



Professor: Edwar Saliba Júnior

1) Analise o código-fonte a seguir e apresente o que será impresso na tela do computador no local adequado.

====[Classe CalculoSimples]=====

```
package poo_exerciciodedepuracao;

public class CalculoSimples {
    private int x;
    protected int y;
    protected int z;

    public CalculoSimples(int x, int y, int z) {
        this.x = x - 1500;
        this.y = y;
        this.z = z;
    }

    public int calcSimples01(){
        return x - 2 * (y + z);
    }

    public int calcSimples02(){
        return x - 2 * y + z;
    }

    public int calcSimples03(){
        return (x - 2) * y + z;
    }

    public int getX() {
        return x;
    }
}
```

====[Classe CalculoComplexo]=====

```
package poo_exerciciodedepuracao;

import java.util.ArrayList;
import java.util.Stack;

public class CalculoComplexo extends CalculoSimples {

    private int num1;
    private int num2;
    private int total;
    private int valor;
    private int val1;
    private int val2;
    private int val3;
    private int fator;
```

```

public CalculoComplexo(int v1, int v2, int v3, int fator){
    super(v3, v1, v2);

    num1 = 0;
    num2 = 0;
    total = 0;
    this.fator = fator;
    val1 = v1;
    val2 = v2;
    val3 = v3;

    valor = val3 - (val2 * val1) - super.calcSimples02() - 1780 - fator;
}

public int calc01(int v1, int v2){
    int res, aux = valor > 0 ? valor - 2015 : valor + 2015;
    res = (v1 + y) - aux - fator;
    return(res - z);
}

public int calc02(){
    ArrayList<Integer> numeros = new ArrayList<>();
    numeros.add(super.calcSimples01());
    total = 10;
    numeros.add(total - val2);
    numeros.add(val1);
    for(int i = 1; i <= 2; i++){
        num1++;
        z--;
        num2 -= 3;
        if(numeros.get(1) % 2 == 0)
            numeros.add(num1 + z - num2 - fator);
        else
            numeros.add(z - num1 - num2 + fator);

        numeros.remove(0);
    }
    numeros.remove(0);
    numeros.remove(1);

    for(Integer val : numeros)
        total += val;

    return(total);
}

public void calc03(){
    int n2 = super.calcSimples03();
    if(num2 < num1)
        total = 1 - val1 + n2 - fator;
    else
        total = num1 + n2 - super.getX() + fator;
}

public int calc04() {
    return (valor % 2 == 1) ? valor - fator : 4 * val1 - val2 - fator;
}

public void calc05() {
    Stack<Integer> pilha = new Stack<>();
    pilha.push(val1);
}

```

```

    pilha.push(valor + z);
    pilha.push(fator - 3);

    int val = pilha.peek();

    pilha.push(num1 - num2);

    for(int i = 0; i < pilha.size(); i++) {
        val += pilha.pop();
    }

    total = val;
}

public int getTotal(){
    return(total);
}

public void setNum1(int v1){
    num1 = v1 + 3;
}

public void setNum2(int v1){
    num2 = v1;
}
}

```

====[Classe P00_ExercicioDeDepuracao]=====

```

package poo_exerciciodedepuracao;

public class P00_ExercicioDeDepuracao {

    public static void main(String[] args) {
        CalculoComplexo ca;
        int op, FATOR;

        FATOR = 150;

        int valor1 = 5,
            valor2 = 4,
            resultado = 0;

        int i = 1, val1, val2, val3;

        val1 = 17;
        val2 = 9;
        val3 = 1999;

        ca = new CalculoComplexo(val1, val2, val3, FATOR);

        while(i < 3){
            resultado += ca.calc01(valor1, i);
            i++;
        }

        System.out.println("\n\nResultado - calc01: " + resultado);

        ca.setNum1(valor1);
        ca.setNum2(valor2);
        resultado = ca.calc02();
    }
}

```

```

System.out.println("\n\nResultado - calc02: " + resultado);

ca.setNum1(valor1);
ca.setNum2(valor2);
ca.calc03();
resultado = ca.getTotal();

System.out.println("\n\nResultado - calc03: " + resultado);

ca.setNum1(resultado);
ca.setNum2(1);
resultado = ca.calc04();

System.out.println("\n\nResultado - calc04: " + resultado);

ca.setNum1(2);
ca.setNum2(1);
ca.calc05();
resultado = ca.getTotal();

System.out.println("\n\nResultado - calc05: " + resultado + "\n\n");
}
}

```

Local para Resposta do Exercício	
Resultado - calc01:	
Resultado - calc02:	
Resultado - calc03:	
Resultado - calc04:	
Resultado - calc05:	