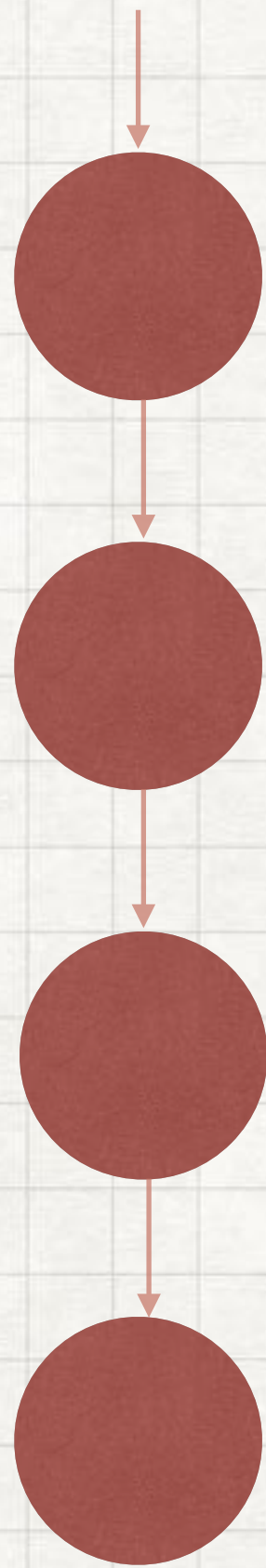
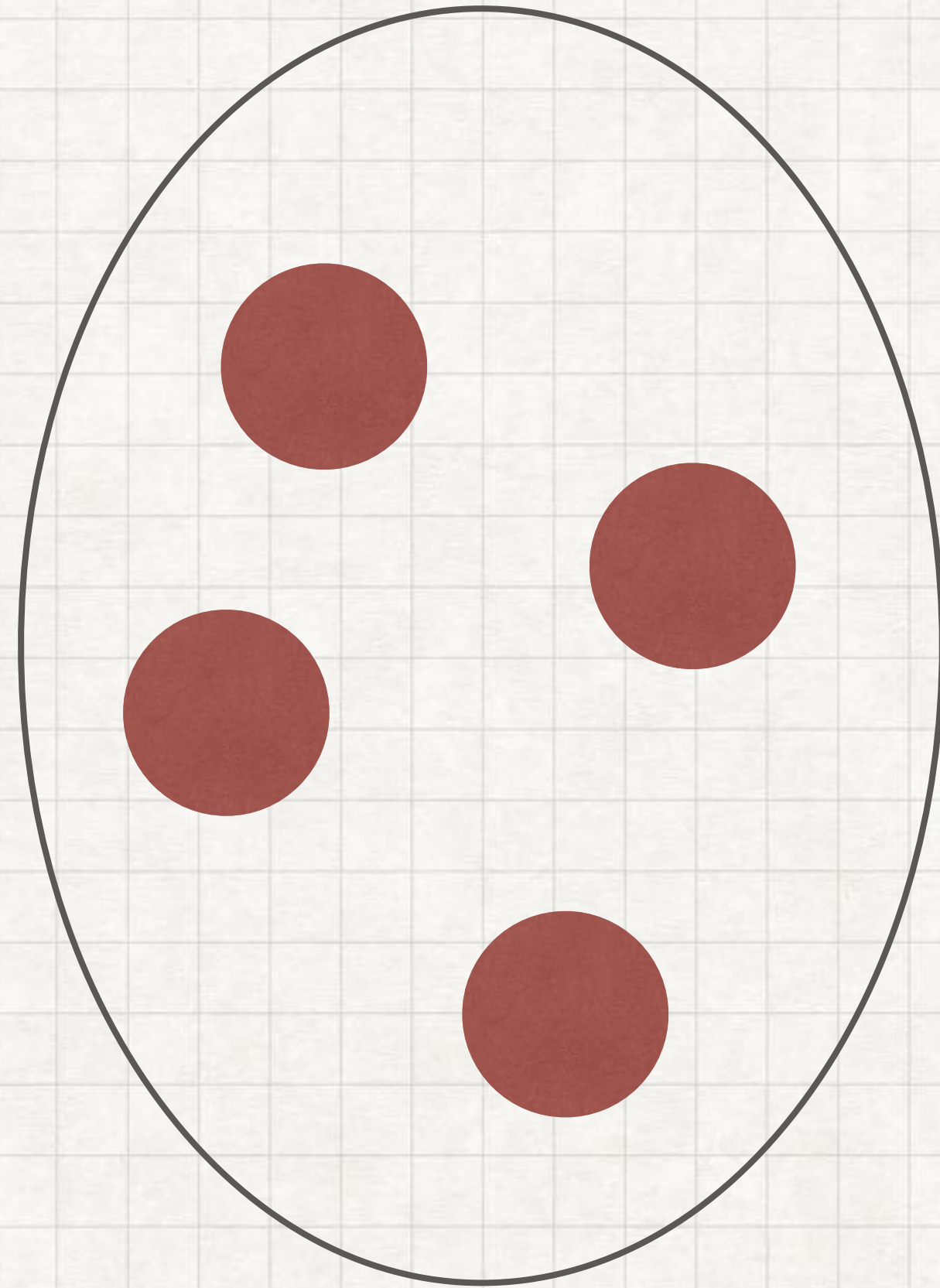


