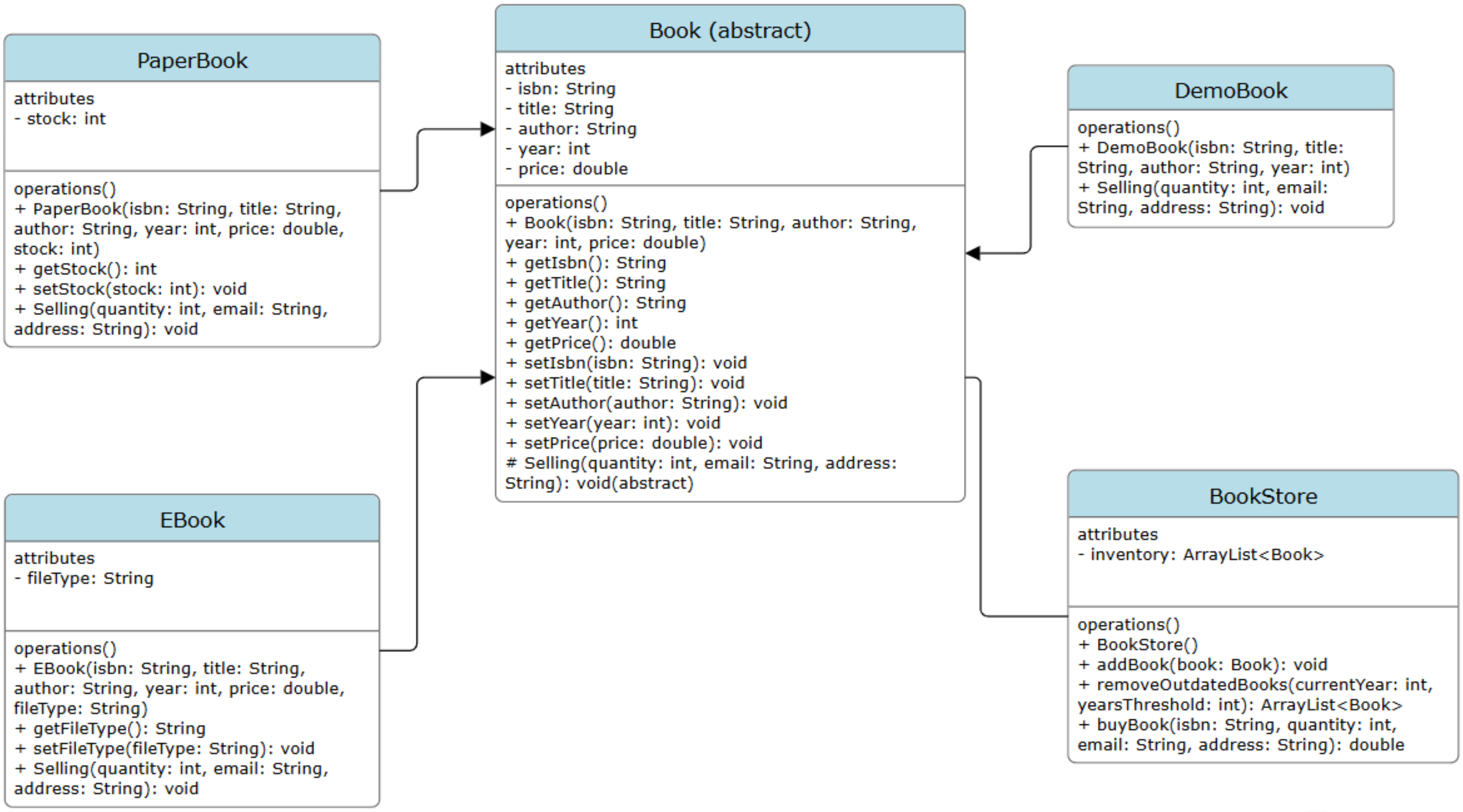
Quantum Bookstore

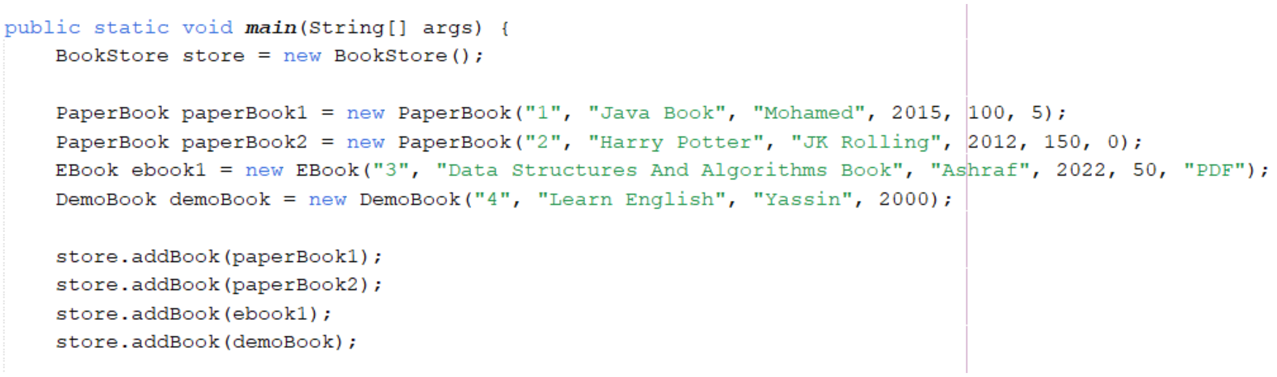
Mohamed Ashraf Khalaf Hafez

1st UML Class Diagram:

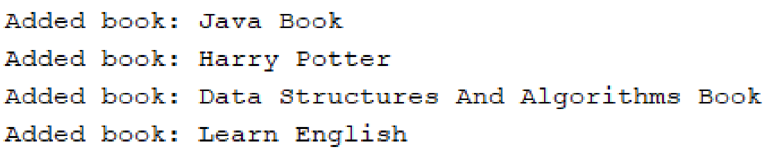

PaperBook
attributes
- stock: int
operations()
+ PaperBook(isbn: String, title: String, author: String, year: int, price: double, stock: int)
+ getStock(): int
+ setStock(stock: int): void
+ Selling(quantity: int, email: String, address: String): void
Book (abstract)
attributes
- isbn: String
- title: String
- author: String
- year: int
- price: double
operations()
+ Book(isbn: String, title: String, author: String, year: int, price: double)
+ getIsbn(): String
+ getTitle(): String
+ getAuthor(): String
+ getYear(): int
+ getPrice(): double
+ setIsbn(isbn: String): void
+ setTitle(title: String): void
+ setAuthor(author: String): void
+ setYear(year: int): void
+ setPrice(price: double): void
# Selling(quantity: int, email: String, address: String): void(abstract)
DemoBook
operations()
+ DemoBook(isbn: String, title: String, author: String, year: int)
+ Selling(quantity: int, email: String, address: String): void
EBook
attributes
- fileType: String
operations()
+ EBook(isbn: String, title: String, author: String, year: int, price: double, fileType: String)
+ getFileType(): String
+ setFileType(fileType: String): void
+ Selling(quantity: int, email: String, address: String): void
BookStore
attributes
- inventory: ArrayList<Book>
operations()
+ BookStore()
+ addBook(book: Book): void
+ removeOutdatedBooks(currentYear: int, yearsThreshold: int): ArrayList<Book>
+ buyBook(isbn: String, quantity: int, email: String, address: String): double

2nd Testing:

Objects for testing:

