Enumerations

▸ There are cases when a variable can only take one of a small set of predefined constant values, e.g., compass direction (N, S, E, W) and the days of a week (MON, TUE, etc.)

▸ In such cases, you should use an `enum` type to define a set of constants represented as unique identifiers

```
public enum Direction {
    NORTH, SOUTH, EAST, WEST
}
```

Enumerations

- `Direction` is a type called an **enumeration**, which is a special kind of class introduced by the keyword `enum` and a type name

- Inside the braces {} is a comma-separated list of enumeration constants, each representing a unique value (you don't need to care about the underlying implementation or the exact values)

- The identifiers in an `enum` must be unique

```
public enum Direction {
    NORTH, SOUTH, EAST, WEST
}
```

Enumerations

- Variables of the type `Direction` can be assigned only the four constants declared in the enumeration (other values are illegal, won't compile)
  - `Direction d = Direction.NORTH;`

- Like classes, all `enum` types are reference types

```
public enum Direction {
    NORTH, SOUTH, EAST, WEST
}
```

Enumerations

▸ Each `enum` declaration declares an `enum` class with the following restrictions:

- `enum` constants are implicitly `final` (constants that shouldn't be modified)

- `enum` constants are implicitly `static` (no objects need to access them)

- Any attempt to create an object of an `enum` type with operator `new` results in a compilation error (constructor of an `enum` type can only be private or private-private, meaning without any access level modifier)

- `enum` declarations contain two parts: (1) the `enum` constants, (2) the other members such as constructor, fields and methods (optional)

- An `enum` constructor can specify any number of parameters and can be overloaded

Enumerations

- For every `enum`, the compiler generates the `static` method `values` that returns an array of the `enum`'s constants.

- When an `enum` constant is converted to a `String`, the constant's identifier is used as the `String` representation.

```
Direction d = Direction.NORTH;
System.out.println(d.toString()); // prints "NORTH"
```

Example

enum constants (objects in this example) initialized with constructor calls

```java
public enum Book {
    JHTP("Java How to Program", "2012"),
    CHTP("C How to Program", "2007"),
    IW3HTP("Internet & World Wide Web How to Program", "2008"),
    CPPHTP("C++ How to Program", "2012"),
    VBHTP("Visual Basic 2010 How to Program", "2011"),
    CSHARPHTP("Visual C# 2010 How to Program", "2011");

    private final String title;
    private final String copyrightYear;
    private Book(String bookTitle, String year) {
        title = bookTitle;
        copyrightYear = year;
    }
    public String getTitle() {  return title;  }
    public String getCopyrightYear() {  return copyrightYear;  }
}
```

Just like normal classes, defining public methods for clients to use the enum type

Only six Book objects will be created, constants such as Book.JHTP store the references.

Example

```java
import java.util.EnumSet;
    public class EnumTest {
    public static void main(String[] args) {
        System.out.println("All books:\n");
```

Values() returns an array of the **enum**'s constants

```java
for(Book book : Book.values())
    System.out.printf("%-10s%-45s%s\n", book,
            book.getTitle(), book.getCopyrightYear());
```

```java
        System.out.println("\nDisplay a range of enum constants:\n");
```

```java
for(Book book : EnumSet.range(Book.JHTP, Book.CPPHTP))
    System.out.printf("%-10s%-45s%s\n", book,
            book.getTitle(), book.getCopyrightYear());
    }
}
```

EnumSet's method `range()` returns a collection of the **enum** constants in the specified range of constants

Example

```
All books:

JHTP       Java How to Program                         2012
CHTP       C How to Program                            2007
IW3HTP     Internet & World Wide Web How to Program    2008
CPPHTP     C++ How to Program                          2012
VBHTP      Visual Basic 2010 How to Program            2011
CSHARPHTP  Visual C# 2010 How to Program               2011

Display a range of enum constants:

JHTP       Java How to Program                         2012
CHTP       C How to Program                            2007
IW3HTP     Internet & World Wide Web How to Program    2008
CPPHTP     C++ How to Program                          2012
```