ITERATOR PATTERN

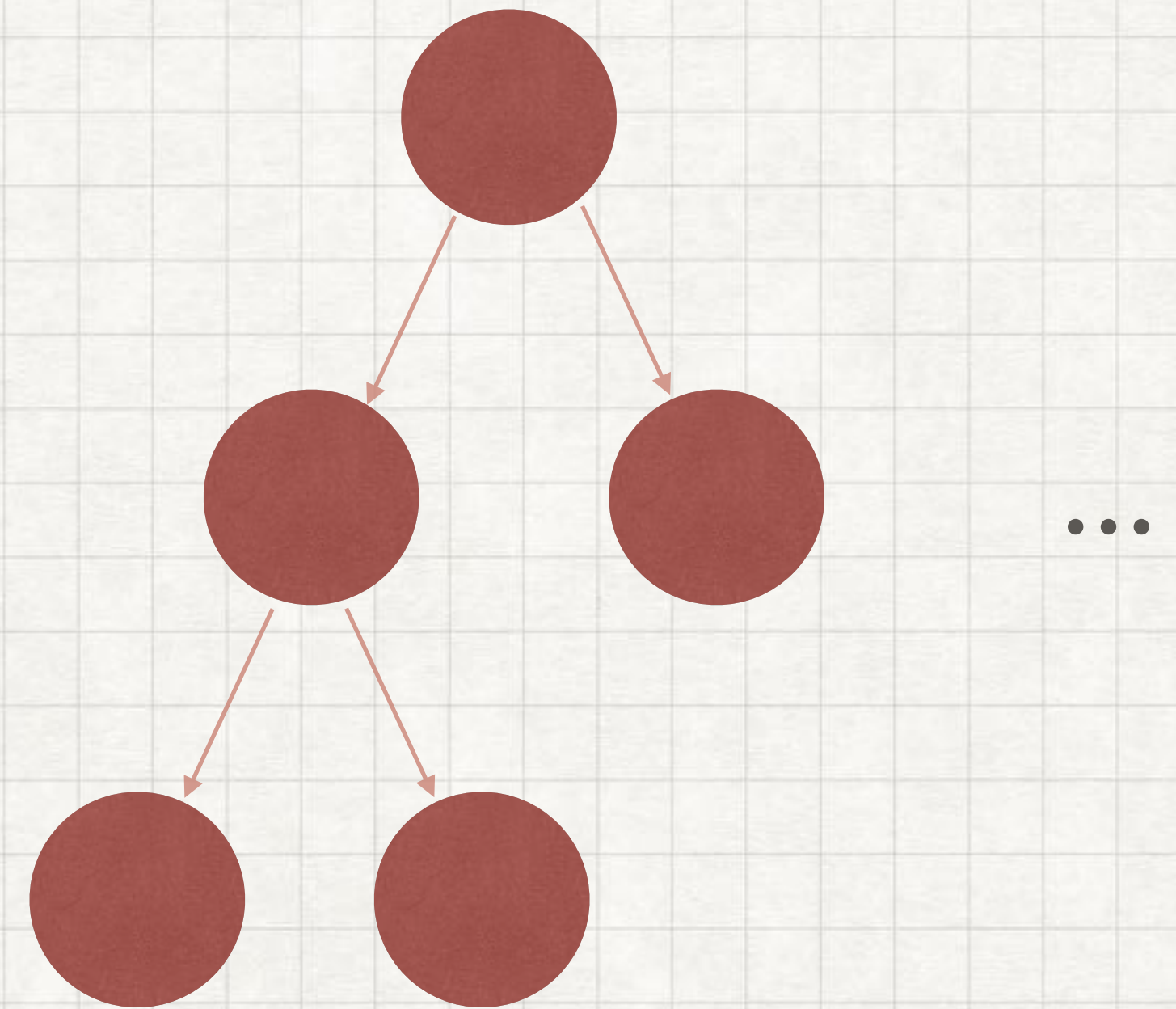
Existen muchas formas de agrupar colecciones de componentes



Listas



Conjuntos



Árboles

La forma de recorrer cada colección es diferente.

Los clientes se pueden encontrar una API diferente para cada tipo de estructura, para esta misma operación

“

Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.

— *GoF*

”

java.util

Interface Iterator<E>

Method Summary

Methods

Modifier and Type	Method and Description
boolean	hasNext () Returns true if the iteration has more elements.
E	next () Returns the next element in the iteration.
void	remove () Removes from the underlying collection the last element returned by this iterator (optional operation).

