Recitation 6

Enums and key Collections data structures

How do we represent . . .

- Suits - Clubs, Spades, Diamonds, Hearts
- Directions - North, South, East, West
- Days of week - Monday, Tuesday . . .
- Planets - Mercury, Venus, Earth . . .

Other small sets of values that do not change

Using constants

```java
public class Suit {
    public static final int CLUBS = 0;
    public static final int SPADES = 1;
    public static final int DIAMONDS = 2;
    public static final int HEARTS = 3;
}
```

Problems:
- no type checking
- readability

Objects as constants

```java
public class Suit {
    public static final Suit CLUBS = new Suit();
    public static final Suit SPADES = new Suit();
    public static final Suit DIAMONDS = new Suit();
    public static final Suit HEARTS = new Suit();
    private Suit() {}  
}
```

- no new Suits can be created
- cannot modify Suit objects

Suit v; … if (v == Suit.CLUBS) { … }

Enum declaration

- could be any access modifier
- new keyword
- static final variables of enum Suit

About enums

1. Can contain methods, fields, constructors
   a. Suit.HEARTS.getColor();
2. Suit’s constructor is private!
   a. Cannot instantiate except for initial constants
3. Suit.values() returns Suit[] of constants in enum
Demo: Enums in action

Create a class PlayingCard and class Deck. What would be the fields for a PlayingCard object?

- Enums
- Enum odds and ends

1. Suit is a subclass of java.lang.Enum
2. ordinal() returns position in list (i.e. the order it was declared)
   - Suit.CLUBS.ordinal() == 0
3. Enums automatically implement Comparable
   - Suit.CLUBS.compareTo(Suit.HEARTS) uses the ordinals for Clubs and Hearts
4. toString(Suit.CLUBS) is "CLUBS"
   - you can override this!

Collections and Map

Power of inheritance and interfaces

- Object
- AbstractCollection<E>
- AbstractList<E>
- ArrayList<E>

Format of ArrayList object

Important interfaces

- Collection<E>
  - add(E)
  - contains(Object)
  - isEmpty()
  - remove(Object)
  - size()
  - ...

- List<E>
  - get(int)
  - indexOf(int)
  - add(int, E)
  - ...

- Map<K,V>
  - put(K,V)
  - get(Object)

- Set<E>
  - No new methods in Set<E>, just changes specifications
Important classes

Collections and Map

```
Collection<E>

Set<E>

List<E>

HashSet<E>

ArrayList<E>

Map<K,V>

HashMap<K,V>

Queues? Stacks?

```

Collections problems

1. Remove duplicates from an array
2. Find all negative numbers in array
3. Create ransom note
4. Implement a Stack with a max API
5. Print a binary tree in level-order
6. Braces parsing

Collections problems

Complete

```
Integer[] removeDuplicates(int[])
```

Remove all duplicates from an array of integers.

```
Very useful HashSet method:
hs.toArray(new Integer[hs.size()]);
```

Collections problems

Find Negative Numbers
Find all negative numbers in array and return an array with those integers

```
Very useful ArrayList method:
lst.toArray(new Integer[lst.size()]);
```

Collections problems

Create Ransom Note
Given a note (String) that you would like to create and a magazine (String), return whether you can create your note from the magazine letters.

```
ungive me the money or java gets it.
```
Implement a Stack\langle E \rangle with a max() function in O(1) time

No matter how full the stack is, the max function should be in constant time. (i.e., you should not iterate through the Linked List to find the maximum element)

Print a binary tree in level-order

Output: 1 2 3 4 5 6

Challenge Problem
Output:
1
2 3
4 5 6

Braces parsing in O(n) time

Return whether a String has the right format of square brackets and parenthesis.

e.g.,
"array[4] = { { new Integer(3) })} ); " <- is true
"( ) [ ] " <- is false
" })(" <- is false
" [ ] ] " <- is false