Book

- author: String
- cost: double
- countExemplars: int
- dateLastReceipt: Date
- dateRelease: Date
- idBook: int
- title: String

+ Book(String, String, Date, double)
+ getAuthor(): String
+ getCost(): double
+ getCountExemplars(): int
+ getDateLastReceipt(): Date
+ getDateRelease(): Date
+ getId(): int
+ getTitle(): String
+ setCost(int): void
+ setCountExemplars(int): void
+ setDateLastReceipt(Date): void
+ setId(int): void

Application

- book: Book
- countExemplar: int
- dateCompletion: Date
- dateCreation: Date
- idApplication: int
- status: int
+ STATUS_COMPLETE: int = 2 {readOnly}
+ STATUS_CREATE: int = 1 {readOnly}

+ Application()
+ Application(Book, int, Date)
+ Application(Book, int, Date, int)
+ getBook(): Book
+ getCountExemplar(): int
+ getDateCompletion(): Date
+ getDateCreation(): Date
+ getId(): int
+ getStatus(): int
+ setBook(Book): void
+ setCountExemplar(int): void
+ setDateCompletion(Date): void
+ setDateCreation(Date): void
+ setId(int): void
+ setStatus(int): void

(partially cut-off class, right edge)

- APPLIC...
- applicat...
- fileWor...
- lastInse...
+ addApp...
+ Applicat...
+ collectF...
+ deleteA...
- genera...
+ getAllAp...
+ getAppl...
- getLast...
- parseFil...
- readFro...
+ updateA...
- writeTo...

-book

-applications

Request

- book: Book
- countRequests: int

+ calculateCountRequests(): int
+ getBook(): Book
+ getCountRequests(): int
+ Request(Book)

BookStore

- BOOK_FILE: String = "books.txt" {readOnly}
- books: Book ([])
- fileWorker: FileWorker {readOnly}
- lastInsertedId: int

+ addBook(Book): void
+ BookStore()
- collectFileLine(Book): String
+ deleteBook(Book): void
- generateId(): int
+ getAllBooks(): Book[]
+ getBook(int): Book
- getLastId(): int
- parseFileLine(String): Book
- readFromFile(): Book[]
+ updateBook(Book): void
- writeToFile(Book[]): void

«interface»
IApplica...

+ addApplication
+ deleteApplicat...
+ getAllApplicat...
+ getApplication
+ updateApplica...

«interface»
IBookStore

+ addBook(Book): void
+ deleteBook(Book): void
+ getAllBooks(): Book[]
+ getBook(int): Book
+ updateBook(Book): void

-bookStore

-book   **-books**   **-book**

OrderItem

- book: Book
- countExemplars: int

+ getBook(): Book
+ getCount(): int
+ OrderItem(Book, int)

Order

- canceled: boolean = false
- dateCompletion: Date
- dateCreation: Date
- idOrder: int
- items: ArrayList<OrderItem> = new ArrayList<O...
+ MIN_DAYS_TO_COMPLETE: int = 2 {readOnly}
- status: int
+ STATUS_CANCEL: int = 4 {readOnly}
+ STATUS_COMPLETE: int = 3 {readOnly}
+ STATUS_CREATE: int = 1 {readOnly}
+ STATUS_NOT_STAFF: int = 2 {readOnly}

+ addItem(OrderItem): void
+ getCanceled(): boolean
+ getDateCompletion(): Date
+ getDateCreation(): Date
+ getId(): int
+ getItems(): ArrayList<OrderItem>
+ getStatus(): int
+ getTotalPrice(): double
+ Order()
+ Order(ArrayList<OrderItem>)
+ Order(Date)
+ Order(ArrayList<OrderItem>, Date)
+ setCanceled(boolean): void
+ setDateCompletion(Date): void
+ setDateCreation(Date): void
+ setId(int): void
+ setStatus(int): void

-orders

ApplicationStore

- …ATION_FILE: String = "applications.txt" {readOnly}
- …ions: Application ([])
- …ker: FileWorker {readOnly}
- …rtedId: int

- …lication(Application): void
- …ionStore()
- …ileLine(Application): String
- …pplication(Application): void
- …teId(): int
- …pplications(): Application[]
- …ication(int): Application
- …Id(): int
- …leLine(String): Application
- …omFile(): Application[]
- …Application(Application): void
- …File(Application[]): void

