단국대학교 응용컴퓨터공학 JAVA 프로그래밍 II 기말고사 (2018년 가을학기) 2018년 12월 18일

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.

```
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 "+ this.m + " " + super.m);
    }
}
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 " + this.m + " " + super.m);
    }
    public String toString() {
        return "c";
    }
}
class D extends C {
    public void method2() {
        System.out.println("d2 " + this.m + " " + super.m);
    }
}
```

학과 _____ 학번 _____ 이름 _____

```
class E extends Thread {
    private Map<String, String> map = new HashMap<String, String>();
    public E(Map<String, String> map) {
        this.map = map;
    }
    public E(String filename) {
        load(filename); // (1)
    }
    public void load(String filename) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader(filename));
        String line = "";
        while ((line = br.readLine()) != null) {
              String[] items = line.split(",");
              map.put(items[0], items[1]);
       }
       br.close();
    }
    public void run() {
        map.forEach((k, v) -> System.out.println(k + " " + v));
    }
}

public class MainTest {

      public static void main(String[] args) {

            // (2)
            A e1 = new A();
            A e2 = new B();
            A e3 = new D();
            B e4 = e2;
            B e5 = e3;
            D e6 = new C();
            I e7 = (v) -> System.out.println("e7 " + v);
            J e8 = () -> System.out.println("e8 " + m);
            J e9 = new D();
            A e10 = new A() {
                  public void method1(int v) {
                    System.out.println("e10 " + v);
                  }
                  public void method2() {
                    System.out.println("e10 " + m);
                  }
            };

            // (3)
            A[] elements = {new C(), new B(), new D(), e10};
            for (int i = 0; i < elements.length; i++) {
                  System.out.println(elements[i]);
                  elements[i].method1(i);
                  elements[i].method2();
                  elements[i].method3();
                  }
```

학과 _____ 학번 _____ 이름 _____

```
            // (5)
            List<Integer> intList= Arrays.asList(5, 3, 4, 7, 2);
            List<String> a = intList.stream()
                                    .filter(i -> i > 3)
                                    .map(i -> "" + i * i)
                                    .collect(Collectors.toList());
            int b = IntStream.range(1, 10)
                                    .filter(i -> i%2 != 0)
                                    .map(i -> i * i)
                                    .reduce(0, (i, j) -> i + j);
            String c = _____// c = "53472"

            // (6)
            new E("test1.csv").start();
            new E("test2.csv").start();
            new E("test3.csv").start();
        }
}
```

단국대학교 응용컴퓨터공학 JAVA 프로그래밍 II 기말고사 (2018년 가을학기) 2018년 12월 18일

학과 _____ 학번 _____ 이름 _____

1. 클래스 E 생성자 내부코드는 compile error를 발생한다. 그 이유를 설명하고, 컴파일 에러가 없어지도록 코드를 수정하라. (10점)

```
public E(String filename) {
    try {
        load(filename); // (1) Unhandled Exception Type IOException
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

IOException은 일반예외(즉, 컴파일 체크 exception)이므로 컴파일 에러를 발생한다.
try/catch 구문을 추가하여 컴파일 에러를 없앨 수 있다.

2. 위의 main() 메소드에서는 e1부터 e10까지 객체를 생성하고 있다. 이 중에서 컴파일 에러와 실행 에러가 발생하는 곳을 찾아서 적고 이유를 자세히 설명하라. (10점)

e1는 compile error; cannot create A abstract class object
e4는 compile error; cannot convert from A to B
e5는 run-time error; ClassCastException: D cannot be cast to B
e6는 compile error; cannot convert from C to D
e8는 compile error; m cannot be resolved to a variable

3. 위의 main() 메소드에서 elements마다 methods1 ~ method3 메소드를 실행한 결과를 적고 이유를 자세히 설명하라 (dynamic binding에 주의). (10점)

```
c // this는 C => this.toString()은 c
c1 0 // C.method1(0)
c2 10 20 // 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()
a // e10의 this는 A => this.toString()은 a
e10 3 // e10.method1(3)
e10 20 // e10.method2() super.m = 20
a // e10.method3() 호출하는데 => this는 A => A.toString()
```

학과 _____ 학번 _____ 이름 _____

4. 추상 클래스란? 추상 메소드란? 인터페이스란? 코드에서 예시를 찾아 자세히 설명하라. 언제 추상 클래스를 사용하고 언제 인터페이스를 사용하는가? (10점)

Abstract Class (추상 클래스)는 반드시 하나 이상의 abstract method(추상메소드)를 가지며, 객체를 생성할 수 없다. 추상 클래스를 상속받은 하위클래스는 반드시 추상 메소드를 재정의(method override) 해야 한다. E.g.) A는 추상클래스.

Abstract Method (추상 메소드)는 메소드 선언만 있고, 구현 내용이 없는 메소드이다. E.g.) void method1(int)와 void method2()는 추상메소드

Interface (인터페이스)는 클래스에서 구현해야 하는 동작을 지정하는 사용하는 추상형. 자바에서 인터페이스를 사용함으로써 다중 상속이 가능하다. E.g.) interface I, interface J

추상 클래스(일반변수+일반메소드+추상메소드 형태)는 공통 부분(abstract)을 상속해서 구현하게 함. 인터페이스(상수+추상메소드 형태)는 기능(abstract) 구현(implements)을 강제함으로써, 구현한 객체들에서 동일한 동작을 보장할 수 있음.

5. 위의 main() 메소드에서 a, b의 실행 결과를 적어라. 그리고, intList를 lambda와 stream을 이용하여, c="53472"로 변환하는 구문을 작성하라. (10점)

a = [25, 16, 49]
b = 165 (=1+9+25+49+81)
c = intList.stream().map(I -> "" + i).reduce("", (i, j) -> i + j); // c=53472

6. 클래스 E를 implements Runnable을 사용하여 Thread 생성하여 실행하는 코드를 작성하라. (즉, 클래스 E와 main() 메소드 내부를 수정) (5점)

