

## Objectives

- Java wrap-up
  - Enumerated Types
  - Comparators
  - Resolving overloaded methods
  - Standard Error
- Language Comparison

Oct 12, 2011

Sprenkle - CSCI209

1

## Review

- What is the Collections Framework made up of?
- What are the three ADT interfaces we discussed?
  - What is an implementation of each of them
- How can we iterate over a collection?
- What are some benefits of the Collections Framework?

Oct 12, 2011

Sprenkle - CSCI209

2

## ENUMERATED TYPES

Oct 12, 2011

Sprenkle - CSCI209

3

## Enumerated Types

Type whose legal values consist of a fixed set of constants

- Also called *enums*
  - More powerful than enums in C
  - Added Java 1.5

Oct 12, 2011

Sprenkle - CSCI209

4

## Int Enum Pattern

```
public static final int APPLE_FUJI      = 0;
public static final int APPLE_EMPIRE   = 1;
public static final int APPLE_GRANNY_SMITH = 2;

public static final int ORANGE_NAVEL    = 0;
public static final int ORANGE_TEMPLE   = 1;
public static final int ORANGE_BLOOD    = 2;
```

- What is this code trying to do/set up?
- Any drawbacks?

Oct 12, 2011

Sprenkle - CSCI209

5

## Int Enum Pattern Discussion

```
public static final int APPLE_FUJI      = 0;
public static final int APPLE_EMPIRE   = 1;
public static final int APPLE_GRANNY_SMITH = 2;

public static final int ORANGE_NAVEL    = 0;
public static final int ORANGE_TEMPLE   = 1;
public static final int ORANGE_BLOOD    = 2;
```

- Drawbacks
  - No type safety (ORANGE vs APPLE?)
  - Compile-time constants
    - Change associated int, other code needs to be recompiled
  - Weak debug information
  - Can't iterate over them reliably; size of group?
- Similar: *String enum pattern*

Oct 12, 2011

Sprenkle - CSCI209

6

## Enum

```
public enum Apple {FUJI, EMPIRE, GRANNY_SMITH};
public enum Orange {NAVEL, TEMPLE, BLOOD};
```

Use:

```
Apple lunch = Apple.FUJI;
```

Each is a public static final instance

- Full-fledged class
  - Can add arbitrary methods and fields
  - Implementations of Object methods, Comparable interface, ...
  - Effectively final

Oct 12, 2011

Sprenkle - CSCI209

Fruit.java

7

## Enumerated Types

- Are like **inner** classes in Java
  - Entirely nested within another class
- Implicitly inherits from `java.lang.Enum`
  - `boolean equals(Object other)`
  - `int compareTo(E o)`
  - `String name()`
    - Returns the name of this enum constant, exactly as declared in its enum declaration
  - `int ordinal()`
    - Returns the ordinal of this enumeration constant, i.e., its position in its enum declaration, where the initial constant is assigned an ordinal of zero

Oct 12, 2011

Sprenkle - CSCI209

8

## Enums

- Has static `values()` method
  - Returns *array* of values in order declared
  - E.g., `FUJI, EMPIRE, GRANNY_SMITH`
- Can be used in `switch` statements

```
switch(lunch) {
    case Apple.FUJI:
        price = 1.43;
    ...
}
```

Oct 12, 2011

Sprenkle - CSCI209

9

## Designing the Playing Card Class

- State?
  - How to represent?
- API?

Implement:

```
boolean sameSuit(Card c)
int getRummyValue()
```

Oct 12, 2011

Sprenkle - CSCI209

10

## Enum Summary

- An inner class
- Inherit implicitly from `java.lang.Enum`
- Can iterate over
- Use when have fixed set of legal values
- More sophisticated things you can do with them

Oct 12, 2011

Sprenkle - CSCI209

11

## COMPARATORS

Oct 12, 2011

Sprenkle - CSCI209

12

## Alternative Sorting

- What if object is **Comparable** but does not sort the way you want?
  - **Special case**
    - Don't want to change class
    - Don't have access to class
  - **Example: want to sort strings so capital and lowercase letters are the same**
- Use **Comparator** interface

Oct 12, 2011

Sprenkle - CSCI209

13

## Comparator<T> Interface

- Declares two methods:
  - `int compare(T o1, T o2)`
    - Compare two objects and return a value as if we called `o1.compareTo(o2)`
  - `boolean equals(Object other)` ← Have default from Object
    - Check if this Comparator equals other
- Overloaded versions of **sort** in Arrays and Collections
  - Arrays: `void sort(Object[] array, Comparator c)`
  - Collections: `void sort(List list, Comparator c)` `EmployeeNameComparator.java`

Oct 12, 2011

Sprenkle - CSCI209

14

## RESOLVING OVERLOADED METHODS

Oct 12, 2011

Sprenkle - CSCI209

15

## Discussion of OverloadPlay.java

Oct 12, 2011

Sprenkle - CSCI209

16

## Resolving Methods

- Java uses dynamic dispatch on the object the method is called on
- For overloaded methods, Java compiler **statically** determines the types of the method arguments to select the best (most-specific) match

Oct 12, 2011

Sprenkle - CSCI209

17

## Discussion of Bigram.java

Oct 12, 2011

Sprenkle - CSCI209

18

## Exam Review

- Grading
  - 25%: Two Exams
  - 45%: Programming assignments
  - 5%: Testing project
  - 20%: Final Team project
  - 5%: Grading, participation
- Part A: Automatic
- Part B: Communicating ideas with appropriate terminology
- Part C: Coding (similar to Assignment 6)
- Average, Median: 84%

Oct 12, 2011

Sprenkle - CSCI209

19

## TO DO

- Assignment 8: Collection, More User Interface practice
- When we return: Software development best practices

Oct 12, 2011

Sprenkle - CSCI209

20