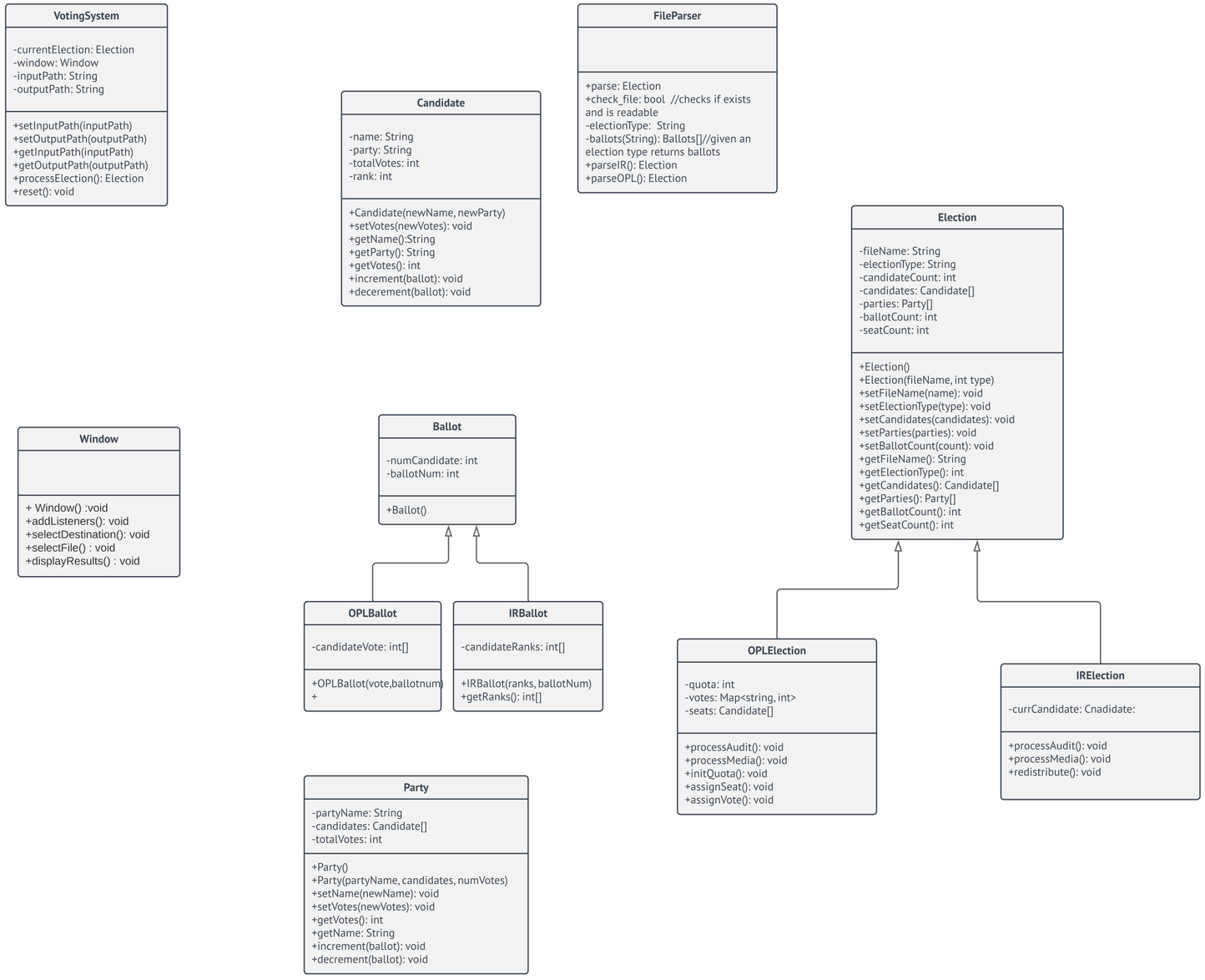
VotingSystem
-currentElection: Election
-window: Window
-inputPath: String
-outputPath: String
+setInputPath(inputPath)
+setOutputPath(outputPath)
+getInputPath(inputPath)
+getOutputPath(outputPath)
+processElection(): Election
+reset(): void
FileParser
+parse: Election
+check_file: bool //checks if exists and is readable
-electionType: String
-ballots(String): Ballots[]//given an election type returns ballots
+parseIR(): Election
+parseOPL(): Election
Candidate
-name: String
-party: String
-totalVotes: int
-rank: int
+Candidate(newName, newParty)
+setVotes(newVotes): void
+getName():String
+getParty(): String
+getVotes(): int
+increment(ballot): void
+decerement(ballot): void
Election
-fileName: String
-electionType: String
-candidateCount: int
-candidates: Candidate[]
-parties: Party[]
-ballotCount: int
-seatCount: int
+Election()
+Election(fileName, int type)
+setFileName(name): void
+setElectionType(type): void
+setCandidates(candidates): void
+setParties(parties): void
+setBallotCount(count): void
+getFileName(): String
+getElectionType(): int
+getCandidates(): Candidate[]
+getParties(): Party[]
+getBallotCount(): int
+getSeatCount(): int
Window
+ Window() :void
+addListeners(): void
+selectDestination(): void
+selectFile() : void
+displayResults() : void
Ballot
-numCandidate: int
-ballotNum: int
+Ballot()
OPLBallot
-candidateVote: int[]
+OPLBallot(vote,ballotnum)
+
IRBallot
-candidateRanks: int[]
+IRBallot(ranks, ballotNum)
+getRanks(): int[]
OPLElection
-quota: int
-votes: Map<string, int>
-seats: Candidate[]
+processAudit(): void
+processMedia(): void
+initQuota(): void
+assignSeat(): void
+assignVote(): void
IRElection
-currCandidate: Cnadidate:
+processAudit(): void
+processMedia(): void
+redistribute(): void
Party
-partyName: String
-candidates: Candidate[]
-totalVotes: int
+Party()
+Party(partyName, candidates, numVotes)
+setName(newName): void
+setVotes(newVotes): void
+getVotes(): int
+getName: String
+increment(ballot): void
+decrement(ballot): void