓으로 클라이언트 소켓이 연결됐는지 listen 하고 accept 함

Socket 클래스는 클라이언트 소켓으로 서버로 connect 하여 getInputStream/getOutputStream 로 데이터 전송

DatagramSocket 클래스는 datagram packet 을 send/receive 하는 클래스

-끝-