**Day 2**

1. **Banking Transaction implementation**

//account->account number, holder name,balance

//bank->holds account

//transaction->withdraw,deposit,transfer

//transaction check balance, account available

import java.util.\*;

public class Banking {

static class Account {

String accno;

String accname;

double balance;

public Account(String accno, String accname, double balance) {

this.accno = accno;

this.accname = accname;

this.balance = balance;

}

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

System.out.println("Deposited Rs." + amount);

System.out.println("Current Balance: Rs." + balance);

} else {

System.out.println("Deposit Failed. Amount must be positive.");

}

}

public void withdraw(double amt) {

if (amt > 0 && balance >= amt) {

balance -= amt;

System.out.println("Withdrawal of Rs." + amt + " success.");

System.out.println("Current Balance: " + balance);

} else if (amt <= 0) {

System.out.println("Enter only positive values.");

} else {

System.out.println("Insufficient Balance.");

}

}

public double getBal() {

return balance;

}

public String getAccinfo() {

return "Account Number: " + accno + ", Account Holder Name: " + accname + ", Balance: " + balance;

}

public String getAccno() {

return accno;

}

}

static class Bank {

public List<Account> accounts;

public Bank() {

this.accounts = new ArrayList<>();

}

public void addAccount(Account account) {

accounts.add(account);

System.out.println("Account Created Successfully");

}

public Account findAccount(String accno) {

for (Account account : accounts) {

if (account.getAccno().equals(accno)) {

return account;

}

}

return null;

}

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

Bank bank = new Bank();

int choice;

do {

System.out.println("\n====== Bank ====== \n 1. Create Account \n 2. Deposit \n 3. Withdraw \n 4. View Account \n 5. Exit \n Enter your Choice : ");

choice = sc.nextInt();

sc.nextLine();

switch (choice) {

case 1:

System.out.print("Enter Account Number: ");

String accNo = sc.nextLine();

System.out.print("Enter Account Holder Name: ");

String accName = sc.nextLine();

System.out.print("Enter Initial Balance: ");

double initBal = sc.nextDouble();

sc.nextLine();

Account acc = new Account(accNo, accName, initBal);

bank.addAccount(acc);

break;

case 2:

System.out.print("Enter Account Number: ");

String depAccNo = sc.nextLine();

Account depAcc = bank.findAccount(depAccNo);

if (depAcc != null) {

System.out.print("Enter amount to deposit: ");

double amount = sc.nextDouble();

sc.nextLine();

depAcc.deposit(amount);

} else {

System.out.println("Account not found.");

}

break;

case 3:

System.out.print("Enter Account Number: ");

String withAccNo = sc.nextLine();

Account withdrawAcc = bank.findAccount(withAccNo);

if (withdrawAcc != null) {

System.out.print("Enter amount to withdraw: ");

double amt = sc.nextDouble();

sc.nextLine();

withdrawAcc.withdraw(amt);

} else {

System.out.println("Account not found.");

}

break;

case 4:

System.out.print("Enter Account Number: ");

String infoAccNo = sc.nextLine();

Account infoAcc = bank.findAccount(infoAccNo);

if (infoAcc != null) {

System.out.println("Account Found!");

System.out.println(infoAcc.getAccinfo());

} else {

System.out.println("Account not found.");

}

break;

case 5:

System.out.println("\*\*\* Thanking you! \*\*\*");

break;

default:

System.out.println("Enter a valid option only ...");

}

} while (choice != 5);

sc.close();

}

}

Output:

====== Bank ======

1. Create Account

2. Deposit

3. Withdraw

4. View Account

5. Transaction

6. Exit

Enter your Choice :

1

Enter Account Number: 123

Enter Account Holder Name: shiva

Enter Initial Balance: 50000

Adding account to the Bank

====== Bank ======

1. Create Account

2. Deposit

3. Withdraw

4. View Account

5. Transaction

6. Exit

Enter your Choice :

1

Enter Account Number: 345

Enter Account Holder Name: balan

Enter Initial Balance: 45000

Adding account to the Bank

====== Bank ======

1. Create Account

2. Deposit

3. Withdraw

4. View Account

5. Transaction

6. Exit

Enter your Choice :

