1. Clase es el modelo de un objeto de la realidad, el objeto es una instancia de una clase.
2. Constructor es el método que se ejecuta al momento de instanciar una clase, el método se utiliza ya cuando se tiene un objeto.
3. La variable local existe solamente dentro de un método o dentro de un contexto, el parámetro es las variables que recibe un método para su manipulación.
4. Si ya que se necesita saber cuáles son sus entradas y salidas, pero no precisa saber cuales son los detalles de su comportamiento
5. Para copiar variables solo hace falta igualar las variables/objetos teniendo en cuenta que son del mismo tipo, la diferencia en que los datos primitivos, al hacer la igualación se creó un nuevo espacio de memoria en donde se introduce el dato a copiar y ambos se pueden manipular independientemente, en cambio con los objetos, si solo se iguala, se estaría teniendo la referencia de un solo objeto en dos variables, con lo cual si se modifica el aspecto de una, se modificara el de la otra ya que se estaría modificando el mismo objeto, para realizar una copia, se debe instanciar un nuevo objeto con la misma información del objeto que se desea copiar.
6. [Modificador] [Tipo de dato] <Nombre del atributo>
7. Por ejemplo se quiere tener un conjunto de perros (Clase perro) en el cual pueden nacer o pueden unirse al grupo, para los que nacen se puede tener un constructor que solo reciba el nombre del perro, ya que los demás valores estarán definidos ya que se sabe el inicio, en cambio si se une un nuevo perro, pues pedirá todos los datos en el constructor.
8. Las clases actrices realizan una clase de trabajo y es una buena practica que terminen en -or -ora, y las utilitarias contienen una gran cantidad de atributos y métodos estáticos.
9. El acoplamiento es básicamente que tanto saben las clases entre ellas, mientras menos sepan las clases de otras, es decir, mientras se tenga un menor acoplamiento se estará haciendo un buen diseño
10. El accesor es el que retorna el valor de un atributo, y el mutador modifica el valor de un atributo de una clase.
11. Ya que en java o en C# todo es enviado por referencia o por si se quiere ver por medio de punteros independiente de ser objetos o números, ya que al momento de enviarlos por parámetros no le enviamos la dirección de memoria, sino, el dato como tal.
12. Es el hecho en que todas las pre-condiciones se cumplan en un método y este promete realizar un trabajo de manera correcta
13. Los métodos de clase no operan sobre las variables de instancia de objetos, mientras que los métodos de instancia tienen acceso a las variables de clase.
14. Al momento de crear una clase utilitaria es muy recomendable realizar una clase estática ya que no se necesitará la instanciación de la clase para hacer uso de los métodos.
15. El método que devuelva la cadena de texto ya que es algo que le compete más a un método que imprimir el contenido, esta tarea puede realizarla la función principal del programa.

PARTE II

Ejercicio 1.

clasePrincipal.java

```
public class P1E1 {
    static Tramo tII = new Tramo(472.01,895.24,0.1,472.00,17.67);
    static Tramo tIII = new Tramo(895.25,2038.10,0.2,895.24,60.00);
    static Tramo tIV = new Tramo(2038.11,0.30,2038.10,288.57);
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Persona me = new Persona("Carlos",2400.00);
        calcular(me);
    }
    public static void calcular(Persona me){
        double salario = me.getSalario();
        Retencion afp = new Retencion(0.0625);
        Retencion isss = new Retencion((me.getSalario()>1000.00)?30.00:0.03);
        salario = salario - (salario * afp.getProcentaje()) ;
        if(me.getSalario() > 1000.00){
            salario = salario - isss.getProcentaje();
        }else{
            salario = salario - (salario * isss.getProcentaje());
        }

        if(salario >= tII.getA() && salario <= tII.getB()){
            salario = salario - ((salario-tII.getExc())*tII.getPer() + tII.getCfija());
        }
        if(salario >= tIII.getA() && salario <= tIII.getB()){
            salario = salario - ((salario-tIII.getExc())*tIII.getPer() + tIII.getCfija());
        }
```

```
        if(salario >= tIV.getA()){
            salario = salario - ((salario-tIV.getExc())*tIV.getPer() + tIV.getCfija());
        }
        me.setSalario(salario);
        System.out.println("Salario neto de: " + me.getNombre() +" - $" +me.getSalario());
    }
}
```

Retencion.java

```
public class Retencion {
    private double porcentaje;

    public Retencion(double procentaje){
        this.porcentaje = procentaje;
    }
    public double getProcentaje(){
        return porcentaje;
    }
}
```

Persona.java

```
private String nombre;
    private double salario;
    public Persona(String nombre, double salario){
        this.nombre = nombre;
        this.salario= salario;
    }

    public String getNombre() {
        return nombre;
    }

    public double getSalario() {
```

```
    return salario;
  }

  public void setSalario(double salario) {
    this.salario = salario;
  }
```

Tramo.java

```
public class Tramo {
  private double a;
  private double b;
  private double per;
  private double exc;
  private double cfija;

  public Tramo(double a, double b, double per, double exc, double cfija) {
    this.a = a;
    this.b = b;
    this.per = per;
    this.exc = exc;
    this.cfija = cfija;
  }

  public Tramo(double a, double per, double exc, double cfija) {
    this.a = a;
    this.per = per;
    this.exc = exc;
    this.cfija = cfija;
  }

  public double getA() {
    return a;
  }
```

```
    public double getB() {
        return b;
    }

    public double getPer() {
        return per;
    }

    public double getExc() {
        return exc;
    }

    public double getCfija() {
        return cfija;
    }
}
```

3.

clasePrincipal.java

```
public class P1E3 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Cajero c = new Cajero(13.5,20.00);
        c.calcularVuelto();
    }

}
```

Cajero.java

```
public class Cajero {
    private double total;
    private double entrada;


    public Cajero(double total, double entrada) {
        this.total = total;
        this.entrada = entrada;
    }
    public void calcularVuelto(){
        int dolares=0;
        int coras =0;
        int centavos10=0;
        int centavos5=0;
        int centavos1=0;
        double vuelto = entrada - total;
        if(vuelto < 0){
            System.out.println("El valor ingresado no cubre los gastos totales");
        }else{
            if(vuelto ==0){
                System.out.println("Gracias por la compra tu vuelto: $0.0");
            }else{
                while(vuelto > 0){
                    String aux = vuelto + "";
                    String a[] = aux.split("[,.]");
                    String v = String.join("", a);
                    int vueltoI = Integer.parseInt(v);
                    if(vueltoI%10==0){
                        dolares++;
                        vuelto -= 1.0;
                    }else if(vueltoI%0.25==0){
```

```
                coras++;
                vuelto -= 0.25;
            }else if(vueltoI%0.1==0){
                centavos10++;
                vuelto -= 0.1;
            }else if(vueltoI%0.05==0){
                centavos5++;
                vuelto -= 0.05;
            }else if(vueltoI%0.01==0){
                centavos1++;
                vuelto -= 0.01;
            }
        }
        System.out.println("Dolares: " + dolares);
        System.out.println("Coras: " + coras);
        System.out.println("Centavos de 10: " + centavos10);
        System.out.println("Centavos de 5: " + centavos5);
        System.out.println("Centavos de 1: " + centavos1);

      }
    }
  }
}
```