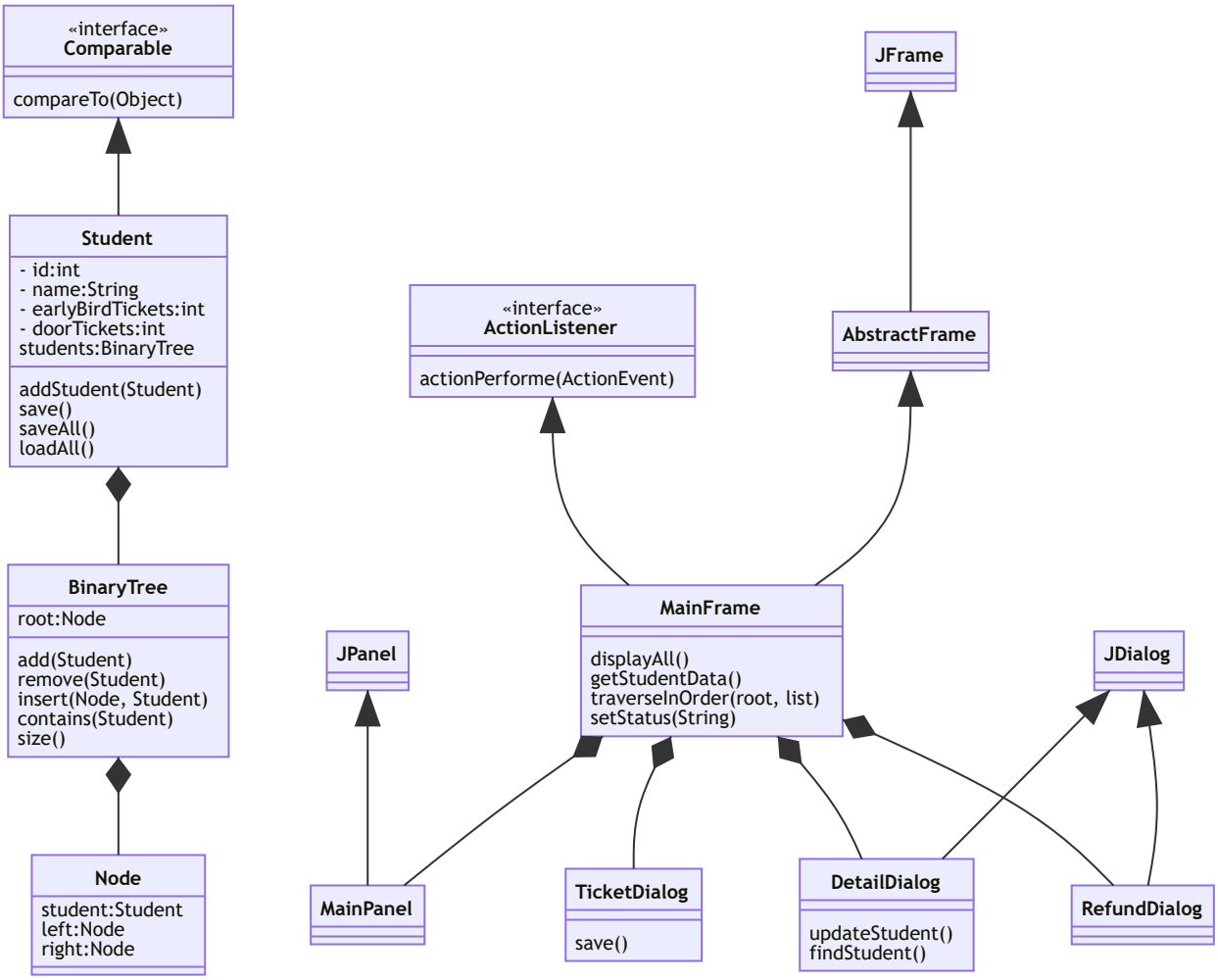
# School Dance Ticket Software

- Students pay for tickets (+ guest)
- Early bird price
- ❔ Cross reference against student number list

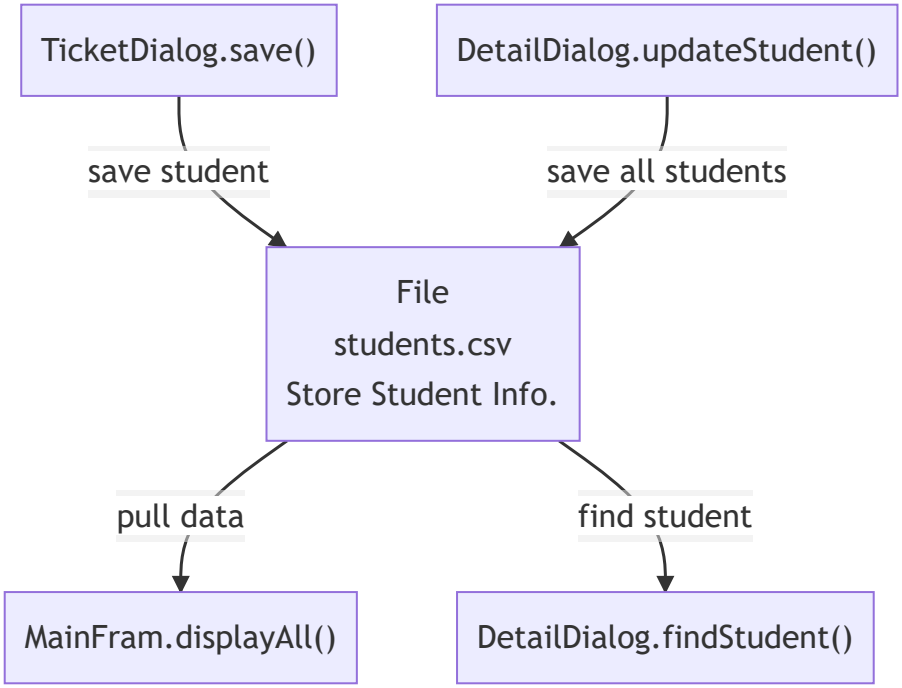
## Completion Check

❔ Methods
❔ Loops
❔ Decision structures
✔ UML - class, attributes, method... 👇See below
❔ Original ideas and design complexity
✔ Data Structure - applicable, Generic [See Node.java and BinaryTree.java]
✔ Functionality - Ticket(early bird, door price, refund), Find(one.update, all), Help
✔ Use of Tree data structure - BinaryTree, Node
✔ Naming conventions (Camel style, upercase for class, lower case for fields and methods)
🔨 Comments/Docs
✔🔨 Code readability (fields and methods naming, function single responsibility)
✔ User Interface (MainFram, MainPanel, DetailDialog, RefundDialog, ...)
❔ UML Layout/presentation
✔ File I/O (see Student.save(), Student.saveAll(), Student.loadAll())
✔❔ Data Structure (ArrayList, LinkedList, Queue)
✔❔ Basic operations (implemented size, contains, add, insert, find, but only used add, find)
✔ Advanced operation - remove() (BinaryTree.remove())
✔ Use of a Tree data structure (BinaryTree.java)

## Class Diagram




# Data store file


TicketDialog.save()
DetailDialog.updateStudent()
save student
save all students
File
students.csv
Store Student Info.
pull data
find student
MainFram.displayAll()
DetailDialog.findStudent()