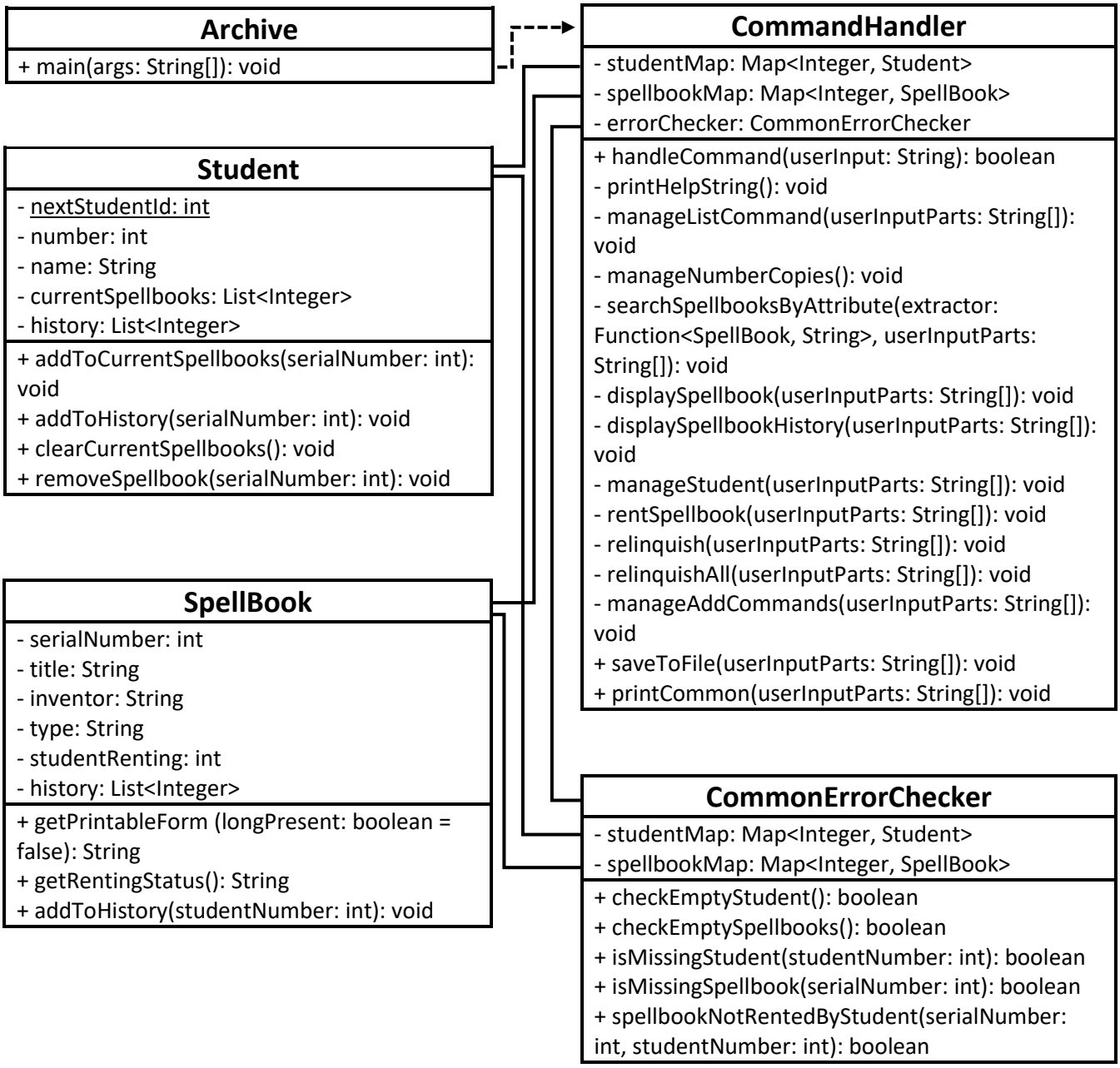
UML Diagram


Archive
+ main(args: String[]): void
CommandHandler
- studentMap: Map<Integer, Student>
- spellbookMap: Map<Integer, SpellBook>
- errorChecker: CommonErrorChecker
+ handleCommand(userInput: String): boolean
- printHelpString(): void
- manageListCommand(userInputParts: String[]): void
- manageNumberCopies(): void
- searchSpellbooksByAttribute(extractor: Function<SpellBook, String>, userInputParts: String[]): void
- displaySpellbook(userInputParts: String[]): void
- displaySpellbookHistory(userInputParts: String[]): void
- manageStudent(userInputParts: String[]): void
- rentSpellbook(userInputParts: String[]): void
- relinquish(userInputParts: String[]): void
- relinquishAll(userInputParts: String[]): void
- manageAddCommands(userInputParts: String[]): void
+ saveToFile(userInputParts: String[]): void
+ printCommon(userInputParts: String[]): void
Student
- nextStudentId: int
- number: int
- name: String
- currentSpellbooks: List<Integer>
- history: List<Integer>
+ addToCurrentSpellbooks(serialNumber: int): void
+ addToHistory(serialNumber: int): void
+ clearCurrentSpellbooks(): void
+ removeSpellbook(serialNumber: int): void
SpellBook
- serialNumber: int
- title: String
- inventor: String
- type: String
- studentRenting: int
- history: List<Integer>
+ getPrintableForm (longPresent: boolean = false): String
+ getRentingStatus(): String
+ addToHistory(studentNumber: int): void
CommonErrorChecker
- studentMap: Map<Integer, Student>
- spellbookMap: Map<Integer, SpellBook>
+ checkEmptyStudent(): boolean
+ checkEmptySpellbooks(): boolean
+ isMissingStudent(studentNumber: int): boolean
+ isMissingSpellbook(serialNumber: int): boolean
+ spellbookNotRentedByStudent(serialNumber: int, studentNumber: int): boolean