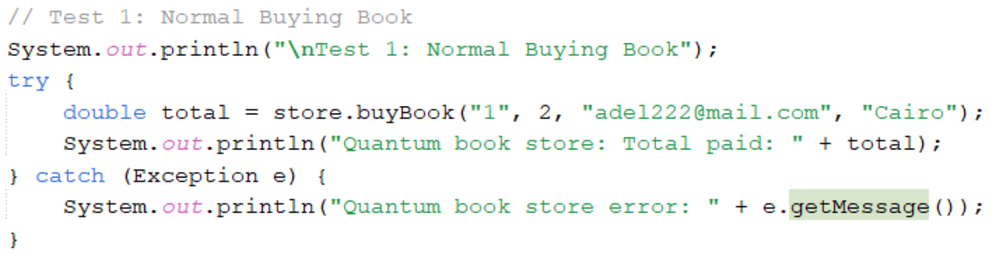


```java
public static void main(String[] args) {
    BookStore store = new BookStore();

    PaperBook paperBook1 = new PaperBook("1", "Java Book", "Mohamed", 2015, 100, 5);
    PaperBook paperBook2 = new PaperBook("2", "Harry Potter", "JK Rolling", 2012, 150, 0);
    EBook ebook1 = new EBook("3", "Data Structures And Algorithms Book", "Ashraf", 2022, 50, "PDF");
    DemoBook demoBook = new DemoBook("4", "Learn English", "Yassin", 2000);

    store.addBook(paperBook1);
    store.addBook(paperBook2);
    store.addBook(ebook1);
    store.addBook(demoBook);
```



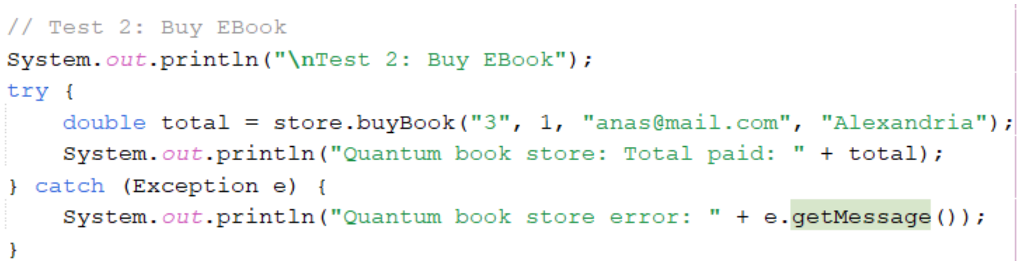
```
Added book: Java Book
Added book: Harry Potter
Added book: Data Structures And Algorithms Book
Added book: Learn English
```

Test 1: Normal Buying Book



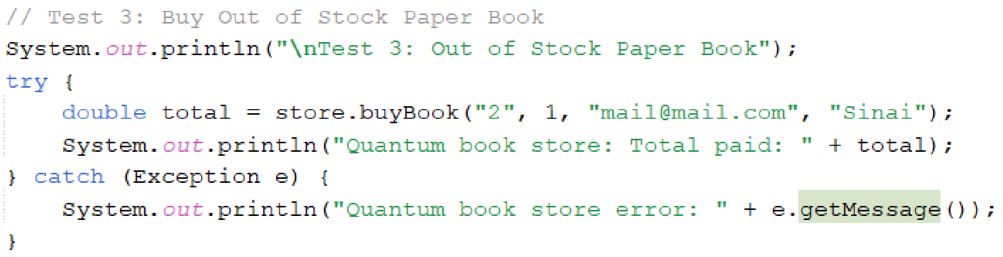
```java
// Test 1: Normal Buying Book
System.out.println("\nTest 1: Normal Buying Book");
try {
    double total = store.buyBook("1", 2, "adel222@mail.com", "Cairo");
    System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
    System.out.println("Quantum book store error: " + e.getMessage());
}
```

Test 2: Buy EBook



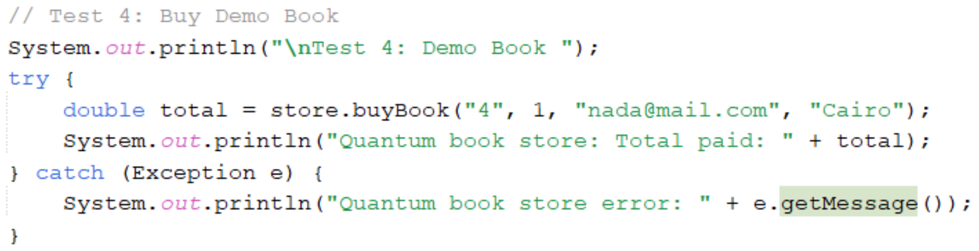
```java
// Test 2: Buy EBook
System.out.println("\nTest 2: Buy EBook");
try {
    double total = store.buyBook("3", 1, "anas@mail.com", "Alexandria");
    System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
    System.out.println("Quantum book store error: " + e.getMessage());
}
```

