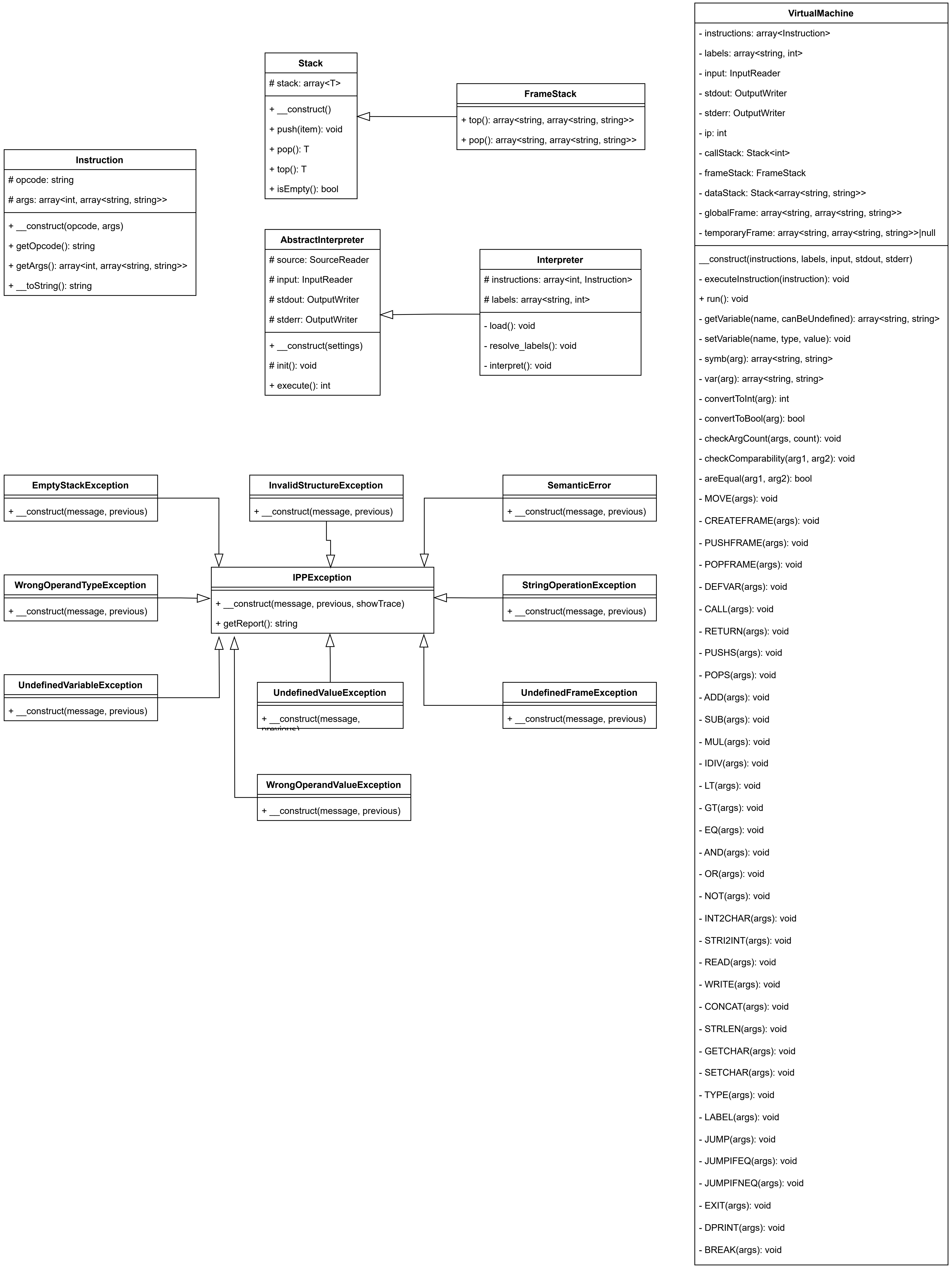

Stack

# stack: array<T>

+ __construct()
+ push(item): void
+ pop(): T
+ top(): T
+ isEmpty(): bool

FrameStack

+ top(): array<string, array<string, string>>
+ pop(): array<string, array<string, string>>

Instruction

# opcode: string
# args: array<int, array<string, string>>

+ __construct(opcode, args)
+ getOpcode(): string
+ getArgs(): array<int, array<string, string>>
+ __toString(): string

AbstractInterpreter

# source: SourceReader
# input: InputReader
# stdout: OutputWriter
# stderr: OutputWriter

+ __construct(settings)
# init(): void
+ execute(): int

Interpreter

# instructions: array<int, Instruction>
# labels: array<string, int>

- load(): void
- resolve_labels(): void
- interpret(): void

EmptyStackException

+ __construct(message, previous)

InvalidStructureException

+ __construct(message, previous)

SemanticError

+ __construct(message, previous)

WrongOperandTypeException

+ __construct(message, previous)

IPPException

+ __construct(message, previous, showTrace)
+ getReport(): string

StringOperationException

+ __construct(message, previous)

UndefinedVariableException

+ __construct(message, previous)

UndefinedValueException

+ __construct(message, previous)

UndefinedFrameException

+ __construct(message, previous)

WrongOperandValueException

+ __construct(message, previous)

VirtualMachine

- instructions: array<Instruction>
- labels: array<string, int>
- input: InputReader
- stdout: OutputWriter
- stderr: OutputWriter
- ip: int
- callStack: Stack<int>
- frameStack: FrameStack
- dataStack: Stack<array<string, string>>
- globalFrame: array<string, array<string, string>>
- temporaryFrame: array<string, array<string, string>>|null

__construct(instructions, labels, input, stdout, stderr)
- executeInstruction(instruction): void
+ run(): void
- getVariable(name, canBeUndefined): array<string, string>
- setVariable(name, type, value): void
- symb(arg): array<string, string>
- var(arg): array<string, string>
- convertToInt(arg): int
- convertToBool(arg): bool
- checkArgCount(args, count): void
- checkComparability(arg1, arg2): void
- areEqual(arg1, arg2): bool
- MOVE(args): void
- CREATEFRAME(args): void
- PUSHFRAME(args): void
- POPFRAME(args): void
- DEFVAR(args): void
- CALL(args): void
- RETURN(args): void
- PUSHS(args): void
- POPS(args): void
- ADD(args): void
- SUB(args): void
- MUL(args): void
- IDIV(args): void
- LT(args): void
- GT(args): void
- EQ(args): void
- AND(args): void
- OR(args): void
- NOT(args): void
- INT2CHAR(args): void
- STRI2INT(args): void
- READ(args): void
- WRITE(args): void
- CONCAT(args): void
- STRLEN(args): void
- GETCHAR(args): void
- SETCHAR(args): void
- TYPE(args): void
- LABEL(args): void
- JUMP(args): void
- JUMPIFEQ(args): void
- JUMPIFNEQ(args): void
- EXIT(args): void
- DPRINT(args): void
- BREAK(args): void