

2. 학생의 이름, 학번, 국, 영, 수 점수를 담을 수 있는 클래스를 정의하라. 데이터 타입은 이름은 **String**, 나머지는 **int**이며, 클래스명은 **Student** 이다.
3. 클래스의 객체 하나를 생성하라.
4. 클래스에 총점을 계산하여 출력하는 메소드를 추가하라.
5. 클래스 타입의 크기가 3인 배열을 생성.

6. 다음과 같은 멤버변수를 갖는 SutdaCard 클래스를 정의하시오.

타입	변수명	설명
int	num	카드의 숫자.(1~ 10 사이의 정수)
boolean	isKwang	광(光)이면 true, 아니면 false

7. 다음과 같은 멤버변수를 갖는
NewStudent클래스를 정의하시오.

타입	변수명	설명
String	name	학생이름
int	ban	반
int	no	번호
int	music	음악점수
int	math	수학점수

8. 앞에서 정의한 **NewStudent** 클래스의 객체를 하나 만들어서 5개의 멤버변수에 값을 할당하는 **main()** 함수를 만들어라.
(**NewStudentMain** 클래스)

9. 객체를 3개 만들어라. (객체배열생성) 객체에 값도 할당하라

10. 위에서 작성한 **NewStudent** 클래스에 다음과 같이 정의된 두개의 메소드 **getTotal()** 과 **getAverage()**를 추가하시오. **main()**에서 호출하시오.

메소드명 : **getTotal**

기능 : 음악(music), 수학(math)의 점수를 모두 더해서 반환한다.

반환타입 : **int**

매개변수 : 없음.

메소드명 : **getAverage**

기능 : 총점을 과목 수로 나눈 평균을 구한다.

반환타입 : **float**

매개변수 : 없음.

11. 매개변수로 정수형 배열을 가지고, 배열에 저장된 값의 최대값과 최소값을 return하는 메소드 `minMax()`를 정의하라. Return해야 할 값이 두 개 이므로 새로운 클래스 (`MyData`)로 정의하여 이용하도록 한다.

복습문제

- ()는 객체를 만들기 위한 틀이며, 객체를 만들려면 반드시 객체의 다양한 특성을 표현해서 만들어야 함
- 객체의 특성을 표현하는 정적인 성질을 ()이라 하며, 프로그래밍에서는 멤버변수로 표현.
- 객체의 일을 처리하거나, 객체들간의 서로 영향을 주고 받는 동적인 일을 처리하는 프로그래밍에서는 ()로 표현
- 클래스로부터 객체를 만들면 각각의 객체는 독립적이다
- 프로그래밍 관점에서 새로운 클래스를 만드는 것은 새로운 ()을 만드는 것이다.

- 아래와 같은 경우, 파일명은?
- 컴파일한 후 결과물인 *.class는 어떤것인가?

- MinMax클래스의 멤버

변수는 무엇이 있는가?

- 같은 패키지 안에

MinMax.java 파일을

추가하려고 한다.
가능한가?

```
package haft;

public class Test {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        int result = 1;
        for (int i = 0; i < 6; i++)
        {
            result = result * (i+1); //1 //2 2 = 1*2//3 1*2*3 //4
            System.out.println(result);
        }

    }

}

class MinMax
{
    int min;
    int max;

}
```


클래스를 정의하라

Calcurator 클래스를 정의하라

멤버변수로 int형의 x, y값을 가지고 있다

멤버 메소드 add메소드는 위의 x,y 값을 더하여 리턴해주는 동작을 정의하고

멤버 메소드 sub메소드는 위의 x,y값을 빼기하여 결과를 돌려주는 동작을 정의하시오.

멤버 메소드 mul메소드는 위의 x,y값을 곱하기하여 결과를 반환하는 동작을 정의하시오

멤버 메소드 divide메소드는 위의 x,y값을 나누기하여 결과를 return하는 동작을 정의하시오

위의 Calcurator의 객체를 CalcuratorMain클래스의 main()에서 생성하시오.

위의 객체로 add메소드, sub메소드, mul메소드, divide메소드 호출하여보시오

main()에서 아래와 같은 동작이 정의해졌을 때,
틀린 부분은 어디인가? 정상동작 할 수 있도록 수정하시오.

```
Calcurator[3] cal = new Calcurator[3];  
cal[0].x=5;  
cal[1].y=9;
```

- 다음과 같은 조건을 만족하는 클래스 CreditCard를 정의하시오.
- long의 데이터 타입의 number멤버변수
- String 데이터 타입의 owner 멤버변수
- 위 클래스의 객체를 main()에서 만드시오.
- 객체에 적당한 값을 할당해 주시오
(ex. number : 3456765487655647L
owner : "유재석")
- 객체의 값을 출력하시오.

- CreditCard클래스에
- 신용카드를 사용하는 메소드 use()를 추가하시오
 - 반환타입: void, 매개변수는 int형 1개
- 동작은 다음과 같다.

```
{  
    balance = balance + 매개변수;  
}
```
- 위와 같이 정의하면, 멤버변수가 1개 추가 되어야한다.
멤버변수가 추가된 CreditCard클래스 전체를 명시하라
(use()멤버메소드도 추가되어 있어야함.)

- CreditCard클
추가하라.

```
void payBill(int amount)
{
    balance = balance - amount;
    addPoint(amount);
}

void addPoint(int amount)
{
    point = point + amount/1000;
}
```

- CreditCard클래스의 객체배열을 main()에
서 선언, 생성하라(5개 배열)
- 각 배열에 번호와 소유자를 임의로 할당해
주시오.

- 결과를 유추 후 실행해보시오

```
public class ShapeMain {  
    public static void main(String[] args) {  
        // TODO Auto-generated method stub  
  
        Rectangle rect = new Rectangle();  
        Point p = new Point();  
        p.x = 5; p.y = 7;  
        rect.setPoint(p);  
        int a = rect.area(rect.getPoint());  
        System.out.println("면적은 "+a);  
  
        Point p2 = new Point();  
        p2.x = 10;  
        p2.y = 19;  
        int b = rect.line(rect.getPoint(), p2);  
        System.out.println("둘레는 " + b);  
    }  
}  
class Rectangle  
{  
    Point p;  
    void setPoint(Point _p)  
    {  
        p = _p;  
    }  
    Point getPoint()  
    {  
        return p;  
    }  
    int area( Point p )  
    {  
        return p.x * p.y;  
    }  
    int line( Point p1, Point p2)//둘레길이.  
    {  
        return Math.abs(p1.x - p2.x)*2 + Math.abs(p1.y - p2.y)*2;  
    }  
}  
class Point  
{  
    int x;  
    int y;  
    void print()  
    {  
        System.out.println("점은 x= " + " y= " + y);  
    }  
}
```

- 로또 번호 생성기를 만들어서 가장 많이 불린 숫자를 찾아주는 프로그램을 작성하시오. (단, Lotto클래스를 자유롭게 만들어서 활용하시오)