Test 3: Out of Stock Paper Book



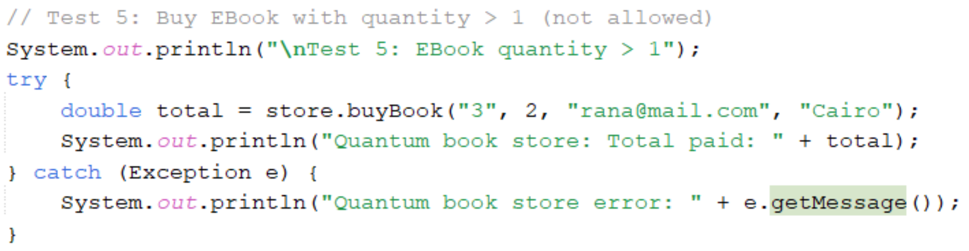
```java
// Test 3: Buy Out of Stock Paper Book
System.out.println("\nTest 3: Out of Stock Paper Book");
try {
    double total = store.buyBook("2", 1, "mail@mail.com", "Sinai");
    System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
    System.out.println("Quantum book store error: " + e.getMessage());
}
```

Test 4: Demo Book



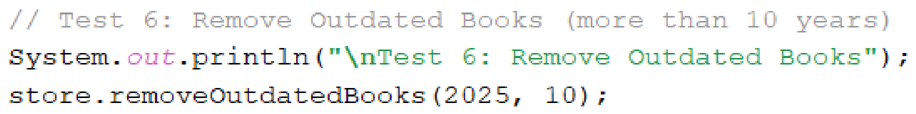
```java
// Test 4: Buy Demo Book
System.out.println("\nTest 4: Demo Book ");
try {
    double total = store.buyBook("4", 1, "nada@mail.com", "Cairo");
    System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
    System.out.println("Quantum book store error: " + e.getMessage());
}
```

Test 5: EBook quantity > 1



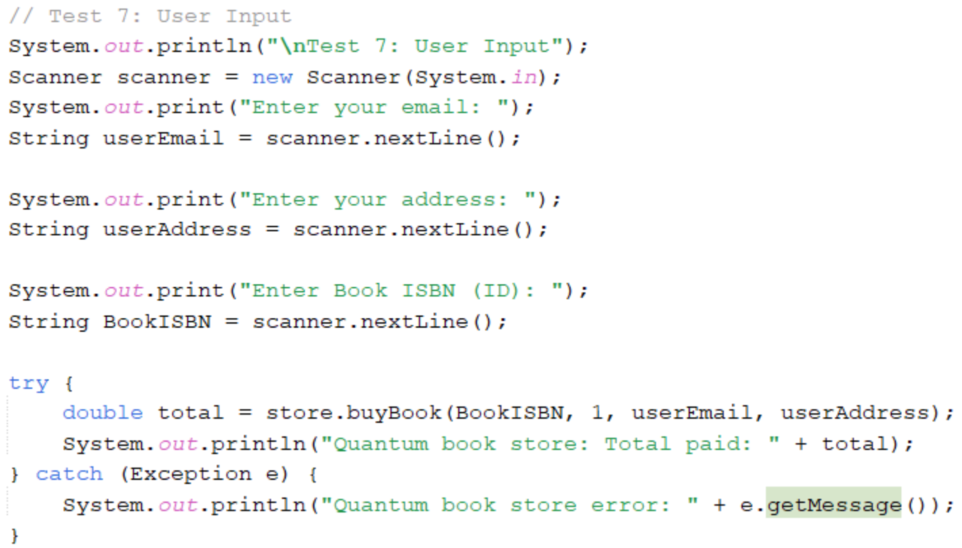
```java
// Test 5: Buy EBook with quantity > 1 (not allowed)
System.out.println("\nTest 5: EBook quantity > 1");
try {
    double total = store.buyBook("3", 2, "rana@mail.com", "Cairo");
    System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
    System.out.println("Quantum book store error: " + e.getMessage());
}
```