VISITOR PATTERN

Cliente

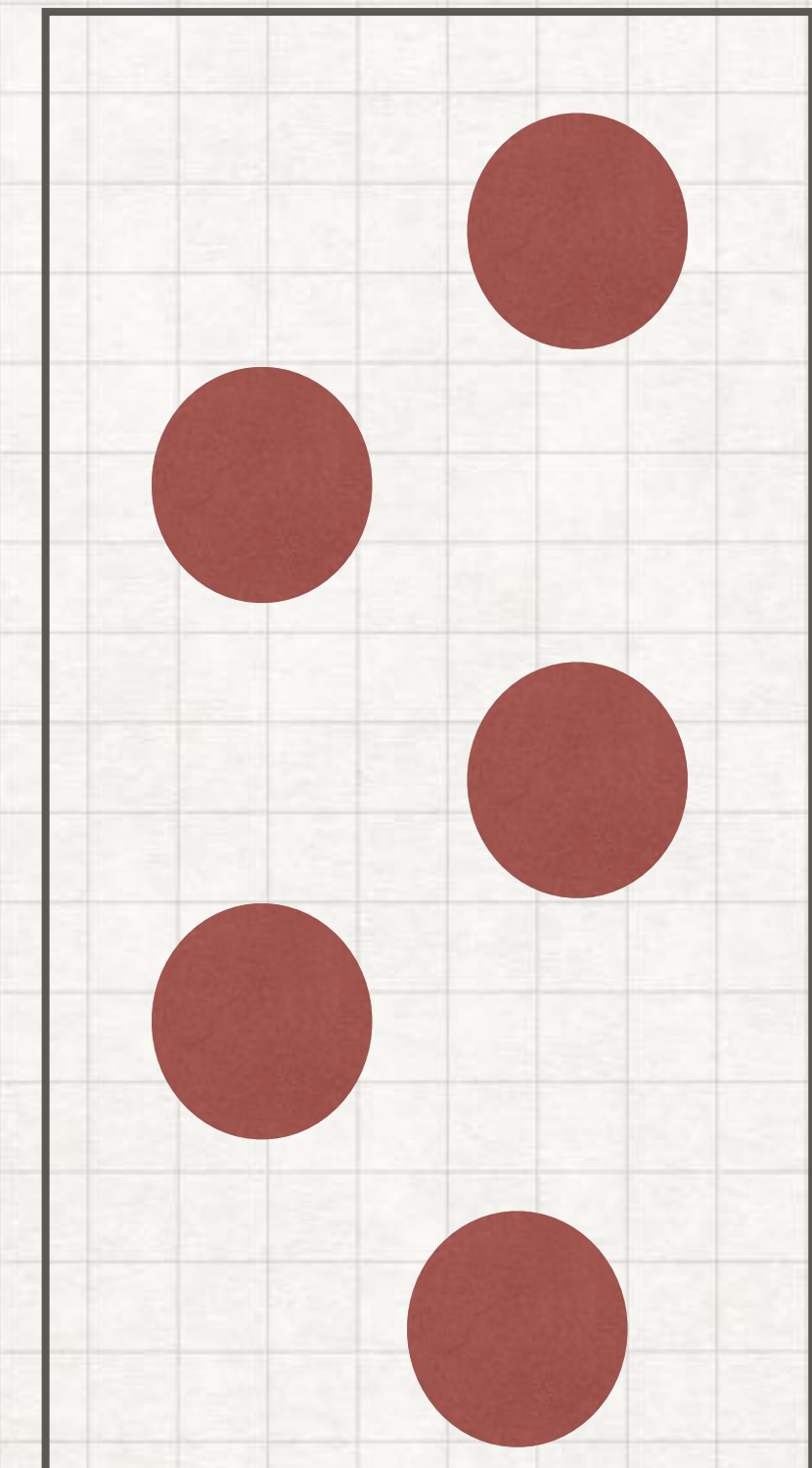
Agrupación

Iterator



Visitor

QUE HACER CON
CADA ELEMENTO



“

Represent an operation to be performed on the elements of an object structure. Visitor lets you define a new operation without changing the classes of the elements on which it operates.


— *GoF*

”


```
public class LibroEntradas {  
    ...  
}
```



```
public class RegistroEntrada {  
  
    private String cliente;  
    private Date fecha;  
    private int importe;  
    private int pagado;  
    private Detalle detalle;  
  
    ...  
}
```



- LibroEntradas tiene muchos registros
- La organización es compleja:
 - por años
 - por meses
 - ...
- Queremos hacer muchas operaciones sobre los libros de entradas:
 - importe total
 - importe medio
 - listar importes que no se han cobrado
 -

```
public class Detalle {  
  
    private List<String> productos;  
  
    ...  
}
```



```
public interface RegistroEntradaVisitor {  
    void visit(RegistroEntrada registro);  
}
```

```
public class LibroEntradas {  
    public void visit (RegistroEntradaVisitor visitor) {  
        // para cada registro: visitor.visit(registro)  
    }  
}
```



```
public class TotalPagadoVisitor implements RegistroEntradaVisitor {  
  
    private int total = 0;  
  
    @Override  
    public void visit(RegistroEntrada registro) {  
        total += registro.getPagado();  
    }  
  
    public int getTotal() {  
        return total;  
    }  
}
```


Cliente

Agrupación

Iterator



Visitor

QUE HACER CON
CADA ELEMENTO