```
class E implements Runable {
// 내부구현 동일함
}
public static void main(String[] args) {
        new Thread(new E("test1.csv")).start();
        new Thread(new E("test2.csv")).start();
        new Thread(new E("test3.csv")).start();
}
```

학과 _____ 학번 _____ 이름 _____

7. 위의 main() 메소드에서 Thread가 3개 생성되었을 때, 실행 순서가 보장되도록(즉, t1, t2, t3가 순차적으로 실행) 코드를 수정하라. (Thread의 join() 메소드 사용) (10점)

```
Thread t1 = new Thread(new E("test1.csv"));
Thread t2 = new Thread(new E("test2.csv"));
Thread t3 = new Thread(new E("test3.csv"));
t1.start();
t1.join();
t2.start();
t2.join();
t3.start();
t3.join();
```

8. Process와 Thread의 차이점에 대하여 자세히 설명하라. (5점)

프로세스(process)란 운영체제로부터 process를 할당받고, 운영되기 위해 필요한 주소공간, 파일, 메모리 등 자원을 할당받는다. 프로세스란 보통 program/application과 동일한 개념이다.

스레드(thread)란 한 프로세스 내에서 동작되는 여러 실행의 흐름으로, 같은 process 내의 주소공간이나 자원들을 thread끼리 공유하면서 실행된다.

9. Mutex, Semaphore, Monitor가 무엇인지 자세히 설명하라. (10점)

Mutex Semaphore(also known as Mutex) controls only one thread at a time executing on the shared resource by lock & unlock. 뮤텍스란 멀티스레딩 프로그램에서 공유자원에 대한 동기화(synchronization)를 위해 한번에 하나의 스레드가 접근 할 수 있도록 lock/unlock 시켜준다.

Counting Semaphore(also known as Semaphore) controls the number of threads executing on the shared resources by acquire & release. 세마포어란 멀티스레딩 프로그램에서 공유자원에 대한 동기화(synchronization)를 위해 지정한 개수의 스레드까지 동시에 실행이 가능하게 하며 acquire/release를 통해 counting 개수가 감소/증가된다.

Monitor controls only one thread at a time, and can execute in the monitor (shared object) by wait & notify/notifyAll. 모니터란 멀티스레딩 프로그램에서 공유자원에 대한 동기화(synchronization)를 위해 사용하는 것으로 뮤텍스와 동일한 개념이나, 공유자원을 모니터 객체로 구현하여 한번에 하나씩 실행하도록 해준다.

학과 _____ 학번 _____ 이름 _____

10. Producer-Consumer Problem이 무엇인지? synchronized 키워드가 무엇인지? wait(), notifyAll() 메소드가 무엇인지? 자세히 설명하라. (10점)

```
Producer() {
   synchronized put(item) {
     while(count >= buffer.length) { wait(); }
     count++
     <<< critical section: Put item into shared buffer >>>
     notifyAll(); // signal
   }
}

Consumer() {
   synchronized take () {
     while(count <=0) { wait(); }
     count- –
     <<< critical section: Take item from shared buffer >>>
     notifyAll(); // signal
     return item;
   }
}
```

Producer-Consumer Problem 은 여러 개의 스레드를 어떻게 동기화할 것인가에 관한 문제이다. 한정된 버퍼(bounded-buffer)에 어러명의 Producer 와 Consumer 가 접근해서, Producer 는 데이터를 생성해서 넣고, Consumer 는 데이터를 버퍼에서 꺼내어 쓴다.

synchronized 키워드를 사용하여 해당 코드 블록이 여러 개의 스레드에 안전하게 한번에 하나씩 동작이 되도록 함

wait() 메소드는 Producer 에서는 버퍼가 가득 찾을 경우 (Consumer 에서는 버퍼가 비었을 경우), 더 이상 데이터를 처리할 수 없으므로 스레드를 일단 대기 상태로 만듦

notifyAll() 메소드는 스레드가 critical section 을 다 사용한 후, 다른 대기하고 있는 스레드를 일깨움.

11. TCP vs UDP에 대하여 자세히 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.)

TCP 서버 – ServerSocket 클래스와 Socket 클래스
TCP 클라이언트 – Socket 클래스
getInputStream(), getOutputStream()를 사용한 데이터 전송

UDP 서버 – DatagramSocket 클래스
UDP 클라이언트 – DatagramSocket 클래스
receive(), send()를 사용한 데이터 전송

-끝-