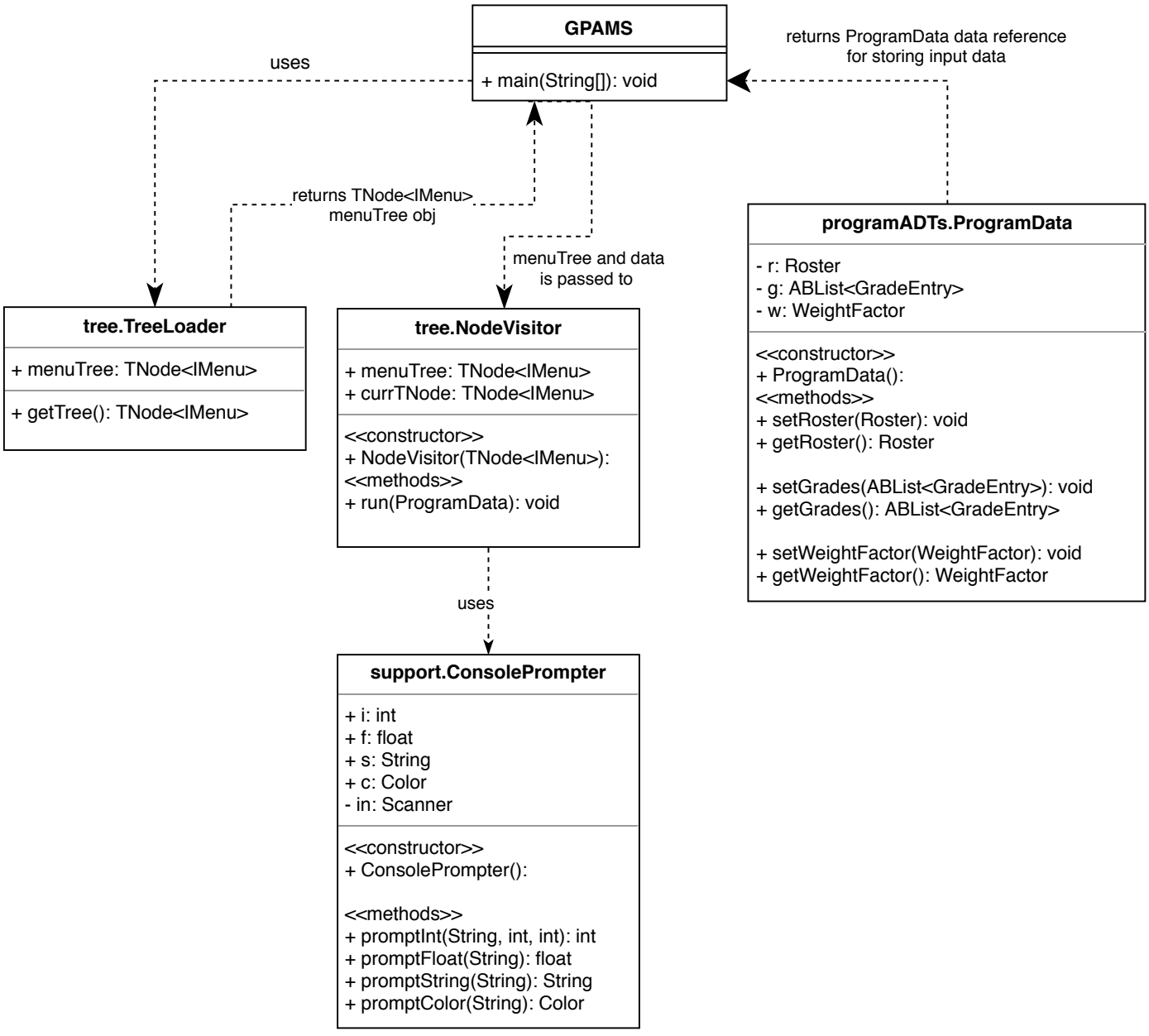
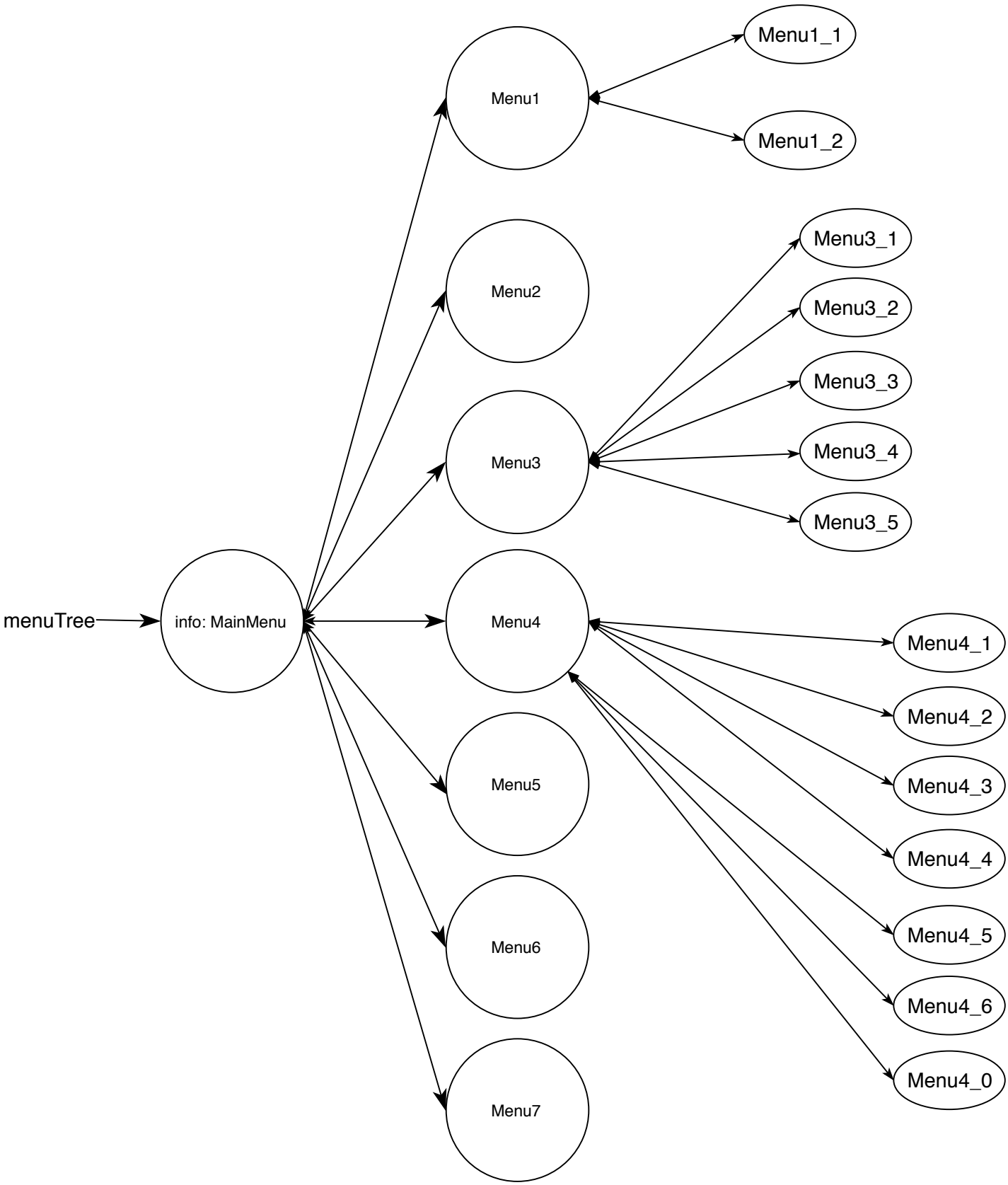
