PLTW COMPUTER SCIENCE

Activity 1.1.5

Your Sci-Fi Name

Introduction

Have you ever wondered where science fiction books and movies get those strange, alien names for their characters? It is possible they used a computer algorithm. In this activity, you will create a science fiction name for yourself using the string methods you know and a new method you do not yet know.

Materials

- Computer with BlueJ

Resources

Reference Card for Basic Constructs

Procedure

Part I: Random Numbers

Part of the algorithm for generating your science fiction name will involve the use of random numbers. Random numbers are like rolling the dice—you never know what numbers you're going to get. Chatbots also use random numbers to give unpredictable responses to questions or to ask random questions themselves.

To generate a random number in Java, you will use a method from a new class, the `Math` class. The method you will use is `random()` and it has a new syntax:

```
Math.random();
```

To assign a random number to a variable, you will use:

```
double r = Math.random();
```

The `Math.random()` method generates a number from 0 to .999999999999999. This is a versatile algorithm that allows a programmer to generate a wide variety of random number values. For example, if you multiply the random number returned from `Math.random()` by 10, you would get random numbers between the values of 0 and 9.99999999999999. If you cast this value to an `int` and add 1 to the result, you will get a random number between 1 and 10.

1. Review the online documentation regarding **Random Numbers** and complete the "Check your understanding" exercise. How would you call or invoke a **static method** called `sqrt(25)` that is part of the `Math` class?

2. Explain how the following algorithms simulate dice rolls. Are both algorithms good for generating a random dice roll? Why or why not?

```
// algorithm 1:
int diceRoll = (int)(Math.random() * 11) + 2;

// algorithm 2:
int diceRoll = (int)(Math.random() * 6) + 1;
diceRoll += (int)(Math.random() * 6) + 1;
```

Part II: Your Personal Science Fiction Name

Your science fiction name will be generated using a combination of `String` methods, random numbers, and concatenation. All input to the algorithm must be at least three characters long, and for best results, use lowercase letters with no spaces.

3. Get a copy of *1.1.5SciFiName_StarterCode_BlueJ* from your teacher. Extract or copy the files to your *`BlueJProjects`* folder and open them in BlueJ.

4. Open the `SciFiName` class and briefly observe the code. You have also been given a class called `UserInput` that gathers input from a user. Feel free to review this code; just know you will not be required to modify it.

5. At the end of the `main` method, you will see a comment that says `// generate a SciFi name`. You will add all of your code below this comment.

6. Your first Sci-Fi name will be generated using the following algorithm:

    - Using the `indexOf` and `substring` methods, get the first *three* letters of your first name.
    - Similarly, get the first *two* letters of your last name.
    - Using concatenation, combine these results to generate your Sci-fi first name.

7. Likewise, your Sci-fi last name will be generated using a similar algorithm:

    - Get the first *two* letters of the city you were born in.
    - Get the get the first *three* letters of your elementary (or previous) school.
    - Using concatenation, combine these results to generate your Sci-Fi last name.

8. Your Sci-Fi place of origin will be more randomly generated and will extract letters from the end of a name:

    - Generate a random number between 1 and (length – 1) of a relative's first name. For example, the result may be 2 as shown in the table:

E	d	i	t	a
0	1	2	3	4

    - Get the last letters of your relative's first name beginning at the random location through to the end of the string.

E	d	i	t	a
0	1	2	3	4

    - Generate a random number between 1 and (length – 1) of another relative's name. You may choose to use a friend's name in place of a relative's name.
    - Get the last letters of this name beginning at the random location through to the end of the string.
    - Using concatenation, combine the results from b and d to generate your Sci-Fi place of origin.

9. Print a friendly message such as:

```
"Hello carki chsal of lesomas. Welcome!"
```

Part III: Enhance Your Algorithm (Optional)

The `String` class has more methods than you will be required to know and some of them can enhance programs to make them more interesting or fun to use.

10. Explore the **Java API online documentation for the String class** and a new method `toLowerCase`. Convert the Sci-Fi names and place of origin to lowercase letters just in case users enter uppercase letters. Then change the user instructions.
11. Explore the new `String` method `toUpperCase`. Along with `substring`, capitalize the first letter of each Sci-Fi name and the place of origin.

Conclusion

1. In this activity, you generated random numbers between 1 and the (length – 1) of your first relative's name. Explain why the algorithm specified this length and describe the pattern that resulted.

2. Explain how the following two statements return different results:

```
int r1 = (int)(Math.random()) * 10 + 1;
int r2 = (int)(Math.random() * 10) + 1;
```