Mr. Robinette's AP Computer Science Class

APLesson_08: Recursion

APLesson_08: Recursion

As we learned in previous lessons, a recursive method is a method that calls itself. Previously, we used recursion to check user input to ensure it is correct. But we can also use recursion to create loops. Remember the following count program from Lesson_06...

```
public static void forCount(int to, int by)
{
    for(int i = by; i <= to; i += by)
    {
        System.out.println(i);
    }
}
```

Below is an example of how we would do the same thing using recursion

```
public static int recurCount(int to, int curr, int by)
{
    if(curr > to)
        return to;
    else
        System.out.println(curr);
        return recurCount(to, curr + by, by);
```

Manipulating Strings with Recursion

```
public static void main(String[]args)
{
    Scanner kb = new Scanner(System.in);
    System.out.println("Please enter a word: ");
    String word = kb.next();
    wordBox(word, 0);
}

public static String wordBox(String word, int num)
{
    if (num >= word.length())
        return "";
```

Forums

- per5Forum
- per4Forum
- per3Forum
- per2Forum

```
    else
    {
        System.out.println(word);
        wordBox(word, num + 1);
    }
    return "";
}
```

Mathematical functions with Recursion

```
public static void main(String[]args)
{
    Scanner kb = new Scanner(System.in);
    System.out.println("Please enter a number");
    int number = kb.nextInt();
    System.out.println(recur(number));
}
public static int recur(int n)
{
    if (n == 0)
        return 1;
    else
        return n * recur(n-1);
}
```

Recursion in AP Java

APLab_08

Ex_01: Underscores

Write a program that takes the spaces out of sentences and replaces them with underscores.

1. main() method - Take user input for a sentence
2. replace() method

    - If a space is not found in sentence
        - return sentence
    - Otherwise
        - recursive call to replace(sentence before space + underscore + sentence after space)

Ex_02: Center

Write a program that centers three words together.

1. main() method - Take user inputs for three words
2. makeCenter() method
    - If word length is greater than or equal to 20
        - return the word
    - Otherwise
        - Recursive call to makeCenter(word + one space on each side)

3. Run makeCenter 3 times, using each word as a parameter.

Your finished product should look like this....

```
      word1
    secondWord
absoluteFinalWord
```

Ex_03: Lucky7s

Write a program that counts the number of 7s in a number.

1. main() method - Take a user input for a number
2. luck() method - number as a parameter
    - If number is greater than 0
        - If the right digit is 7
            - return 1 + recursive call to luck(number/10)
        - Otherwise
            - Return 0 + recursive call to luck(number/10)
    - If none of the above are true, return 0

Ex_04: TreeDeg60

Write a program that creates a triangle similar to the ones we did in previous exercises, but at a 60 degree angle.

1. main() method - take user input for a word
    - Set the stop variable to the length of the word
2. tree() method - word, start, and stop numbers for parameters
    - if start is less than or equal to stop
        - Print substring from 0 to start
        - Add 1 to start
        - recursive call to tree(word, start, stop)

Use print formatting to make your output look like this....

```
   B
  Bu
 Bus
Bust
```

```
 Buste
Buster
```

G+1 Recommend this on Google

No comments:

Post a Comment

Enter your comment...

Comment as: Unknown (Goo ▼ Sign out

Publish Preview □ Notify me

Home

Subscribe to: Posts (Atom)

Travel template. Template images by enot-poloskun. Powered by Blogger.