«interface»
…ationStore

- …n(Application): void
- …ion(Application): void
- …ions(): Application[]
- …(int): Application
- …ion(Application): void

«interface»
IOrderStore

- + addOrder(Order): void
- + deleteOrder(Order): void
- + getAllOrders(): Order[]
- + getOrder(int): Order
- + updateOrder(Order): void

OrderStore

- – fileWorker: FileWorker {readOnly}
- – lastInsertedId: int
- – ORDER_FILE: String = "orders.txt" {readOnly}
- – orders: Order ([])

- + addOrder(Order): void
- – collectFileLine(Order): String
- + deleteOrder(Order): void
- – generateId(): int
- + getAllOrders(): Order[]
- – getLastId(): int
- + getOrder(int): Order
- + OrderStore()
- – parseFileLine(String): Order
- – readFromFile(): Order[]
- + updateOrder(Order): void
- – writeToFile(Order[]): void

BookShop

- + ADDITIONAL_EXEMPLARS: int = 5 {readOnly}
- – applicationStore: IApplicationStore
- – bookStore: IBookStore
- + OLD_BOOKS_MONTHS_NUMBER: int = 6 {readOnly}
- + ORDERBY_ALPHABET: int = 1 {readOnly}
- + ORDERBY_COST: int = 3 {readOnly}
- + ORDERBY_COUNT_EXEMPLARS: int = 5 {readOnly}
- + ORDERBY_COUNT_REQUESTS: int = 6 {readOnly}
- + ORDERBY_DATE_COMPLETION: int = 8 {readOnly}
- + ORDERBY_DATE_CREATION: int = 7 {readOnly}
- + ORDERBY_DATE_LAST_RECEIPT: int = 4 {readOnly}
- + ORDERBY_DATE_RELEASE: int = 2 {readOnly}
- + ORDERBY_STATUS: int = 10 {readOnly}
- + ORDERBY_TOTAL_PRICE: int = 9 {readOnly}
- – orderStore: IOrderStore

- + addApplication(Application): void
- + addBook(Book): void
- + addBookToStock(Book, int): void
- + addOrderWithCheckBookExistance(Order): void
- + BookShop()
- + cancelTheOrder(Order): void
- – checkApplicationsCompletion(): void
- – checkOrdersCompletion(): void
- + completeTheApplication(Application): void
- + completeTheOrder(Order): void
- + deleteApplication(Application): void
- + deleteBook(Book): void
- + deleteBookFromStock(Book): void
- + deleteOrder(Order): void
- – generateApplication(Book, int): Application
- + getApplication(int): Application
- + getApplications(boolean): ArrayList<Application>
- + getBook(int): Book
- + getBooks(int): ArrayList<Book>
- + getBooksRequests(int): ArrayList<Request>
- + getCompleteOrdersCount(Date, Date): int
- + getOldBooks(int): ArrayList<Book>
- + getOrder(int): Order
- + getOrders(Date, Date, int, boolean): ArrayList<Order>
- + getTotalOrdersPrice(Date, Date): double
- + updateApplication(Application): void
- + updateBook(Book): void
- + updateOrder(Order): void

«interface»
IBookShop

- + addApplication(Application): void
- + addBook(Book): void
- + addBookToStock(Book, int): void
- + addOrderWithCheckBookExistance(Orde…
- + cancelTheOrder(Order): void
- + completeTheApplication(Application): voi…
- + completeTheOrder(Order): void
- + deleteApplication(Application): void
- + deleteBook(Book): void
- + deleteBookFromStock(Book): void
- + deleteOrder(Order): void
- + getApplication(int): Application
- + getApplications(boolean): ArrayList<Appli…
- + getBook(int): Book
- + getBooks(int): ArrayList<Book>
- + getBooksRequests(int): ArrayList<Reques…
- + getCompleteOrdersCount(Date, Date): in…
- + getOldBooks(int): ArrayList<Book>
- + getOrder(int): Order
- + getOrders(Date, Date, int, boolean): Arra…
- + getTotalOrdersPrice(Date, Date): double
- + updateApplication(Application): void
- + updateBook(Book): void
- + updateOrder(Order): void

-applicationStore

-orderStore