Garfield AP CS

String methods

Most content from Stuart Reges and Marty Stepp at University of Washington. Thanks!!

Boolean practice

- Write a method isRich that takes two parameters: savings and salary. It should return according rules:
  - Salary > 200,000 -> true
  - Savings > 1,000,000 -> true
  - Savings > 100,000 and salary >50,000 -> true
  - Otherwise -> false

Finding prime numbers

- Write pseudocode for a method that takes one integer parameter and returns a boolean based on whether or not it is prime
- What was wrong with the one from the worksheet on Friday?

Recall: Objects and classes

- **object:** An entity that contains:
  - *data* (variables), and
  - *behavior* (methods).

- **class**: A program, or a type of objects.

- Examples:
  - The class `String` represents objects that store text.
  - The class `DrawingPanel` represents graphical window objects.
  - The class `Scanner` represents objects that read information from the keyboard, files, and other sources.

Recall: Strings

- **string**: An object storing a sequence of text characters.
  - Unlike most other objects, a `String` is not created with `new`.

    ```
    String name = "text";
    String name = expression;
    ```

  - Examples:

    ```
    String name = "Marla Singer";
    int x = 3;
    int y = 5;
    String point = "(" + x + ", " + y + ")";
    ```

Inner workings of Strings

- Characters of a string are numbered with 0-based *indexes*:

```
String name = "P. Diddy";
```

index	0	1	2	3	4	5	6	7
char	P	.		D	i	d	d	y

  - The first character's index is always 0
  - The last character's index is 1 less than the string's length

  - The individual characters are values of type `char`

String methods

Method name	Description
indexOf(**str**)	index where the start of the given string appears in this string (-1 if it is not there)
length()	number of characters in this string

substring(**index1**, **index2**)
or
substring(**index1**)
	the characters in this string from *index1* (inclusive) to *index2* (exclusive);
	if *index2* omitted, grabs till end of string

toLowerCase()	a new string with all lowercase letters
toUpperCase()	a new string with all uppercase letters

- These methods are called using the dot notation:

```
String gangsta = "Dr. Dre";
System.out.println(gangsta.length());   // 7
```

Modifying strings

- Methods like `substring`, `toLowerCase`, etc. create/return a new string, rather than modifying the current string.

```
String s = "lil bow wow";
s.toUpperCase();
System.out.println(s);    // lil bow wow
```

- To modify a variable, you must reassign it:

```
String s = "lil bow wow";
s = s.toUpperCase();
System.out.println(s);    // LIL BOW WOW
```

Strings as user input

- `Scanner`'s `next` method reads a word of input as a `String`.

```
Scanner console = new Scanner(System.in);
System.out.print("What is your name? ");
String name = console.next();
name = name.toUpperCase();
System.out.println(name + " has " + name.length() +
    " letters and starts with " + name.substring(0, 1));
```

Output:

```
What is your name? Madonna
MADONNA has 7 letters and starts with M
```

- The `nextLine` method reads a line of input as a `String`.

```
System.out.print("What is your address? ");
String address = console.nextLine();
```