Final Review

Dec 13

Strings

methods on strings

- about 67 methods defined for String and about 12 constructors
  - charAt
  - length
  - indexOf
  - equals
  - compareTo
  - substring
  - split

Arrays of Objects

- String[] arrayOfStrings = {"hello", "this", "is", "a", "test"};
- arrayOfStrings[oo] .equals("hello"); // true

```
public class Stringer {
    public static void main(String[] args) {
        String[] arrayOfStrings = {"hello", "this", "is", "a", "test"};
        for (int i = 0; i < arrayOfStrings.length; i++) {
            System.out.print(arrayOfStrings[i].charAt(0));
        }
        System.out.println();
        System.out.println();
        for (int i = 0; i < arrayOfStrings.length; i++) {
            for (int j = 0; j < arrayOfStrings.length; j++) {
                System.out.print(arrayOfStrings[i].compareTo(arrayOfStrings[j]) + "  ");
            }
            System.out.println();
        }
    }
}
```

Null and NullPointerException

- Any object -- anywhere and be assigned to "null"
  - the default value of an object
    - except for "definite assignment"

```
public static void main2(String[] args) {
    String aa = null;
    System.out.println(aa.charAt(0));
}
```

Creating Objects

- Constructors
- "private"variables
- accessor Methods
  - get...
  - set...
- toString

```
public class Charity5 {
    private String name;
    private int donationTarget;
    private int donationsReceived;

    public Charity5(String nm, int dt, int dr) {
        this.name = nm;
        this.donationTarget = dt;
        this.donationsReceived = dr;
    }

    public String getName() {
        return name;
    }

    public int getDonationTarget() {
        return donationTarget;
    }

    public int getDonationsReceived() {
        return donationsReceived;
    }

    public void adjustDonationsReceived(int dr) {
        donationsReceived += dr;
    }

    public double percentageOfGoal() {
        return (double) (donationsReceived * 100) / donationTarget;
    }

    public String toString() {
        return name + " has a donation target of " + donationTarget + ". It has received "
                + " which is " + percentageOfGoal() + " of its goal.";
    }
}
```

Selection Sort

find, swap, repeat

```
public static void selectionSort(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        int best = 0;
        for (int j = 1; j < (arr.length - i); j++) {
            if (arr[best] < arr[j]) {
                best = j;
            }
        }
        int temp = arr[best];
        arr[best] = arr[arr.length - i - 1];
        arr[arr.length - i - 1] = temp;
    }
}
```

The Last Activity

0	6
1	6
2	107
3	79
4	104
5	132
6	148
7	37
8	96
9	89
10	15
11	74
12	71
13	109
14	115

Show the array at the end of each "i loop"

```
public static void selectionSort(int[] arr) {
    for (int i = 0; i < arr.length; i++) {
        int best = 0;
        for (int j = 1; j < (arr.length - i); j++) {
            if (arr[best] > arr[j]) {
                best = j;
            }
        }
        int temp = arr[best];
        arr[best] = arr[arr.length - i - 1];
        arr[arr.length - i - 1] = temp;
    }
}
```

Search

- Linear
  - Always works
- Binary
  - Only on sorted lists

Efficiency

- Big O notation
- Selection sort == O(n*n)
- Linear search == O(n)
- BinarySearch == O(lg(n))

- partitioning (from hw9) ==

More making Objects

- Using objects within objects
- Overriding the equals method
- Inheritance
  - public class EuropeanDate extends Date