JA111

Day-5

Pure Encapsulation

- The class must be public
- All the fields should be private
- For each field there should be corresponding public getter and setter method.
- It should have a zero argument constructor.
- It may have parameterized constructor (it is not the minimum requirement)

Advantages

- The JavaBean properties and methods can be exposed to another application.
- It provides an easiness to reuse the software components.

Taking input in java

- **Add following line outside the class, after package statement**

```
import  java.util.Scanner;
```

- **Create an object of Scanner class; pass system.in as parameter while creating object**

```
Scanner sc = new Scanner(System.in);
```

- **Call the method(s) of Scanner class to take input**

```
int nextInt()                    double nextDouble()
String next()                    String nextLine()
```

- **close the object of Scanner class using close() method**

```
Sc.close();
```

String class in Java

- java does not implement string as a array of character instead of that java provides us a inbuilt class that is String.
- In java every object of String class is immutable that is it is impossible to change characters that a String object have but it is perfectly acceptable to update reference of String class.
- Each time we made changes in String literal we are creating a new object of String class. Previous String object remain unchanged. Immutability is looking a serious restriction side but due to performance, security and synchronization reason that restriction is imposed.
- String class implements CharSequence interface and declared as final hence it is not possible to create a subclass of String class.
- Anything inside the double quotation ("") marks is considered as String literal, An object of String class can be created using new keyword also.

Important Methods of String class

char charAt(int position)
byte[] getBytes()
static String valueOf(int b)
static String valueOf(Object b)
boolean equalsIgnoreCase(Object str)
int compareToIgnoreCase(String str)
boolean endsWith(String str)
int indexOf(String str)
int lastIndexOf(String str)
String trim()
String replace(char ori, char repl)
String toUpperCase()
int length()
char[] toCharArray()
static String valueOf(double b)
boolean equals(Object str)
int compareTo(String str)
boolean startsWith(String str)
int indexOf(int ch)
int lastIndexOf(int ch)
String substring(int startIndex)
String substring(int start, int end)
String toLowerCase()

StringBuffer & StringBuilder class in Java

- Both classes provides for mutable sequence of characters.
- Both classes are final so they cannot be inherited.
- Both implements CharSequence interface and declared as final hence it is not possible to create a subclass of os both classes.
- StringBuilder class is not thread-safe but StringBuffer is thread-safe.
- StringBuilder class is faster than the StringBuffer class

int capacity()	void setCharAt(int position, char ch)
StringBuffer reverse()	StringBuffer insert(int index, String str)
StringBuffer append(String str)	StringBuffer delete(int start, int end)
void setLength(int len)	StringBuffer deleteCharAt(int location)