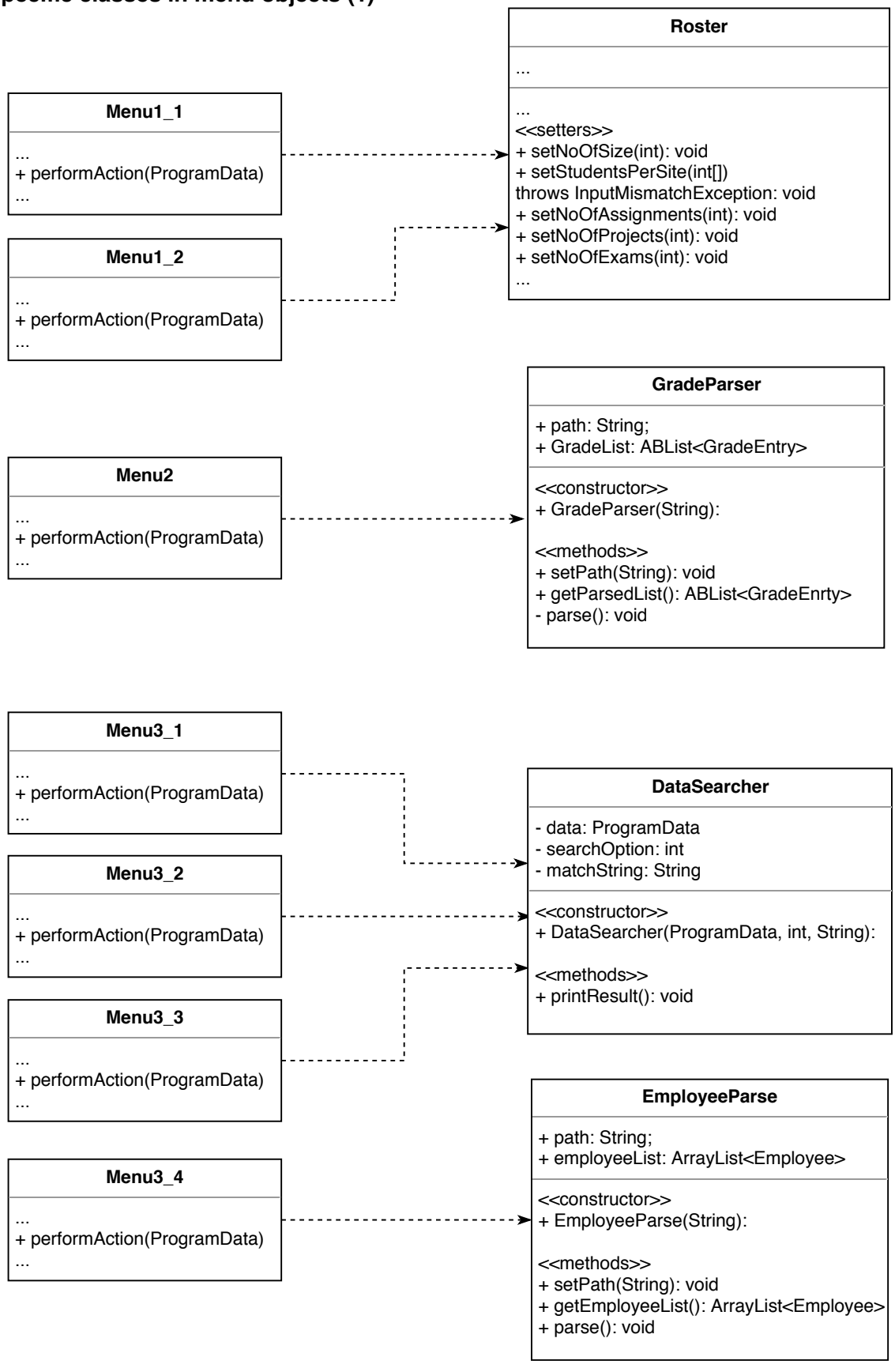

