

# 1B1a

## PhoneBook

### Example

1 Copyright © 2003, Graham Roberts

Department of Computer Science



## Agenda

- Designing a simple One-Class program.

2 Copyright © 2003, Graham Roberts

Department of Computer Science



## Specification

- Write a program to store a list of names + phone numbers.
- Should be able to add and remove entries.
- Should be able to search by name.
- Should be able to load/save to file.

3 Copyright © 2003, Graham Roberts

Department of Computer Science



## Overall Behaviour

```
while (user has not quit)
{
  display a list of actions (menu)
  select an action
  do action
}
```

4 Copyright © 2003, Graham Roberts

Department of Computer Science



## Actions?

- Add – input name and phone number
- Remove – input name, find entry and remove name+number.
- Search – input name, find entry, display result.
- Load – load list of entries from file and add to current list.
- Save – store all current entries in a file.

5 Copyright © 2003, Graham Roberts

Department of Computer Science



## Entry?

- How are names and phone numbers represented when program is run?
  - Choose String as representation.
  - Need a collection (list of).
  - Use ArrayList
    - An ArrayList of names
    - An ArrayList of numbers

6 Copyright © 2003, Graham Roberts

Department of Computer Science



## Filling in details

```
public class OneClassPhoneBook
{
    private KeyboardInput in = new KeyboardInput();
    private ArrayList names = new ArrayList();
    private ArrayList numbers = new ArrayList();

    public void go()
    {
        boolean quit = false;
        while (!quit)
        {
            displayMenu();
            int item = in.readInteger();
            quit = doAction(item);
        }
    }
}
```

Top-down  
Main Loop uses displayMenu  
and doAction

7 Copyright © 2003, Graham Roberts

Department of Computer Science



```
private void displayMenu()
{
    System.out.println("\nPhone Book");
    System.out.println("1. Search for a person's phone number");
    System.out.println("2. Add a new person and phone number");
    System.out.println("3. Remove a person and phone number");
    System.out.println("4. Save phone list to a file");
    System.out.println("5. Add a phone list from a file");
    System.out.println("6. Quit");
    System.out.print("\nSelect an item from the menu: ");
}
}
```

8 Copyright © 2003, Graham Roberts

Department of Computer Science



```
private boolean doAction(int item)
{
    switch (item)
    {
        case 1 :
            searchForEntry(); break;
        case 2 :
            addEntry(); break;
        case 3 :
            removeEntry(); break;
        case 4 :
            saveList(); break;
        case 5 :
            loadList(); break;
        case 6 :
            return true; default :
                System.out.println("\nSorry - don't recognise that item, try again");
    }
    return false;
}
```

9 Copyright © 2003, Graham Roberts

Department of Computer Science



## Stub Methods to compile

```
Public searchForEntry()
{
    // Leave method body empty
}
```

10 Copyright © 2003, Graham Roberts

Department of Computer Science



```
public void addEntry()
{
    System.out.println("\nAdd entry to phone list");
    System.out.print("Enter the person's name: ");
    String name = in.readString();
    System.out.print("Enter the person's phone number: ");
    String phone = in.readString();
    names.add(name);
    numbers.add(phone);
}
```

11 Copyright © 2003, Graham Roberts

Department of Computer Science



```
public void removeEntry()
{
    System.out.println("\nRemove entry from phone list");
    System.out.print("Enter the person's name: ");
    String name = in.readString();
    remove(name);
}
```

12 Copyright © 2003, Graham Roberts

Department of Computer Science



```

private void remove(String name)
{
    for (int i = 0; i < names.size(); i++)
    {
        String entry = (String) names.get(i);
        if (entry.equals(name))
        {
            names.remove(i);
            numbers.remove(i);
            return;
        }
    }
    System.out.println("Name not found - nothing removed");
}

```

13 Copyright © 2003, Graham Roberts

Department of Computer Science



```

public void saveList()
{
    System.out.println("\nSave phone list to a file");
    System.out.print("Enter the file name: ");
    String fileName = in.readString();
    FileOutputStream out = new FileOutputStream(fileName);
    writeToFile(out);
    out.close();
}

```

14 Copyright © 2003, Graham Roberts

Department of Computer Science



```

private void writeToFile(FileOutputStream file)
{
    for (int i = 0; i < names.size(); i++)
    {
        file.writeString((String) names.get(i));
        file.writeNewline();
        file.writeString((String) numbers.get(i));
        file.writeNewline();
    }
}

```

15 Copyright © 2003, Graham Roberts

Department of Computer Science



```

public void loadList()
{
    System.out.println("\nAppend phone list from a file");
    System.out.print("Enter the file name: ");
    String fileName = in.readString();
    FileInputStream fileIn = new FileInputStream(fileName);
    readFromFile(fileIn);
    fileIn.close();
}

```

16 Copyright © 2003, Graham Roberts

Department of Computer Science



```

private void readFromFile(FileInputStream file)
{
    String name = file.readString();
    String phone = file.readString();
    while (!file.eof())
    {
        names.add(name);
        numbers.add(phone);
        name = file.readString();
        phone = file.readString();
    }
}

```

17 Copyright © 2003, Graham Roberts

Department of Computer Science



```

public static void main(String[] args)
{
    OneClassPhoneBook phoneBook = new OneClassPhoneBook();
    phoneBook.go();
}

```

18 Copyright © 2003, Graham Roberts

Department of Computer Science



## Summary

---

- Top-down design.
- Step-by-step implementation.

