


```

        System.out.println("_____");
        doAddChangeWorker();
        break;
    case 2:
        System.out.println("_____");
        doLookupAccount();
        break;
    case 3:
        System.out.println("_____");
        doRemoveAccount();
        break;
    case 4:
    case 5:
        doSave();
        break;
    default:
        System.out.println("*** Invalid choice "
            + choice
            + " - try again!");
    }
} catch (InputMismatchException ex) {
    System.out.println("*** Incorrect data entry - "
        + "enter an integer between 0 and "
        + (commands.length - 1));
    scIn.nextLine(); // Discard bad input.
    choice = -1;
}
} while (choice != commands.length - 1);
System.exit(0);
}

/**
 * Method to add or change an account. pre: The program exists and has been
 * loaded with data. post: A new name is added, or the value for the name is
 * changed, modified is set to true.
 */
private void doAddChangeAccount() {
    // Request the name.
    System.out.println("Enter name");
    String newName = scIn.nextLine();
    if (newName.equals("")) {
        return;
    }
    // Request the account number.
    System.out.println("Enter Account number");
    String newAccountNumber = scIn.nextLine();
    if (newAccountNumber.equals("")) {
        return;
    }
    // Request the account balance.
    System.out.println("Enter Account balance");
    String newAccountBalance = scIn.nextLine();
    if (newAccountBalance.equals("")) {
        return;
    }
    // Insert/change name-account numbe-account balance.
    String oldAccountNumber =
        theProgram.addOrChangeAccount(newName, newAccountNumber,
            newAccountBalance);

    String message;
    if (oldAccountNumber == null) { // New entry.
        message = newName + " was added to the program"
            + "\nNew Account number: " + newAccountNumber
            + "\nNew Balance: " + newAccountBalance + "$";
    } else { // Changed entry.
        message = "AccountNumber for " + newName + " was changed"
            + "\nOld Account number: " + oldAccountNumber

```

```
        + "\nNew number: " + newAccountNumber;
    }
    // Display confirmation message.
    System.out.println(message);
}

/**
 * Method to add or change a worker. pre: The program exists and has been
 * loaded with data. post: A new worker is added, or the value for the worker
 * name is changed, modified is set to true.
 */
private void doAddChangeWorker() {
    // Request the name.
    System.out.println("Enter Bank Worker name");
    String newName = scIn.nextLine();
    if (newName.equals("")) {
        return;
    }
    // Insert/change name.
    String oldWorkerName =
        theProgram.addOrChangeWorker(newName);
    String message;
    if (oldWorkerName == null) { // New entry.
        message = newName + " was added to the program\n"
            + "Welcome to new office!";
    } else { // Changed entry.
        message = "" + newName + " was changed"
            + oldWorkerName;
    }
    // Display confirmation message.
    System.out.println(message);
}

/**
 * Method to look up a account. pre: The program has been loaded with data.
 * post: No changes made to the program data base.
 */
private void doLookupAccount() {
    // Request the account number.
    System.out.println("Enter Account number:");
    String theAccountNumber = scIn.nextLine();
    if (theAccountNumber.equals("")) {
        return; // Dialog was cancelled.
    }
    // Look up the account.
    String theAccountBalance = theProgram.lookupAccount(theAccountNumber);
    String message;
    if (theAccountBalance != null) { // Account was found.
        message = "The account for " + theAccountNumber + " has money $"
            + theAccountBalance;
    } else { // Account was not found.
        message = theAccountNumber + " is not listed in the program";
    }
    // Display the result.
    System.out.println(message);
}

/**
 * Method to look up a worker. pre: The program has been loaded with data.
 * post: No changes made to the program data base.
 */
private void doLookupWorker() {
    // Request the bank worker name.
    System.out.println("Enter Bank Worker name:");
    String theBankWorkerName = scIn.nextLine();
    if (theBankWorkerName.equals("")) {
        return; // Dialog was cancelled.
    }
}
```

```
    }
    // Look up the bank worker.
    String control = theProgram.lookupBankWorker(theBankWorkerName);
    String message;
    if (control != null) { // Account was found.
        message = "" + theBankWorkerName + " still works here!";
    } else { // Account was not found.
        message = theBankWorkerName + " does not work here!";
    }
    // Display the result.
    System.out.println(message);
}

/**
 * Method to remove a account Pre: The program has been loaded with data.
 * Post: The requested account number is removed, modified is set true
 */
private void doRemoveAccount() {
    // Request the account number.
    System.out.println("Enter account number:");
    String theAccountNumber = scIn.nextLine();
    if (theAccountNumber.equals("")) {
        return;
    }
    // Remove the account
    if (theProgram.removeAccount(theAccountNumber) != null)
    {
        // Display confirmaion message
        System.out.println(theAccountNumber + " has been removed from"
            + " the program!");
    }
    else
        System.out.println(theAccountNumber + " not found!");
}

/**
 * Method to save the program to the data file. pre: The program has
 * been loaded with data. post: The current contents of the program have
 * been saved to the data file.
 */
private void doSave() {
    theProgram.save();
}
}
```