GPAMS
+ main(String[]): void
uses
returns ProgramData data reference
for storing input data
returns TNode<IMenu>
menuTree obj
menuTree and data
is passed to
tree.TreeLoader
+ menuTree: TNode<IMenu>
+ getTree(): TNode<IMenu>
tree.NodeVisitor
+ menuTree: TNode<IMenu>
+ currTNode: TNode<IMenu>
<<constructor>>
+ NodeVisitor(TNode<IMenu>):
<<methods>>
+ run(ProgramData): void
programADTs.ProgramData
- r: Roster
- g: ABList<GradeEntry>
- w: WeightFactor
<<constructor>>
+ ProgramData():
<<methods>>
+ setRoster(Roster): void
+ getRoster(): Roster
+ setGrades(ABList<GradeEntry>): void
+ getGrades(): ABList<GradeEntry>
+ setWeightFactor(WeightFactor): void
+ getWeightFactor(): WeightFactor
uses
support.ConsolePrompter
+ i: int
+ f: float
+ s: String
+ c: Color
- in: Scanner
<<constructor>>
+ ConsolePrompter():
<<methods>>
+ promptInt(String, int, int): int
+ promptFloat(String): float
+ promptString(String): String
+ promptColor(String): Color

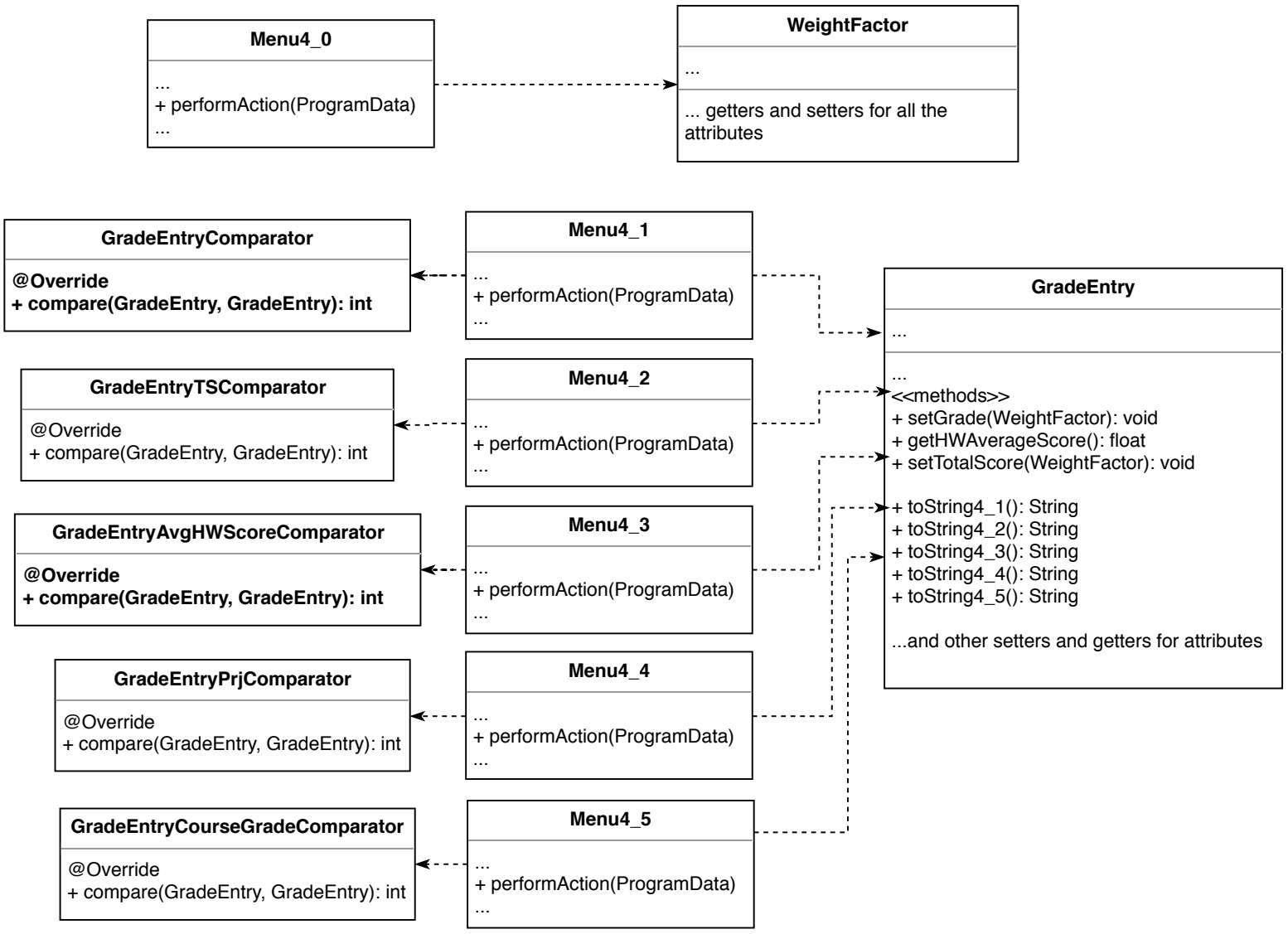
Tree ADT created by TreeLoader.getTree()

