Writing Java Programs

- US time zone program
  - Times as objects
  - Conversion methods
  - Object creation
  - Use of `this` to refer to receiver
- More on Java methods
- If statements
Time Conversion Specification

• Write a program that takes a time in hours and minutes plus a US time zone, and returns that corresponding time in that time zone.

• UStime class
  Attributes: int hours, minutes;
  Methods: constructor + ways to convert times, and a way to convert a UStime object to a String for output.
UStime Class

• Methods:
  – conversion methods: cvrtPacific();
    cvrtMountain();  cvrtCentral();
  – constructor: UStime(int h, int m);
  – conversion to String: toString();

• Algorithm
  – New time is
    \[(\text{old time} + 12 - \text{time zone difference}) \mod 12\]
  except when this yields 0 which corresponds to 12
  (either midnight or noon)
public UStime cvrtPacific(){
    //creates new UStime object, converts it to Pacific time and then returns it
    UStime t = new UStime(0,0);
    // 3 hours difference
    t.hours = (this.hours + 9) % 12;
    if(t.hours == 0) t.hours = 12;
    t.minutes = this.minutes;
    return t;}

see UStime.java
Simple If Statement

\[ \text{<ifStmt> } \rightarrow \text{if (<condition>) <statement>}
\]

\[ \hspace{1cm} [\text{else <statement>} ] \]

\begin{align*}
\text{if } (x>0) & \text{ return } 2*x; \\
\text{if } ((x>0) \&\& (y==5)) & \text{ z = x*y;} \\
\text{if } (x<0) & \text{ y = 5;} \\
\hspace{1cm} & \text{else return } 2*x; \\
\text{if } (x<0) & \text{ return } x*y; \\
\hspace{1cm} & \text{else if } (x==0) \text{ return } x;
\end{align*}
public String toString(){
    return (this.hours + " : " + this.minutes);
}

• Takes instance variables of UStime object, which are ints and converts them to String using default toString() on ints plus concatenation.
main method

- Creates USTime object for 12 noon
- Calls each of 3 conversion methods on this object and prints their results
main method in UStime class

```java
public static void main(String[] args) {
    //time is twelve noon EST
    System.out.print("Twelve noon EST is ");
    UStime z = new UStime(12, 0);
    System.out.println((z.cvrtCentral()) + "in Chicago");
    System.out.println();
    System.out.print("Twelve noon EST is ");
    System.out.println((z.cvrtMountain()) + "in Denver" + "\n");
    System.out.print("Twelve noon EST is ");
    System.out.println((z.cvrtPacific()) + "in San Francisco" + "\n");
}
```
Things to Notice

• Constructor has parameters, conversion methods need no parameters
• Object creation with `new` in main method
• Use of `println` and `print` for output
• Use of `this` to refer to receiver object within a method
• **If statement** for conditional execution
• Definition of `toString()` function to provide for output of UStime objects
Methods

<method_decl> → [ <modifiers> ] <kind> <method_name> ( [ <parameters> ] ) { <declarations> <statements> return <expr>; // if not void type }

<modifiers> → public | private
<kind> → void | <type> | <classname>
<type> → int | bool | double | etc.
Examples

from UStime class:

```java
public UStime cvrtCentral () {
    ...
}
public String toString( ) { ... }  
public static void main (String [] args ) { ... }  
public UStime timeConvert (String s) { ... }
```
Parameters

- Parameters used to input extra information to a method, besides the instance variables of the object to which it is applied

\[
\text{<formal>} \rightarrow \text{<type>} \text{<var>} | \text{<classname> <objectname>}
\]

\[
\text{<parameters>} \rightarrow \text{<parameters>}, \text{<formal>} | \text{<formal>}
\]
Method Invocation

\(<\text{method}\_\text{call}>) \rightarrow \text{<method}\_\text{name}>([\text{<actuals> }]);

//need actuals if have parameters

//action is performed on receiver (this) - implied

\(<\text{method}\_\text{call}>) \rightarrow \text{<objectname> . <method}\_\text{name}>([\text{<actuals> }]);//action is performed on

//object

\(<\text{method}\_\text{call}>) \rightarrow \text{<classname> . <method}\_\text{name}>([\text{<actuals> }]);//class method - to see later

\text{<actuals>} \rightarrow \text{<expr>} | \text{<actuals> , <expr>
Parameters

- Must be same type and number of formals in method declaration as actuals in method invocation
- Before the method begins to execute, the value of each actual is assigned to the corresponding formal; call by value semantics
- Value returned by a non-void method is value of expression in return statement
Examples

- in red oval is first kind of method call where the implied object is the receiver(this);
- in blue dashed oval is second kind of call, where a NimState object is what is returned as a value from method invocation (removeOne())

\[ z.cvrtCentral(); \]
\[ UStime(12,0); \]
Procedural Abstraction

• Examine the code in UStime program and find redundancy in conversion methods
• Extract common functionality and can parameterize calculation
• Encapsulate in a method
• Goal: to obtain shorter, more understandable code which perhaps can be used again; should do at algorithm design time BEFORE CODING

see UStime-procs.java
New Single Conversion Method

```java
public UStime timeConvert(String s) {
    int timeDiff;//local variable
    UStime t = new UStime(0,0);
    if (s == "Pacific") timeDiff = 3;
    else if (s == "Mountain") timeDiff = 2;
    else if (s == "Central")  timeDiff = 1;
    else {
        System.out.println("Error in input time" + " zone");
        exit(1);
        return null;
    }
    t.hours = (this.hours + 12 - timeDiff)%12;
    if (t.hours == 0)  t.hours = 12;
    t.minutes = this.minutes;
    return t;
}
```

nested if statement
Nested If Statements

- Check on related conditions or membership in a range of values

if (<cond1>) <stmt>  // <cond1> is true
    else if (<cond2>) <stmt>
        // !<cond1> && <cond2> is true
    else if (<cond3>) <stmt>
        // !<cond1> && !<cond2> && <cond3>
        // is true
    else <stmt>  // !<cond1> && !<cond2> && !<cond3>
        // is true

(Note: ! means logical not)
Nested If Statements

• Movie theatre tickets - disjoint categories
  
  ```java
  if (age>=12 && age<20) person="student";
    else if (age>65) person="senior_citizen";
    else if (age<3) person="free_admit";
    else person="regular_customer";
  ```

• Character sequence - range of values
  
  ```java
  if(c<'a')System.out.println("not letter");
    else if (c <= ‘z’)
      System.out.println("letter");
    else System.out.println("not a letter");
  ```