Test 6: Remove Outdated Books



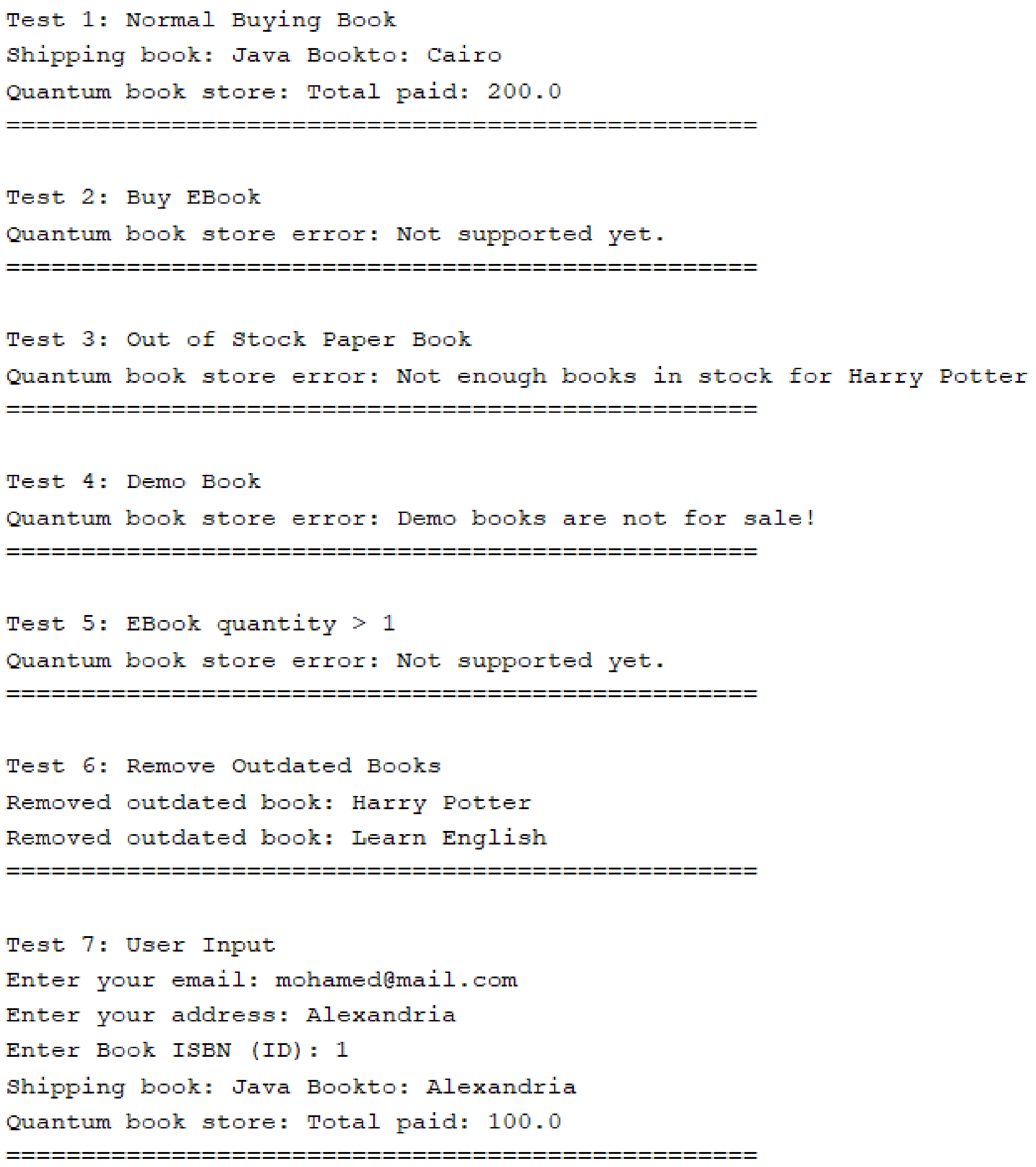
```java
// Test 6: Remove Outdated Books (more than 10 years)
System.out.println("\nTest 6: Remove Outdated Books");
store.removeOutdatedBooks(2025, 10);
```

