

Today's topics

Java

Information Retrieval

Upcoming

Intellectual Property (Prof Forbes)

Reading

***Great Ideas*, Chapter 4**

Information Retrieval

- Often want to use program for information storage and retrieval
- On line phone book is good example
- Using “Parallel” or “Corresponding Arrays”
 - Array (of Strings) for Names
 - Array (of Strings) for Phone Numbers

Name	Number
“J.Able”	“613-1978”
“D.Ramm”	“660-6532”
“D.Ramm”	“732-7616”
“R.Odom”	“681-6326”
“M.Salter”	“684-8111”
“W.Tars”	“613-1978”

Phone Lookup Program

```
public class LookUp extends java.applet.Applet implements
                                   ActionListener
{
    TextField gName, gNum, mInstruct;
    String Name[]={"J.Able", "D.Ramm", "D.Ramm", "R.Odom",
                  "M.Salter", "W.Tars"};
    String Num[]= {"613-1978","660-6532","732-7616",
                  "681-6326","684-8111","613-1978"};

    Button bByName, bByNum;
    TextArea mResults;
    String name, num;
    int size;

    public void init()
    {
        mInstruct = new TextField(60);
        mInstruct.setText(
            "Enter search info; push related button");
    }
}
```

Phone Lookup Program.2

```
gName = new TextField(25);
gNum = new TextField(25);
bByName = new Button("ByName");
bByNum = new Button("ByNum");
mResults = new TextArea(10, 60);
bByName.addActionListener(this);
bByNum.addActionListener(this);
add(mInstruct); add(gName); add(bByName); add(gNum);
add(bByNum); add(mResults);
size = 6;
}

public void actionPerformed(ActionEvent event)
{
    Object cause = event.getSource();
    int k=0;
```

Phone Lookup Program.3

```
    if (cause == bByName)
    { name = gName.getText();
      while (k < size)
      { if (name.equals(Name[k]))
          mResults.append("Name: "+name+" shows number: "
                          + Num[k]+"\\n");

          k = k + 1;
        }
      }
    if (cause == bByNum)
    { num = gNum.getText();
      while (k < size)
      { if (num.equals(Num[k]))
          mResults.append("Number: "+num+" shows name: "
                          + Name[k]+"\\n");

          k = k + 1;
        }
      }
    }
  }
}
CPS 001
```

Search Using a Single Array

- **Put all relevant info about one phone in one string**
 - **Need additional String method**
- **indexOf method**

```
int indexOf(String key)
```

 - **Searches for first occurrence of key with object string**
 - **Returns position: is ≥ 0**
- **Resulting program much more flexible**

Search Program

```
public class Search extends java.applet.Applet implements
                                   ActionListener
{
    TextField gKey, mInstruct;
    String Info[]={ "J.Able, 613-1978", "D.Ramm, 660-6532",
                    "D.Ramm, 732-7616", "R.Odom, 681-6326",
                    "M.Salter, 684-8111", "W.Tars, 613-1978"};

    Button bSearch;
    TextArea mResults;
    String key;
    int size = 6;

    public void init()
    {
        mInstruct = new TextField(60);
        mInstruct.setText(
            "Enter search info; push related button");
        gKey = new TextField(35);
        bSearch = new Button("Search");
    }
}
```

Search Program.2

```
mResults = new TextArea(10, 60);
bSearch.addActionListener(this);
add(mInstruct); add(gKey); add(bSearch); add(mResults);
}
```

```
public void actionPerformed(ActionEvent event)
{ Object cause = event.getSource();
  int k=0;
  if (cause == bSearch)
  { key = gKey.getText();
    while (k < size)
    { if (Info[k].indexOf(key) >= 0)
      mResults.append(Info[k]+"\\n");
      k = k + 1;
    }
  }
}
```

Database

- Extend ideas used in phone program
- Can have many “parallel” arrays
- Design Used Car Database

Make	Style	Color	Owner	Year
“Mercedes”	“4 door”	“white”	“M.Ramm”	1987
“Mercedes”	“4 door”	“ivory”	“D.Ramm”	1987
“Ford”	“truck”	“blue”	“M.Ramm”	1972
“Mercedes”	“4 door”	“green”	“D.Ramm”	1971
“Datsun”	“2 door”	“red”	“K.Ramm”	1978
“Ford”	“4 door”	“blue”	“D.Ramm”	1978