Book Application UML Diagram

«interface» **BookInterface**
+ displaLastSixBooks(allBooks: Book[], count: int) : void + getGenreCount(g: String): int + getTotalCost(): double

↑ Implements

«abstract class» *Book*
+ title : String + author : String + genre : String + cost : double
+ Book(t : String, a : String, g : String, c : double) + getTitle() : String + getAuthor() : String + getGenre() : String + getCost() : double + displayInfo(): void + abstract computeCost(): double + abstract getTotalCost(): double + getGenreCount(g : String): int

↑ Extends   ↑ Extends

PrintedBook
- pages: int - static totalPages: int - static printedCount: int - static printedCostTotal: double - static lastPrinted: PrintedBook[3]
+ PrintedBook(t:String, a:String, g:String, p:int) + getPages(): int + computeCost(): double + displayInfo(): void + getTotalCost(): double + getGenreCount(g:String): int - static updateLastPrinted(pb: PrintedBook): void + static getAvePages(): double + static getTotalPages(): int + static getPrintedCount(): int + static getTotalPrintedCost(): double + static displayLast3Printed(): void

AudioBook
- minutes: int - static totalMinutes: int - static audioCount: int - static audioCostTotal: double - static lastAudio: AudioBook[3]
+ AudioBook(t:String, a:String, g:String, m:int) + getMinutes(): int + computeCost(): double + displayInfo(): void + getTotalCost(): double + getGenreCount(g:String): int - static updateLastAudio(ab: AudioBook): void + static getAveMinutes(): double + static getTotalMinutes(): int + static getAudioCount(): int + static getTotalAudioCost(): double + static displayLast3Audio(): void