5

Enter Sender Account Number: 123

Enter Receiver Account Number: 345

Enter amount to transfer: 3500

Transfer of Rs. 3500.0 Success

Updated Balance of Receiver: Rs. 48500.0

====== Bank ======

1. Create Account

2. Deposit

3. Withdraw

4. View Account

5. Transaction

6. Exit

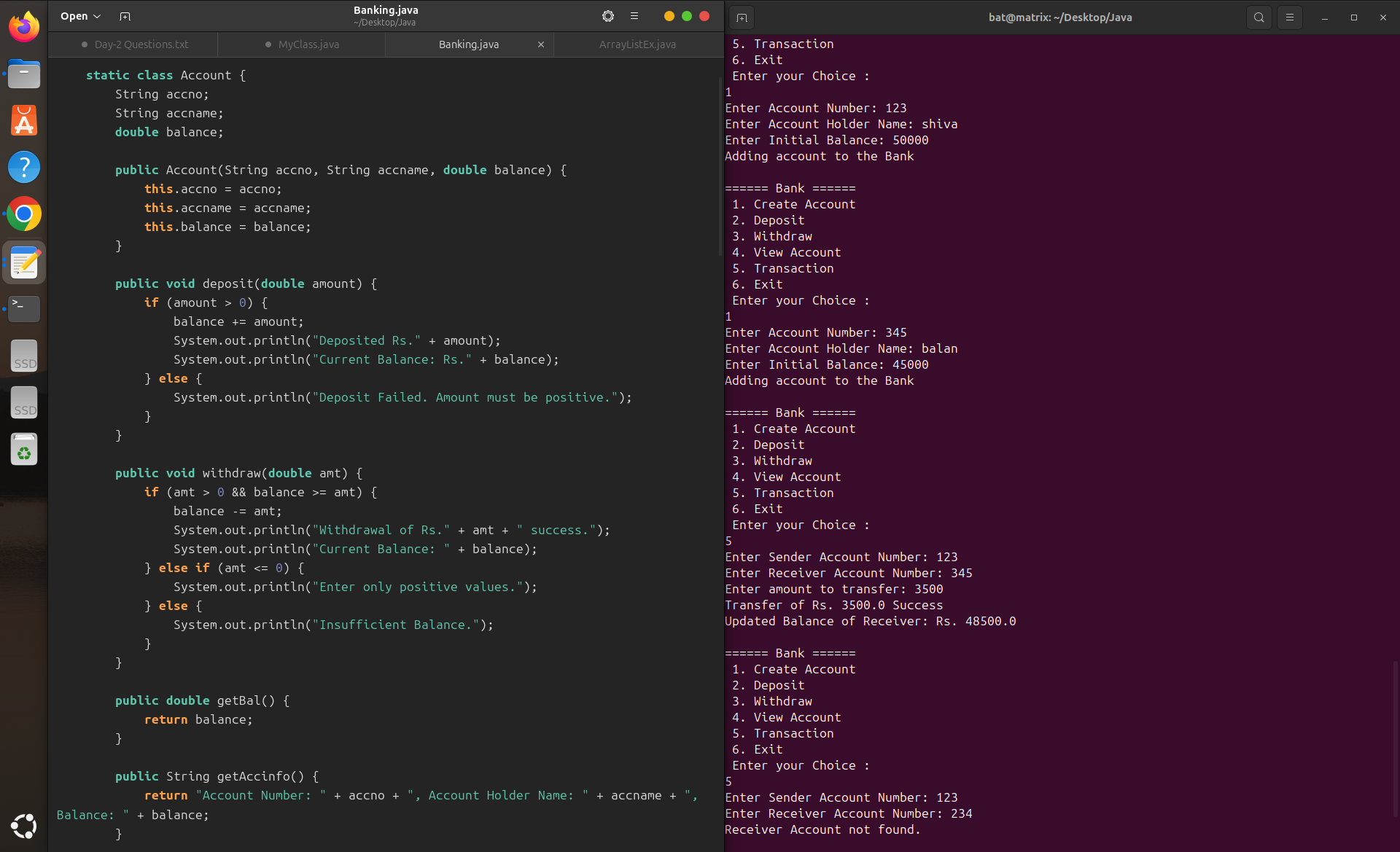
Enter your Choice :

5

Enter Sender Account Number: 123

Enter Receiver Account Number: 234

Receiver Account not found.