Test 7: User Input



```java
// Test 7: User Input
System.out.println("\nTest 7: User Input");
Scanner scanner = new Scanner(System.in);
System.out.print("Enter your email: ");
String userEmail = scanner.nextLine();

System.out.print("Enter your address: ");
String userAddress = scanner.nextLine();

System.out.print("Enter Book ISBN (ID): ");
String BookISBN = scanner.nextLine();

try {
    double total = store.buyBook(BookISBN, 1, userEmail, userAddress);
    System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
    System.out.println("Quantum book store error: " + e.getMessage());
}
```

Output:



```
Test 1: Normal Buying Book
Shipping book: Java Bookto: Cairo
Quantum book store: Total paid: 200.0
==================================================

Test 2: Buy EBook
Quantum book store error: Not supported yet.
==================================================

Test 3: Out of Stock Paper Book
Quantum book store error: Not enough books in stock for Harry Potter
==================================================

Test 4: Demo Book
Quantum book store error: Demo books are not for sale!
==================================================

Test 5: EBook quantity > 1
Quantum book store error: Not supported yet.
==================================================

Test 6: Remove Outdated Books
Removed outdated book: Harry Potter
Removed outdated book: Learn English
==================================================

Test 7: User Input
Enter your email: mohamed@mail.com
Enter your address: Alexandria
Enter Book ISBN (ID): 1
Shipping book: Java Bookto: Alexandria
Quantum book store: Total paid: 100.0
==================================================
```

3rd Classes code:

Book (Abstract):

```
package com.mycompany.quantum_bookstore;



public abstract class Book {

  protected String isbn;

  protected String title;
  protected String author;
  protected int year;
  protected double price;

  public Book(String isbn, String title, String author, int year, double price) {
    this.isbn = isbn;
    this.title = title;
    this.author = author;
    this.year = year;
    this.price = price;
  }

  public void setIsbn(String isbn) {
    this.isbn = isbn;
  }

  public void setTitle(String title) {
    this.title = title;
  }

  public void setAuthor(String author) {
    this.author = author;
  }

  public void setYear(int year) {
    this.year = year;
  }

  public void setPrice(double price) {
        this.price = price;
    }

    public String getIsbn() {
        return isbn;
    }

    public String getTitle() {
        return title;
    }

    public String getAuthor() {
        return author;
    }

    public int getYear() {
        return year;
    }

    public double getPrice() {
        return price;
    }

    abstract void Selling(int quantity, String email, String address);

}
```

PaperBook:

```
package com.mycompany.quantum_bookstore;
```

```
public class PaperBook extends Book {
  private int stock;

  public PaperBook(String isbn, String title, String author, int year, double price, int stock) {
    super(isbn, title, author, year, price);
    this.stock = stock;
  }

  public void setStock(int stock) {
    this.stock = stock;
  }

  public int getStock() {
    return stock;
  }

  @Override
  void Selling(int quantity, String email, String address) {
    if (stock < quantity) {
      throw new IllegalArgumentException("Not enough books in stock for " + title);
    }
    stock -= quantity;
    System.out.println("Shipping book: " + title + "to: " + address);
  }
}
```

EBook:

```
package com.mycompany.quantum_bookstore;

public class EBook extends Book {
    private String fileType;

    public EBook(String isbn, String title, String author, int year, double price, String fileType) {
        super(isbn, title, author, year, price);
        this.fileType = fileType;
    }

    public void setFileType(String fileType) {
        this.fileType = fileType;
    }



    public String getFileType() {
        return fileType;
    }

    @Override
    void Selling(int quantity, String email, String address) {
        throw new UnsupportedOperationException("Not supported yet.");
    }
}
```

DemoBook:

```
package com.mycompany.quantum_bookstore;

public class DemoBook extends Book {
    public DemoBook(String isbn, String title, String author, int year) {
        super(isbn, title, author, year, 0.0);
    }
```

```
  @Override
  void Selling(int quantity, String email, String address) {
    throw new IllegalArgumentException("Demo books are not for sale!");
  }
}
```