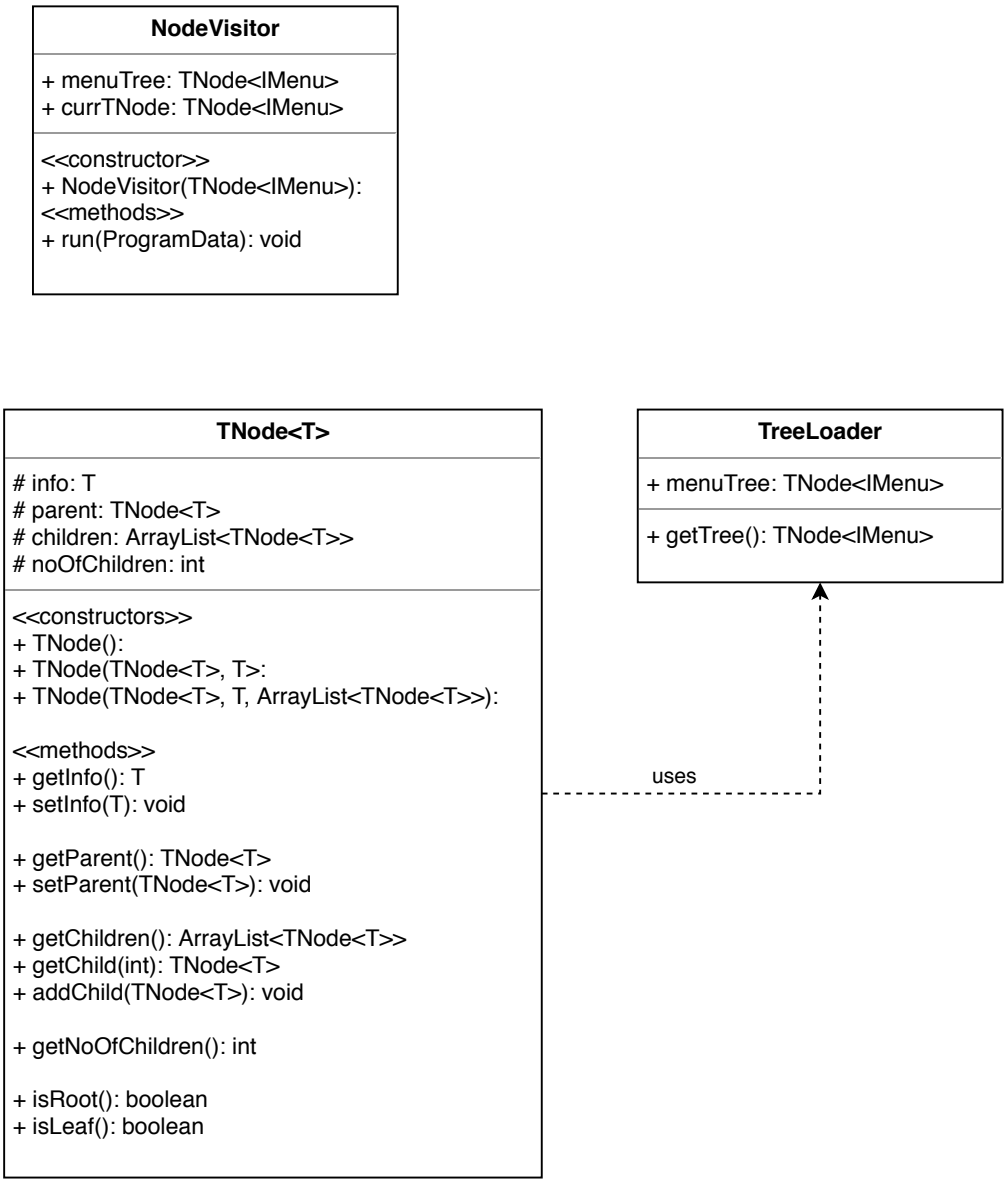


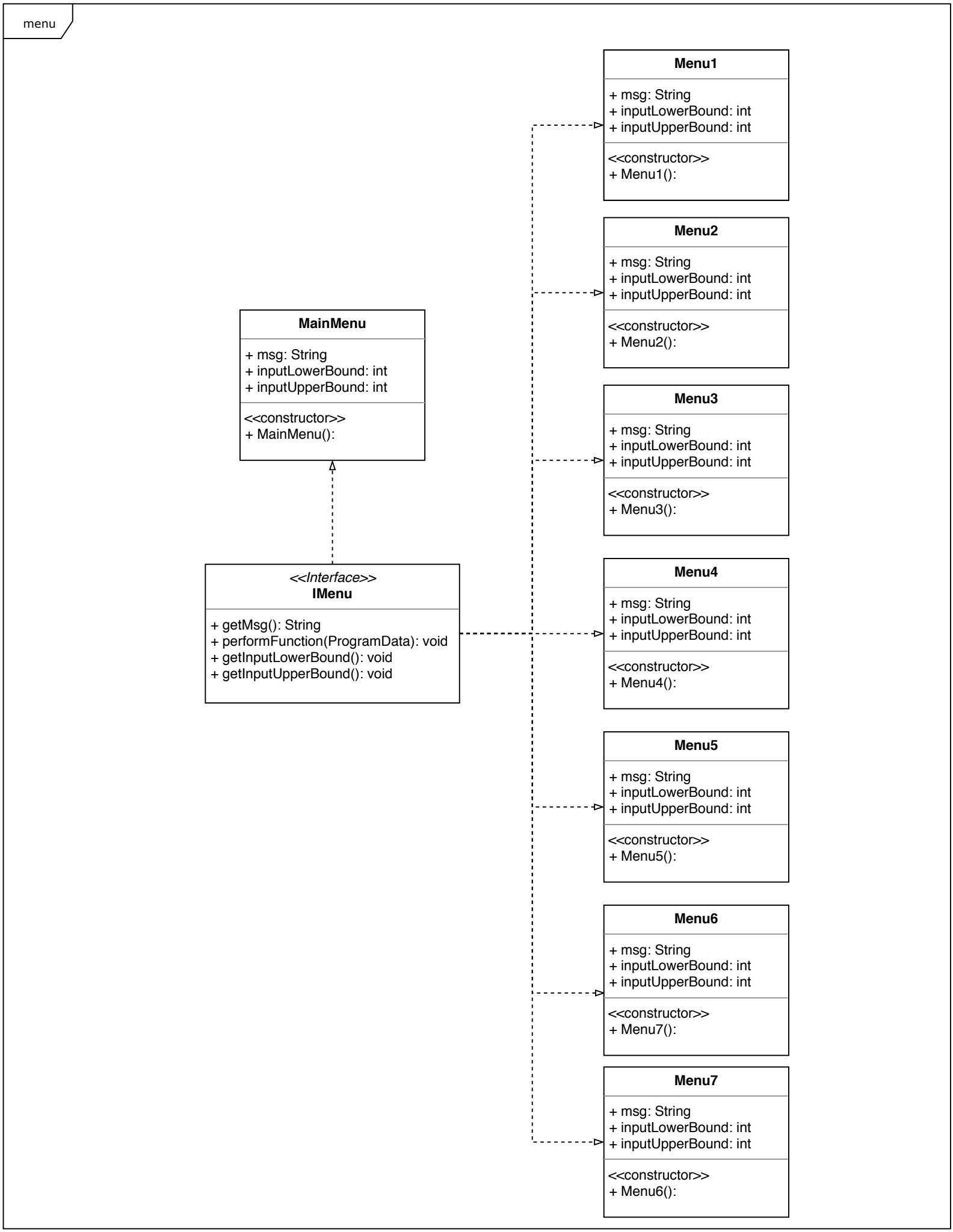
Use of specific classes in menu objects (1)