1. **Library Management:**

//library management

//lib-> isbn, book name, author

//transaction -> borrow, return

//borrow check for availability

import java.util.\*;

public class Library {

static class Book {

public String isbn;

public String booktitle;

public String author;

public boolean isBorrowed;

public Book(String isbn, String booktitle, String author) {

this.isbn = isbn;

this.booktitle = booktitle;

this.author = author;

this.isBorrowed = false;

}

public String getDetails() {

return "Book Title: " + booktitle + " | Author: " + author + " | ISBN: " + isbn + " | Status: " + (isBorrowed ? "Borrowed" : "Available");

}

}

static ArrayList<Book> books = new ArrayList<>();

public static void addBook(String isbn, String title, String author) {

books.add(new Book(isbn, title, author));

System.out.println("Book added successfully.");

}

public static void findBook(String isbn) {

for (Book book : books) {

if (book.isbn.equals(isbn)) {

System.out.println(book.getDetails());

return;

}

}

System.out.println("Book not found.");

}

public static void borrowBook(String isbn) {

for (Book book : books) {

if (book.isbn.equals(isbn)) {

if (!book.isBorrowed) {

book.isBorrowed = true;

System.out.println("Book borrowed: " + book.booktitle);

} else {

System.out.println("Book is already borrowed.");

}

return;

}

}

System.out.println("Book not found.");

}

public static void returnBook(String isbn) {

for (Book book : books) {

if (book.isbn.equals(isbn)) {

if (book.isBorrowed) {

book.isBorrowed = false;

System.out.println("Book returned: " + book.booktitle);

} else {

System.out.println("Book was not borrowed.");

}

return;

}

}

System.out.println("Book not found.");

}

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int choice;

do {

System.out.println("\n===== Library Management System ===== \n 1.Add Book \n2.Find Book \n3.Borrow Book \n4.Return Book \n5.Display All Books\n6.Exit\nEnter Your Choice : ");

choice = sc.nextInt();

sc.nextLine();

switch (choice) {

case 1:

System.out.print("Enter ISBN number: ");

String isbn = sc.nextLine();

System.out.print("Enter Book Title: ");

String booktitle = sc.nextLine();

System.out.print("Enter Book Author Name: ");

String author = sc.nextLine();

addBook(isbn, booktitle, author);

break;

case 2:

System.out.print("Enter ISBN to find book: ");

String findisbn = sc.nextLine();

findBook(findisbn);

break;

case 3:

System.out.print("Enter ISBN to borrow: ");

String borrowisbn = sc.nextLine();

borrowBook(borrowisbn);

break;

case 4:

System.out.print("Enter ISBN to return: ");

String returnisbn = sc.nextLine();

returnBook(returnisbn);

break;

case 5:

System.out.println("\nList of all books:");

for (Book b : books) {

System.out.println(b.getDetails());

}

break;

case 6:

System.out.println("\*\*\*Thanking You\*\*\*");

break;

default:

System.out.println("Enter Only Valid Options...");

}

} while (choice != 6);

sc.close();

}

}

Output:

===== Library Management System =====

1.Add Book

2.Find Book

3.Borrow Book

4.Return Book

5.Display All Books

6.Exit

Enter Your Choice :

1

Enter ISBN number: 1234

Enter Book Title: java

Enter Book Author Name: oracle

Book added successfully.

===== Library Management System =====

1.Add Book

2.Find Book

3.Borrow Book

4.Return Book

5.Display All Books

6.Exit

Enter Your Choice :

1

Enter ISBN number: 231

Enter Book Title: python

Enter Book Author Name: cython

Book added successfully.

===== Library Management System =====

1.Add Book

2.Find Book

3.Borrow Book

4.Return Book

5.Display All Books

6.Exit

Enter Your Choice :

2

Enter ISBN to find book: 321

Book not found.

===== Library Management System =====

1.Add Book

2.Find Book

3.Borrow Book

4.Return Book

5.Display All Books

6.Exit

Enter Your Choice :

3

Enter ISBN to borrow: 123

Book not found.

===== Library Management System =====

1.Add Book

2.Find Book

3.Borrow Book

4.Return Book

5.Display All Books

6.Exit

Enter Your Choice :

3

Enter ISBN to borrow: 1234

Book borrowed: java

===== Library Management System =====

1.Add Book

2.Find Book

3.Borrow Book

4.Return Book

5.Display All Books

6.Exit

Enter Your Choice :

5

List of all books:

Book Title: java | Author: oracle | ISBN: 1234 | Status: Borrowed

Book Title: python | Author: cython | ISBN: 231 | Status: Available

===== Library Management System =====

1.Add Book

2.Find Book

3.Borrow Book

4.Return Book

5.Display All Books

6.Exit

Enter Your Choice :

6

\*\*\*Thanking You\*\*\*

