

프로그래밍언어론

2014년 시행 5급(기술) 공채 제2차시험

응시번호 :

성명 :

제 1 문. 아래 문맥자유문법(CFG: context-free grammar)에 대하여 다음 물음에 답하시오.
(단, 시작기호는 A이다) (총 14점)

$$A \rightarrow V \mid V, A$$

$$V \rightarrow 0 \mid 1$$

- 1) 위 문법이 생성하는 언어가 무엇인지 설명하시오. (2점)
- 2) 순환하강(recursive-descent) 구문분석을 적용할 수 있도록 위 문법을 수정하시오.
(단, 필요하다면 새로운 비단말기호(nonterminal)를 추가할 수 있다) (6점)
- 3) 위 문법이 생성하는 문장(sentence)에 대해서 0의 개수와 1의 개수를 계산하는 속성문법(attribute grammar)을 작성하시오. (6점)

제 2 문. 아래 C++ 프로그램에 대하여 다음 물음에 답하시오.

(총 12점)

```
class A {
public :
    A() {
        a = new int [3];
        for (int i=0; i<3; i++)
            a[i] = i;
    }
    ~A() {
        delete[] a;
    }
private :
    int *a;
};

void init(A& x) {
    A y;
    x = y;
}

int main() {
    A p;
    ...
    init(p);           // a)
    ...
}
```

- 1) 위 프로그램에서 a)의 init(p) 실행 후 발생하는 문제점을 설명하시오. (6점)
- 2) 1)에서 설명한 문제점을 해결하도록 클래스 A에 연산자 중복(operator overloading) 함수를 추가하시오. (6점)

제 3 문. 아래 Java 프로그램에 대하여 다음 물음에 답하시오.

(총 12점)

```
abstract class Vehicle {
    abstract public int speed();
    abstract public void ride();
    public void message() {
        ride();
    }
}

class Bike extends Vehicle {
    public int SAFE = 30;
    public int safeSpeed() {
        return SAFE;
    }

    public int speed() {
        return safeSpeed();           // a)
    }

    public void ride() {
        System.out.println("Bike: It's fun");
    }
}

class Car extends Vehicle {
    String kind;
    Car(String kind) {
        this.kind = kind;
    }
}
```

```
private int currentSpeed(int l) {
    return (int)(Math.random()*l);    // b)
}

public final int speed() {
    return currentSpeed(300);         // c)
}

public void ride() {
    System.out.println(kind+": It's boring");
}

}

public class Main {
    static Vehicle v = new Bike();
    public static void main(String[] args) {
        Car c1 = new Car("Pony");
        System.out.println(v.speed());    // d)
        System.out.println(c1.speed());    // e)
        v.message();                     // f)

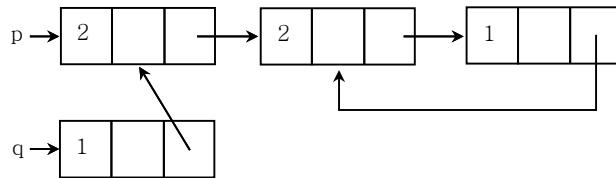
        Car c2 = new Car("Sports Car");

        test(c1);
        test(c2);
    }

    private static void test(Vehicle v) {
        v.ride();                         // g)
    }
}
```

- 1) 위 프로그램의 밑줄 친 a) ~ g)의 메소드 호출 중에서 정적 메소드 바인딩(static method binding)이 적용되는 것을 모두 선택하고, 각각에 대하여 이유를 설명하시오. (6점)
- 2) Java 언어에서 동적 메소드 바인딩(dynamic method binding)을 구현하는 방법을 위 프로그램의 경우를 활용하여 도식적으로 설명하시오. (6점)

제 4 문. 힙(heap) 쓰레기 수거(garbage collection) 기법의 하나인 참조횟수 계수(reference counting) 기법과 관련하여, 아래 그림에 보인 힙 상태를 가정하고 다음 물음에 답하시오. (각 노드의 첫번째 필드는 참조횟수, 세번째 필드는 노드를 가리키는 주소이다. p와 q는 노드를 가리키는 포인터이며 자유리스트(free list)는 생략되었다) (총 12점)



- 1) 위 힙 메모리 상태를 활용하여 참조횟수 계수 기법에서 생길 수 있는 문제점을 설명하시오. (6점)
- 2) 또 다른 힙 쓰레기 수거 기법인 표시 후 청소(mark and sweep) 기법에서는 1)에서 설명한 문제가 어떻게 해결되는지 설명하시오. (6점)

안전행정부 시험출제과장