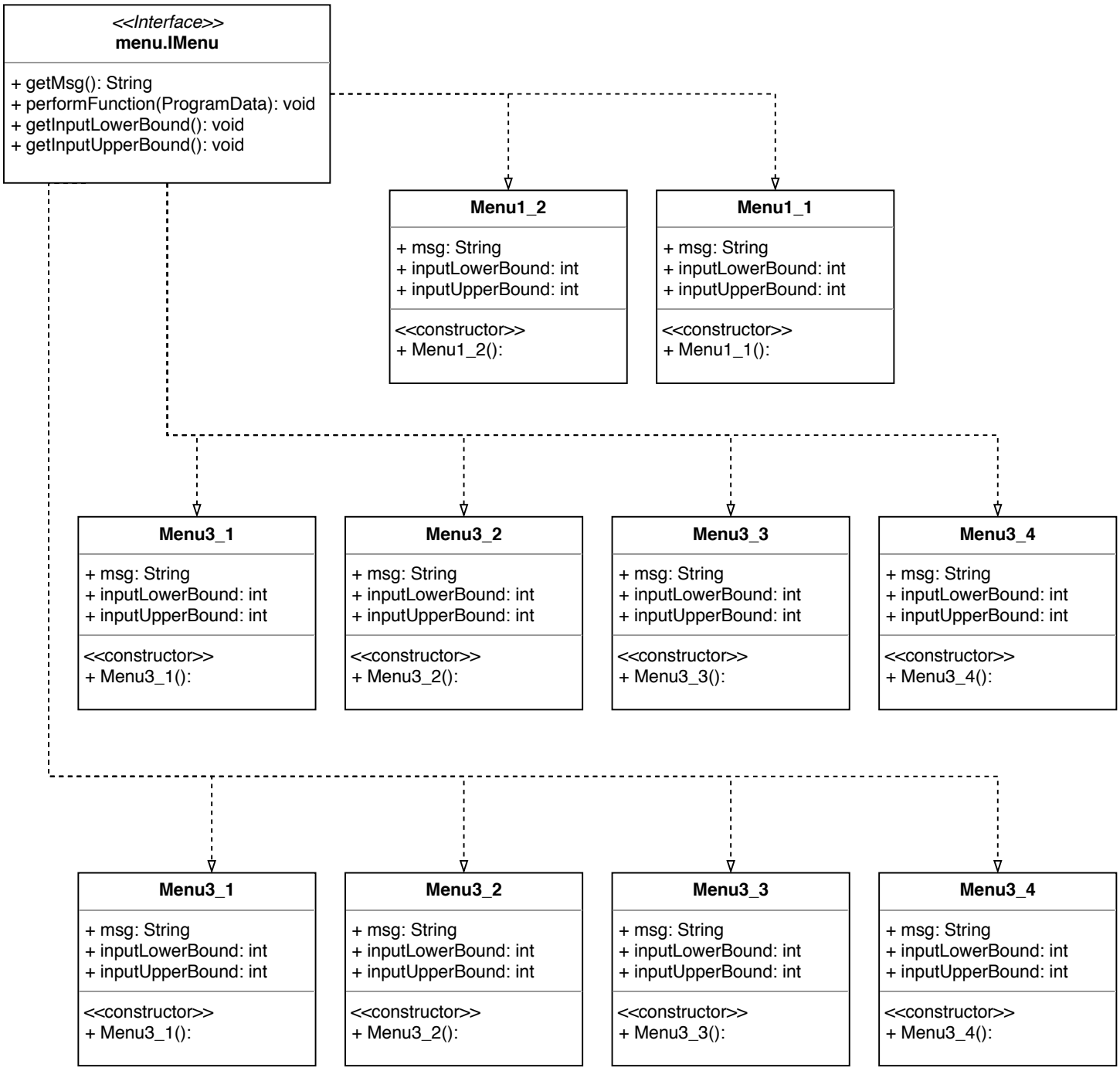
Use of specific classes in menu objects (2)



