Module 2 Assignment

Hide Assignment Information

Instructions

Similar to the Module 1 assignment each subsection of Module 2 should be broken up into packages. Each subsection will be out of 10 marks. Submit the project as a github link.

Module 2.1 assignment:

For the following class create an equals, hashcode and toString method:

```
public class Planet {
    private String designation;
    private double massKg;
    private double orbitEarthYears;
    private PlanetType type;

    public Planet(String designation, PlanetType type){
        this.designation = designation;
        this.type = type;
    }

    public Planet(String designation, double massKg, double orbitEarthYears, PlanetType type) {
        this.designation = designation;
        this.massKg = massKg;
        this.orbitEarthYears = orbitEarthYears;
        this.type = type;
    }
}
```

Use the designation for the equals and hashcode and designation and type in the toString.

Module 2.2 assignment:

Simulate a deck of cards. Use an Enum to represent the suits and make sure there is a way to distinguish between face cards and number cards. Each card should also have a value associated with it, for simplicity Ace will be 1.

Next put them into a linked list and simulate a stack using a method that creates a deck of cards. Try pulling a hand of seven cards stored in a second list.

Module 2.3 assignment:

Copy the code from 2.2 into 2.3 and create 2 custom comparators, one that sorts by suits and the other that sorts by face card or not. (Ace is not a face card)

Module 2.4 assignment:

Download the following text file: illiad.txt

Create a class called Word.
Using a set, count the number of unique words inside the text. Note the following:
"Home", "home", "home.", "home?", "home," are all the same word.

Module 2.5 assignment:

Caesar Cipher

Introduction
The Caesar cipher is one of the earliest known and simplest ciphers. It is a type of substitution cipher in which each letter in the plaintext is 'shifted' a certain number of places down the alphabet. For example, with a shift of 1, A would be replaced by B, B would become C, and so on. The method is named after Julius Caesar, who apparently used it to communicate with his generals.

Example

To pass an encrypted message from one person to another, it is first necessary that both parties have the 'key' for the cipher, so that the sender may encrypt it and the receiver may decrypt it. For the caesar cipher, the key is the number of characters to shift the cipher alphabet.

Here is a quick example of the encryption and decryption steps involved with the caesar cipher. The text we will encrypt is 'defend the east wall of the castle', with a shift (key) of 1.

plaintext: defend the east wall of the castle
ciphertext: efgfoe uif fbtu xbmm pg uif dbtumf

It is easy to see how each character in the plaintext is shifted up the alphabet. Decryption is just as easy, by using an offset of -1.

plain: abcdefghijklmnopqrstuvwxyz
cipher: bcdefghijklmnopqrstuvwxyza

Obviously, if a different key is used, the cipher alphabet will be shifted a different amount.

Mathematical Description

First we translate all of our characters to numbers, 'a'=0, 'b'=1, 'c'=2, ... , 'z'=25. We can now represent the caesar cipher encryption function, e(x), where x is the character we are encrypting, as:

$$e(x) = (x + k) \pmod{26}$$

Where k is the key (the shift) applied to each letter. After applying this function the result is a number which must then be translated back into a letter. The decryption function is :

$$e(x) = (x - k) \pmod{26}$$

From <http://practicalcryptography.com/ciphers/caesar-cipher/>

Assignment:

Using Maps create a class that allows someone to instantiate with a number between 1 and 25 representing the number of shifts. Have 2 methods one that encrypts a message from their chosen shift and one that decrypts from chosen shift.

Bonus

Create a method that cracks the shift between 1 and 25. Use the following text document to assist you. words_alpha.txt

Submit Assignment

Allowed File Extensions

zip

Files to submit *

(0) file(s) to submit

After uploading, you must click Submit to complete the submission.

Add a File **Record Audio**

Comments

Submit **Cancel**