BookStore:

```
package com.mycompany.quantum_bookstore;

import java.util.ArrayList;
import java.util.Iterator;

public class BookStore {
  private ArrayList<Book> inventory;

  public BookStore() {
    inventory = new ArrayList<>();
  }

  public void addBook(Book book) {
    inventory.add(book);
    System.out.println("Added book: " + book.title);
  }

  public ArrayList<Book> removeOutdatedBooks(int currentYear, int yearsThreshold) {
    ArrayList<Book> removedBooks = new ArrayList<>();
    Iterator<Book> iterator = inventory.iterator();
```

```
        while (iterator.hasNext()) {
            Book book = iterator.next();
            if (currentYear - book.getYear() > yearsThreshold) {
                removedBooks.add(book);
                iterator.remove();
                System.out.println("Removed outdated book: " + book.title);
            }
        }
        return removedBooks;
    }

    public double buyBook(String isbn, int quantity, String email, String address) {
        for (Book book : inventory) {
            if (book.getIsbn().equals(isbn)) {
                book.Selling(quantity, email, address);
                return book.price*quantity;
            }
        }
        throw new IllegalArgumentException("Book: " + isbn + " is not found!");
    }
}
```

MainTest:

```
package com.mycompany.quantum_bookstore;
import java.util.Scanner;

public class MainTest {

    public static void main(String[] args) {
```

```
BookStore store = new BookStore();


PaperBook paperBook1 = new PaperBook("1", "Java Book", "Mohamed", 2015, 100, 5);
PaperBook paperBook2 = new PaperBook("2", "Harry Potter", "JK Rolling", 2012, 150, 0);
EBook ebook1 = new EBook("3", "Data Structures And Algorithms Book", "Ashraf", 2022, 50, "PDF");
DemoBook demoBook = new DemoBook("4", "Learn English", "Yassin", 2000);


store.addBook(paperBook1);
store.addBook(paperBook2);
store.addBook(ebook1);
store.addBook(demoBook);


// Test 1: Normal Buying Book
System.out.println("\nTest 1: Normal Buying Book");
try {
   double total = store.buyBook("1", 2, "adel222@mail.com", "Cairo");
   System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
   System.out.println("Quantum book store error: " + e.getMessage());
}
System.out.println("========================================================");




// Test 2: Buy EBook
System.out.println("\nTest 2: Buy EBook");
try {
   double total = store.buyBook("3", 1, "anas@mail.com", "Alexandria");
   System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
   System.out.println("Quantum book store error: " + e.getMessage());
}
System.out.println("===================================================");



// Test 3: Buy Out of Stock Paper Book
System.out.println("\nTest 3: Out of Stock Paper Book");
try {
   double total = store.buyBook("2", 1, "mail@mail.com", "Sinai");
   System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
   System.out.println("Quantum book store error: " + e.getMessage());
}
System.out.println("===================================================");



// Test 4: Buy Demo Book
System.out.println("\nTest 4: Demo Book ");
try {
   double total = store.buyBook("4", 1, "nada@mail.com", "Cairo");
   System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
   System.out.println("Quantum book store error: " + e.getMessage());
}
System.out.println("===================================================");
```

```
// Test 5: Buy EBook with quantity > 1 (not allowed)
System.out.println("\nTest 5: EBook quantity > 1");
try {
   double total = store.buyBook("3", 2, "rana@mail.com", "Cairo");
   System.out.println("Quantum book store: Total paid: " + total);
} catch (Exception e) {
   System.out.println("Quantum book store error: " + e.getMessage());
}
System.out.println("=====================================================");



// Test 6: Remove Outdated Books (more than 10 years)
System.out.println("\nTest 6: Remove Outdated Books");
store.removeOutdatedBooks(2025, 10);
System.out.println("=====================================================");



// Test 7: User Input
System.out.println("\nTest 7: User Input");
Scanner scanner = new Scanner(System.in);
System.out.print("Enter your email: ");
String userEmail = scanner.nextLine();

System.out.print("Enter your address: ");
String userAddress = scanner.nextLine();
```

```
        System.out.print("Enter Book ISBN (ID): ");
        String BookISBN = scanner.nextLine();

        try {
            double total = store.buyBook(BookISBN, 1, userEmail, userAddress);
            System.out.println("Quantum book store: Total paid: " + total);
        } catch (Exception e) {
            System.out.println("Quantum book store error: " + e.getMessage());
        }
        System.out.println("======================================================");
    }
}
```