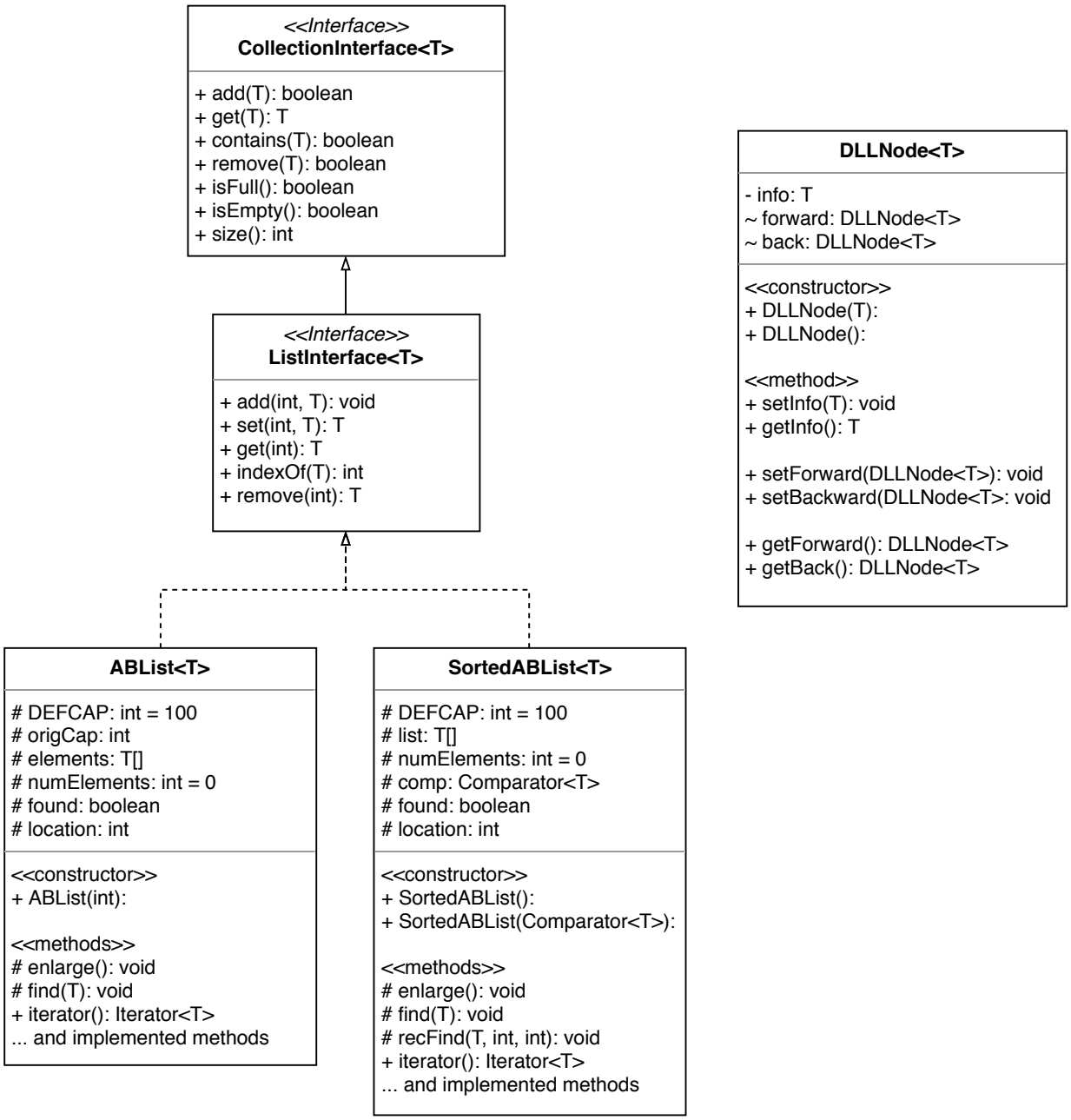

NodeVisitor
+ menuTree: TNode<IMenu>
+ currTNode: TNode<IMenu>
<<constructor>>
+ NodeVisitor(TNode<IMenu>):
<<methods>>
+ run(ProgramData): void
TNode<T>
info: T
parent: TNode<T>
children: ArrayList<TNode<T>>
noOfChildren: int
<<constructors>>
+ TNode():
+ TNode(TNode<T>, T>:
+ TNode(TNode<T>, T, ArrayList<TNode<T>>):
<<methods>>
+ getInfo(): T
+ setInfo(T): void
+ getParent(): TNode<T>
+ setParent(TNode<T>): void
+ getChildren(): ArrayList<TNode<T>>
+ getChild(int): TNode<T>
+ addChild(TNode<T>): void
+ getNoOfChildren(): int
+ isRoot(): boolean
+ isLeaf(): boolean
TreeLoader
+ menuTree: TNode<IMenu>
+ getTree(): TNode<IMenu>
uses

