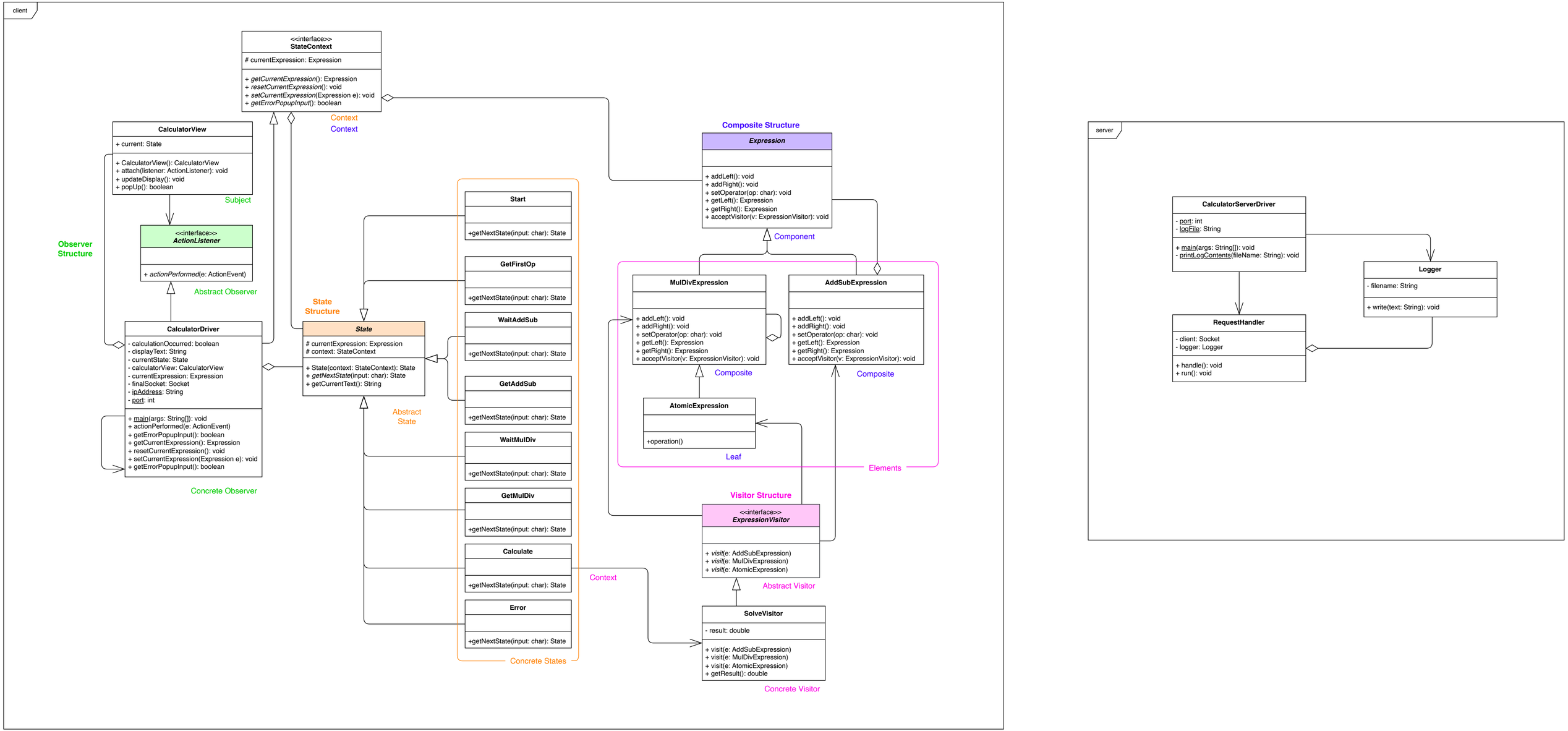
client
<<interface>>
StateContext
currentExpression: Expression
+ getCurrentExpression(): Expression
+ resetCurrentExpression(): void
+ setCurrentExpression(Expression e): void
+ getErrorPopupInput(): boolean
Context
Context
CalculatorView
+ current: State
+ CalculatorView(): CalculatorView
+ attach(listener: ActionListener): void
+ updateDisplay(): void
+ popUp(): boolean
Subject
Observer Structure
<<interface>>
ActionListener
+ actionPerformed(e: ActionEvent)
Abstract Observer
CalculatorDriver
- calculationOccurred: boolean
- displayText: String
- currentState: State
- calculatorView: CalculatorView
- currentExpression: Expression
- finalSocket: Socket
- ipAddress: String
- port: int
+ main(args: String[]): void
+ actionPerformed(e: ActionEvent)
+ getErrorPopupInput(): boolean
+ getCurrentExpression(): Expression
+ resetCurrentExpression(): void
+ setCurrentExpression(Expression e): void
+ getErrorPopupInput(): boolean
Concrete Observer
State Structure
State
currentExpression: Expression
context: StateContext
+ State(context: StateContext): State
+ getNextState(input: char): State
+ getCurrentText(): String
Abstract State
Start
+getNextState(input: char): State
GetFirstOp
+getNextState(input: char): State
WaitAddSub
+getNextState(input: char): State
GetAddSub
+getNextState(input: char): State
WaitMulDiv
+getNextState(input: char): State
GetMulDiv
+getNextState(input: char): State
Calculate
+getNextState(input: char): State
Error
+getNextState(input: char): State
Concrete States
Composite Structure
Expression
+ addLeft(): void
+ addRight(): void
+ setOperator(op: char): void
+ getLeft(): Expression
+ getRight(): Expression
+ acceptVisitor(v: ExpressionVisitor): void
Component
MulDivExpression
+ addLeft(): void
+ addRight(): void
+ setOperator(op: char): void
+ getLeft(): Expression
+ getRight(): Expression
+ acceptVisitor(v: ExpressionVisitor): void
Composite
AddSubExpression
+ addLeft(): void
+ addRight(): void
+ setOperator(op: char): void
+ getLeft(): Expression
+ getRight(): Expression
+ acceptVisitor(v: ExpressionVisitor): void
Composite
AtomicExpression
+operation()
Leaf
Elements
Visitor Structure
<<interface>>
ExpressionVisitor
+ visit(e: AddSubExpression)
+ visit(e: MulDivExpression)
+ visit(e: AtomicExpression)
Abstract Visitor
Context
SolveVisitor
- result: double
+ visit(e: AddSubExpression)
+ visit(e: MulDivExpression)
+ visit(e: AtomicExpression)
+ getResult(): double
Concrete Visitor
server
CalculatorServerDriver
- port: int
- logFile: String
+ main(args: String[]): void
- printLogContents(fileName: String): void
Logger
- filename: String
+ write(text: String): void
RequestHandler
- client: Socket
- logger: Logger
+ handle(): void
+ run(): void