Computing Beginners Programming Tutorial 3

Susan Eisenbach

5 December 2007

1. Assume that a `Line` is declared as:

```
class Line{
      char[20] buf = new char[20];
      //number of characters in the Line
      int len;
}
```

   a. Write a method `readLine` to read a `Line` from the keyboard. Put the characters into `buf` and the number of characters into `len`.
   b. Write a method `removeSpaces` that takes a `Line` and returns a new `Line` with all the space characters removed.
   c. Write a method `readLineNoSpaces` such that

```
readLineNoSpaces() == removeSpaces(readLine())
```

2. In the lecture 9 (slides 14-16) notes there are other methods that are needed to implement `readExpression`.

```
Expression readExpression(){
Line line;
Expression e;
      line = readLineNoSpaces();
      if (isValid(line))
       {e = convert(line); return e;}
      else
       {println(syntaxError); return readExpression();}
}
```

   Assume that a `Line` is declared as above. Write Java code for:

```
a. boolean isValid(Line l) { your code goes here }
b. Expression convert(Line l) { your code goes here }
```

   You may need to write other methods to implement these.

1.

a.
```
Line readLine( ){
   Line temp;
      temp.buf[temp.len] = read();
      while (temp.len <= temp.buf.length-2 &&
               temp.buf[temp.len] != '\n'){
            temp.len = temp.len + 1;
            temp.buf[temp.len] = read();
      }
      return temp;
   }
```

b.
```
Line removeSpaces(Line l){
   Line temp;
      temp.len = 0 ;
      for (int i=0; i<l.len; i++){
            if (l.buf[i] != ' ') {
                  temp.buf[temp.len] = l.buf[i];
                  temp.len = temp.len + 1;
            }
      }
      return temp;
   }
```

c.
```
Line readLineNoSpaces( ){
   Line temp;
      temp.buf[temp.len] = readChar();
      while (temp.len <= temp.buf.length-2 && temp.buf[temp.len] !=
   '\n'){
            temp.len = temp.len + 1;
            temp.buf[temp.len] = readChar();
      }
      return temp;
   }
```

2.

a.
```
boolean isValid(Line l){
   int i = 0;
   int j;
      while (isDigit(l.buf[i]) && i < l.len){i = i+1;}
      if ( i==0 || i == l.len) {return false;}
      if (isOp(l.buf[i])) {
            i = i+1;
            j = i;
            while (isDigit(l.buf[i]) &&  i < l.len){i = i+1;}
                        if ( i == j) { return false;}
            }
      else {return false;}
      return i == l.len;
   }
```

b.
```
Expression convert(Line l){
   assert (isValid( l ));
   Expression e;
   int i=0;
      while (isDigit(l.buf[i]) && i < l.len){
            e.first = 10*e.first + convertDigit(l.buf[i]);
            i = i+1;
      }
      e.op = l.buf[i];
      i = i+1;
      while (isDigit(l.buf[i]) && i < l.len){
            e.sec = 10*e.sec + convertDigit(l.buf[i]);
            i = i+1;
      }
      return e;
   }
```