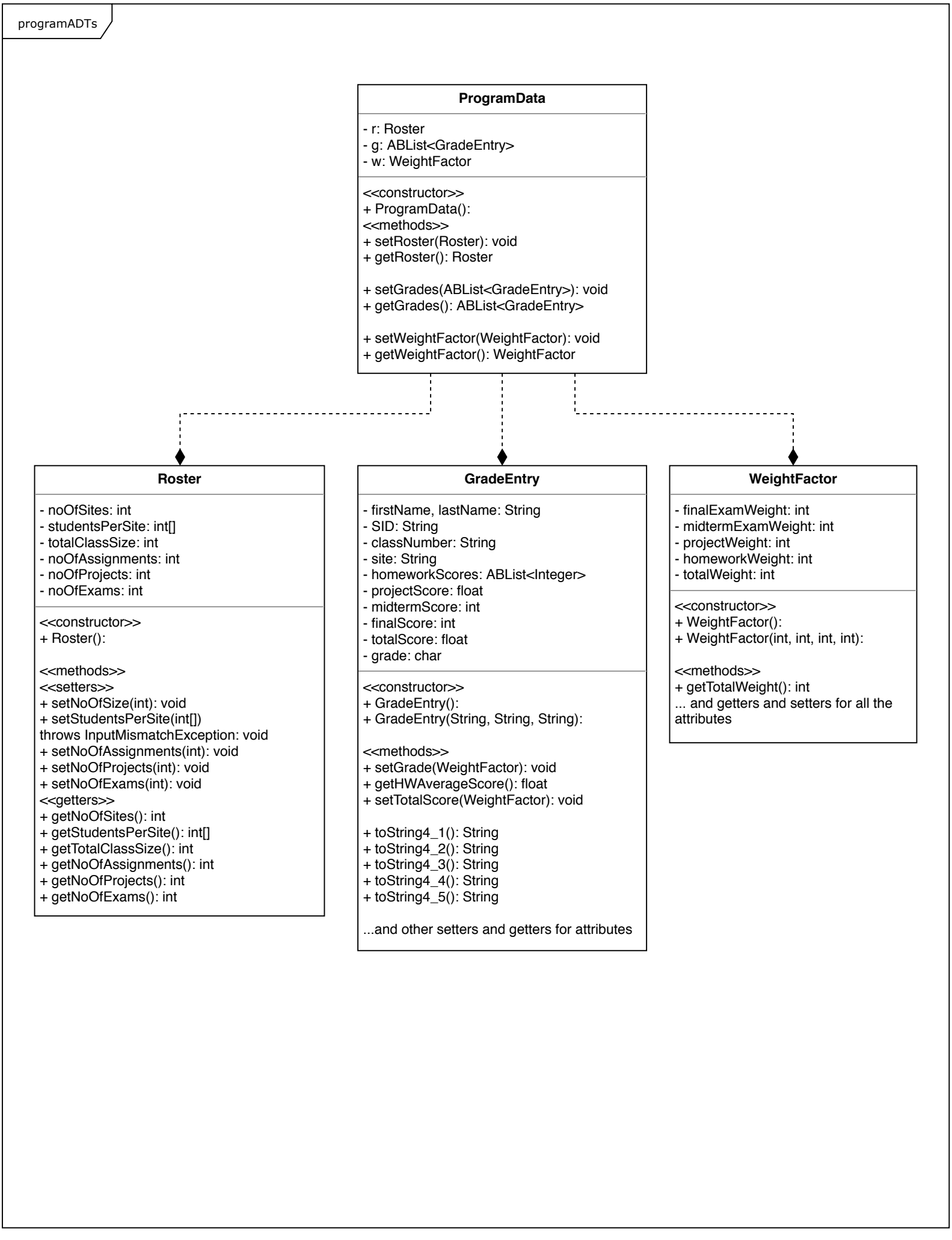

menu
Menu1
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ Menu1():
Menu2
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ Menu2():
MainMenu
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ MainMenu():
Menu3
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ Menu3():
<<Interface>>
IMenu
+ getMsg(): String
+ performFunction(ProgramData): void
+ getInputLowerBound(): void
+ getInputUpperBound(): void
Menu4
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ Menu4():
Menu5
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ Menu5():
Menu6
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ Menu7():
Menu7
+ msg: String
+ inputLowerBound: int
+ inputUpperBound: int
<<constructor>>
+ Menu6():

menu.submenu

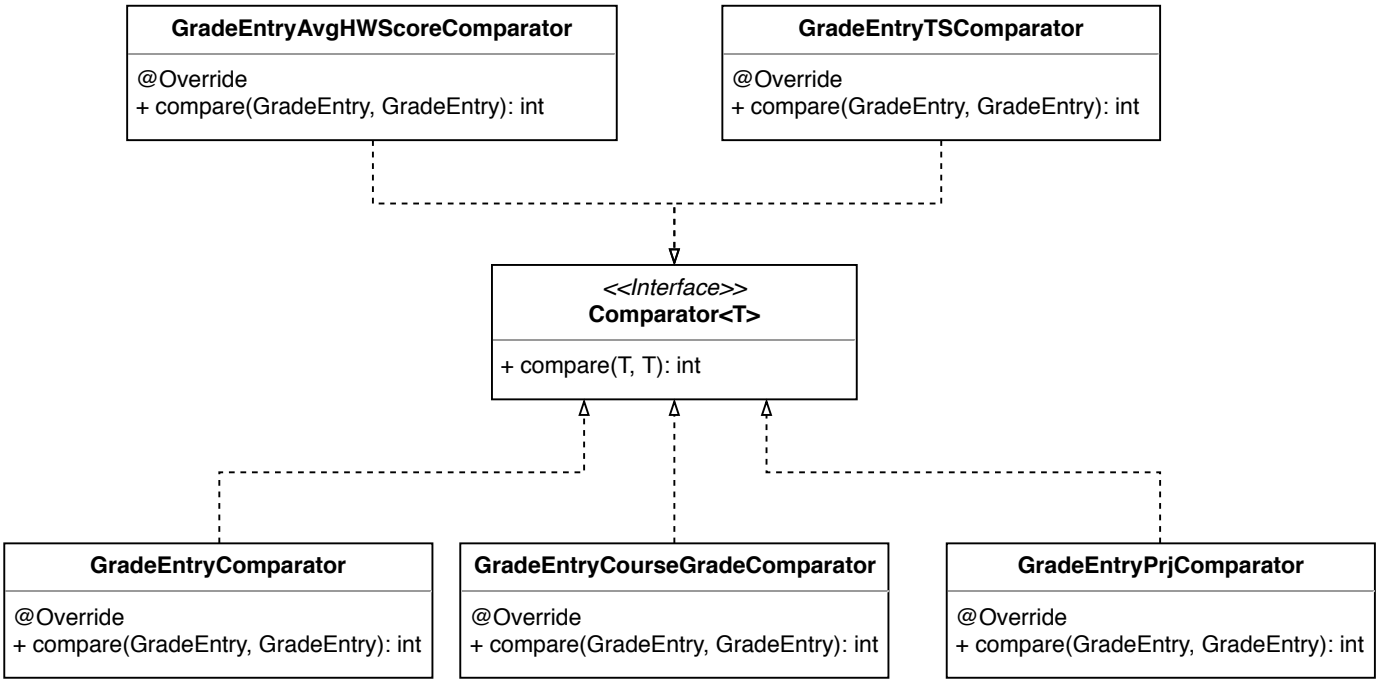


customADTs


<<Interface>>
CollectionInterface<T>
+ add(T): boolean
+ get(T): T
+ contains(T): boolean
+ remove(T): boolean
+ isFull(): boolean
+ isEmpty(): boolean
+ size(): int
<<Interface>>
ListInterface<T>
+ add(int, T): void
+ set(int, T): T
+ get(int): T
+ indexOf(T): int
+ remove(int): T
DLLNode<T>
- info: T
~ forward: DLLNode<T>
~ back: DLLNode<T>
<<constructor>>
+ DLLNode(T):
+ DLLNode():
<<method>>
+ setInfo(T): void
+ getInfo(): T
+ setForward(DLLNode<T>): void
+ setBackward(DLLNode<T>: void
+ getForward(): DLLNode<T>
+ getBack(): DLLNode<T>
ABList<T>
DEFCAP: int = 100
origCap: int
elements: T[]
numElements: int = 0
found: boolean
location: int
<<constructor>>
+ ABList(int):
<<methods>>
enlarge(): void
find(T): void
+ iterator(): Iterator<T>
... and implemented methods
SortedABList<T>
DEFCAP: int = 100
list: T[]
numElements: int = 0
comp: Comparator<T>
found: boolean
location: int
<<constructor>>
+ SortedABList():
+ SortedABList(Comparator<T>):
<<methods>>
enlarge(): void
find(T): void
recFind(T, int, int): void
+ iterator(): Iterator<T>
... and implemented methods


