Question 1

```
                              ┌──────────────┐
                              │ <<interface>>│
┌──────────────┐             │    Layout    │
│   JFrame     │────────────▷├──────────────┤
├──────────────┤         1   │              │
├──────────────┤             ├──────────────┤
└──────────────┘             │ +organize()  │
                              └──────────────┘
                                     △ △ △
                                    ╱  ┊  ╲
                          ╱┈┈┈┈┈┈┈┈┈   ┊    ┈┈┈┈┈┈┈┈┈╲
          ┌──────────────┐ ┌──────────────┐ ┌──────────────┐
          │  FlowLayout  │ │  GridLayout  │ │ BorderLayout │
          ├──────────────┤ ├──────────────┤ ├──────────────┤
          │ +organize()  │ │ +organize()  │ │ +organize()  │
          └──────────────┘ └──────────────┘ └──────────────┘
```

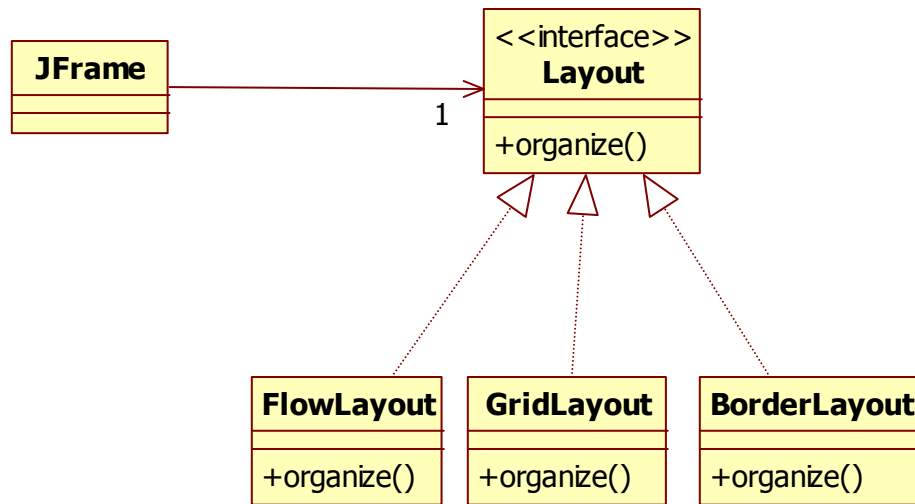**Question 2**

The biggest advantage of solution 2 is that we separate the interest calculation logic from the account logic. This makes it very easy to:

- Add new interest algorithms
- Change the interest algorithm from a certain account without changing the account (open-closed principle)
- Reuse the interest algorithm for something other than an Account or reuse a certain interest algorithm for multiple account types.

These 3 advantages you do not have in solution 1.

Solution 3:

HistoryList

+addCommand()
+undo()
+redo()

<<interface>> ICommand

+execute()
+unexecute()

IncrementCommand

+execute()
+unexecute()

DecrementCommand

+execute()
+unexecute()

ButtonGUI

+increment()
+decrement()
+undo()
+redo()

NumberGUI

+showNumber()

RectangleGUI

+drawRectangle()

CircleGUI

+drawCircle()

CounterService

+increment()
+decreement()
+update()

<<interface>> IObserver

+update()

Subject

+addObserver()
+notify()

Counter

+value

+increment()
+decrement()
+setstate()

CounterDAO

+save()
+update()
+load()

Logger

+log()
+update()

<> TellerState

<>+computeValue()

nextState   1

SingleDigitState

+computeValue()

DoubleDigitState

+computeValue()

TripleDigitState

+computeValue()

*

1

1

1

1

1

1

1