Today’s topics

Revisiting numbers & text
Methods
Loops
Arrays

Reading

Great Ideas, Chapter 4

Types for Numbers

- The type String is not a built-in type, technically it’s a class

- There are many numerical types in Java. We’ll use two
  - int, represents integers: [...-3,-2,-1,0,1,2,3,...]
    - Conceptually there are an infinite number of integers, but the range is limited to \([-2^{31}, 2^{31}-1]\) or \([\text{Integer.MIN_VALUE}, \text{Integer.MAX_VALUE}]\)
    - Alternatives? Why is range limited?

- double, represents real numbers like \(\pi, \sqrt{2}\)
  - Not represented exactly, so expressions like \(100*0.1\) may yield unexpected results
  - Double precision floating point numbers, another type float exists, but it’s a terrible choice (generates poor results)

GIGO: program as good as its data?

- In calculations involving floating point numbers it’s easy to generate errors because of accumulated approximations:
  - What is \(10^{23} + 1\)?
  - When is \((x + y) + z\) different from \(x + (y + z)\) ?

- The type int is severely constrained on 16-bit computers, e.g., running DOS, largest value is 32,767 \((2^{16} - 1)\)
  - Even on 32-bit machines, how many seconds in a millennium? \(60*60*24*365*1000\), problems?
  - On UNIX machines time is measure in seconds since 1970, problems?
  - What was Y2K all about?

Manipulating Strings

- Methods for manipulation
  - int length()
  - int indexOf(String st)
  - String substring(int start, int end)

- Getting String Data from user
  - The TextField class has getText() method
  - Use:
    - message = mg.getText();
      - where mg is a TextField and message is a String
More expressions

```java
int n = 1 - 2 * 3 - 4 + 5;
```

What is `n`?
1. 4
2. -2
3. 0
4. 2
5. 4
6. error

```java
int n = 12 + "hello"
```

1. 0
2. 12
3. 17
4. unknown
5. error

```java
int x = 8 * (7 - 6 + 5) % (54 + 3 / 2) - 1;
```

What is `x`?
1. -1
2. 0
3. 2
4. 3
5. error
6. something else

Repeating code

- Repeating code is bad
- Writing repetitive code is tedious
- Debugging repetitive code is hard
- Avoid repeating code through:
  - Subroutines/methods
  - Loops

Loops

- If statements need to repeat, then you probably need a loop
- Describe portion of program as:
  - Repeat
  - Continue until
  - For each value from 1 to `n`
  - For every object of a set, do something
- We have already used iteration by using the buttons
  - How?

Problems

- We want to:
  - Print out all numbers from 0 up to 100 incrementing by 0.5 each time
  - Sum up the numbers from 1 to 100
  - ...

- New Java syntax
  - New object type `TextArea` which is basically a big scrolling textbox
    - `tArea` is 80 character wide and 20 rows high text box with 20 rows
      ```java
      TextArea tArea = new TextArea(20,80);
      ```
    - Add characters to the end of the `TextArea` using `append`
      ```java
      tArea.append("Hello\n");
      ```
    - `\n` is called a newline character which moves the next character to the next line
Anatomy of a while loop

- While loops are one way to get rid of repetitive code
- Print out numbers up to 100 by increments of 0.5

```python
x = 0.0;
while (x < 100)
    x = x + 0.5;
    print x;
```

Another loop

- Summing the numbers 1 ... 100
  ```python
  sum = 0;
k = 0;
while (k < 100)
    k = k + 1;
    sum = sum + 1;
  
Does this code do the right thing?

Other Loop designs
  - Count down
  - Stopping and starting at computed values
  - Data dependent loop

Functions/Methods

- Function example: distance from point (x,y) to origin
- Function declaration
  - Name of the function
  - Type of each argument to the function with a descriptive name for each argument
  - The type of value a function returns

Function calling mechanics

- The value of each argument are computed
- The value of each argument is copied into the corresponding formal parameter
- The statements in the function body are evaluated until a return statement appears
- The value of the return expression is evaluated
- The calling program continues, with the returned value substituted in place of the call
Functions can return strings

```java
String weekDay(int day) {
    if (0 == day)  {   return "Sunday";
    }
    else if (1 == day)  {   return "Monday";
    }
    else if (2 == day)  {   return "Tuesday";
    }
    else if (3 == day)  {   return "Wednesday";
    }
    // ...
}
```

- What function call looks like?
  ```java
  String dayName;
  int dayNum = 4;
  dayName = weekDay(dayNum);
  ```

- Which is/are ok? Why?
  ```java
  result.setText(weekDay(5));
  int j = weekDay(0);
  result.setText(weekDay(2.1));
  String s = weekDay(22);
  ```

Using arrays

- **subscript or index** to access element
  ```java
  x[5] = 20;
  foo.setText("Result is " + x[5]);
  ```

- Often used in loops
  ```java
  int k = 0; sum = 0;
  while ( k < 10 ) {
      k = k + 1;
      sum = sum + name[k];
  }
  ```

Creating Arrays

- Declaration
  ```java
  double weights[];
  ```

- Definition
  ```java
  weights = new double[50];
  ```

- Combine
  ```java
  double weights[] = new double[50];
  ```

```java
int num[] = new int[6];
```
Arrays & Loops

```c
int k = 2;
while(k<6)
{
    num[k] = k*k;
    k = k+1;
}
```