programADTs
ProgramData
- r: Roster
- g: ABList<GradeEntry>
- w: WeightFactor
<<constructor>>
+ ProgramData():
<<methods>>
+ setRoster(Roster): void
+ getRoster(): Roster
+ setGrades(ABList<GradeEntry>): void
+ getGrades(): ABList<GradeEntry>
+ setWeightFactor(WeightFactor): void
+ getWeightFactor(): WeightFactor
Roster
- noOfSites: int
- studentsPerSite: int[]
- totalClassSize: int
- noOfAssignments: int
- noOfProjects: int
- noOfExams: int
<<constructor>>
+ Roster():
<<methods>>
<<setters>>
+ setNoOfSize(int): void
+ setStudentsPerSite(int[])
throws InputMismatchException: void
+ setNoOfAssignments(int): void
+ setNoOfProjects(int): void
+ setNoOfExams(int): void
<<getters>>
+ getNoOfSites(): int
+ getStudentsPerSite(): int[]
+ getTotalClassSize(): int
+ getNoOfAssignments(): int
+ getNoOfProjects(): int
+ getNoOfExams(): int
GradeEntry
- firstName, lastName: String
- SID: String
- classNumber: String
- site: String
- homeworkScores: ABList<Integer>
- projectScore: float
- midtermScore: int
- finalScore: int
- totalScore: float
- grade: char
<<constructor>>
+ GradeEntry():
+ GradeEntry(String, String, String):
<<methods>>
+ setGrade(WeightFactor): void
+ getHWAverageScore(): float
+ setTotalScore(WeightFactor): void
+ toString4_1(): String
+ toString4_2(): String
+ toString4_3(): String
+ toString4_4(): String
+ toString4_5(): String
...and other setters and getters for attributes
WeightFactor
- finalExamWeight: int
- midtermExamWeight: int
- projectWeight: int
- homeworkWeight: int
- totalWeight: int
<<constructor>>
+ WeightFactor():
+ WeightFactor(int, int, int, int):
<<methods>>
+ getTotalWeight(): int
... and getters and setters for all the attributes

programADTs.GradeEntryComparators



support

ConsolePrompter

+ i: int
+ f: float
+ s: String
+ c: Color
- in: Scanner

<<constructor>>
+ ConsolePrompter():

<<methods>>
+ promptInt(String, int, int): int
+ promptFloat(String): float
+ promptString(String): String
+ promptColor(String): Color

DataSearcher

- data: ProgramData
- searchOption: int
- matchString: String

<<constructor>>
+ DataSearcher(ProgramData, int, String):

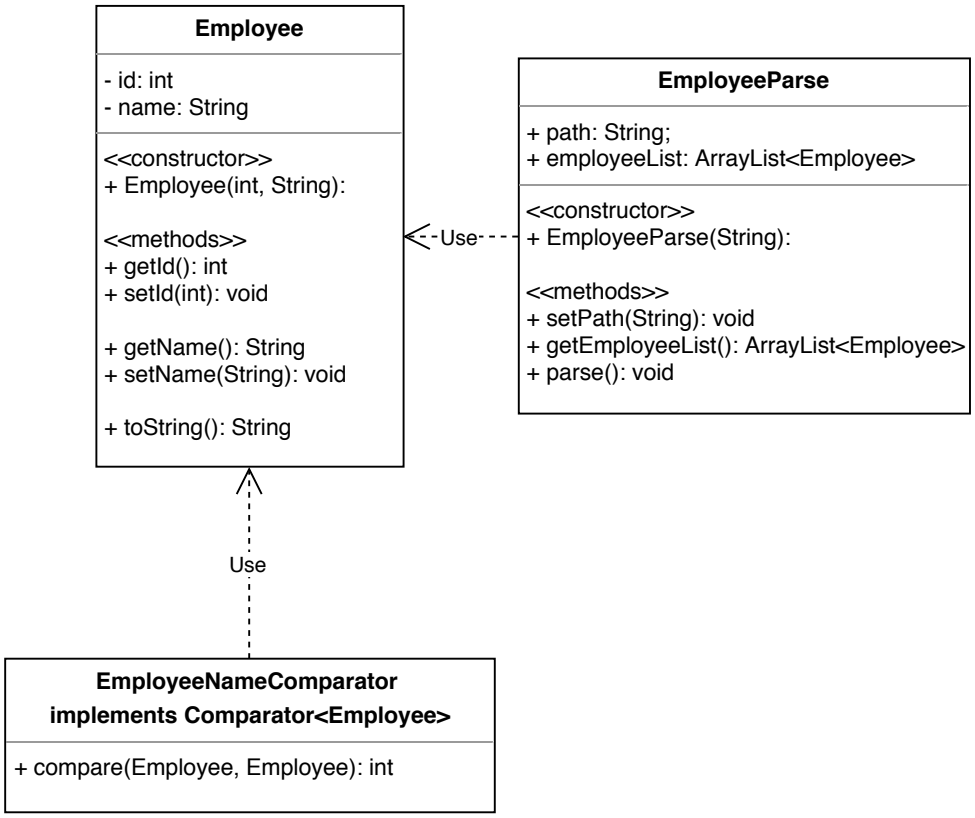
<<methods>>
+ printResult(): void

GradeParser

+ path: String;
+ GradeList: ABList<GradeEntry>

<<constructor>>
+ GradeParser(String):

<<methods>>
+ setPath(String): void
+ getParsedList(): ABList<GradeEnrty>
- parse(): void


Employee
- id: int
- name: String
<<constructor>>
+ Employee(int, String):
<<methods>>
+ getId(): int
+ setId(int): void
+ getName(): String
+ setName(String): void
+ toString(): String
EmployeeParse
+ path: String;
+ employeeList: ArrayList<Employee>
<<constructor>>
+ EmployeeParse(String):
<<methods>>
+ setPath(String): void
+ getEmployeeList(): ArrayList<Employee>
+ parse(): void
Use
Use
EmployeeNameComparator
implements Comparator<Employee>
+ compare(Employee, Employee): int