Program6: Develop a JAVA program to create an interface Resizable with methods resizeWidth(int width) and resizeHeight(int height) that allow an object to be resized. Create a class Rectangle that implements the Resizable interface and implements the resize methods

```
interface Resizable
{
  void resizeWidth(int width);
  void resizeHeight(int height);
}
class Rectangle implements Resizable {
  private int width,height;

  public Rectangle(int width, int height)
  {  this.width = width;  this.height = height; }

  @Override
  public void resizeWidth(int width)
  { this.width = width; }

  @Override
  public void resizeHeight(int height)
  { this.height = height; }

  public void displayDimensions()  {
   System.out.println("Rectangle dimensions: Width = "
                  +width+", Height = "+height);
  }
}

public class Main {
  public static void main(String[] args) {
    Rectangle rect = new Rectangle(100, 50);

    System.out.println("Original dimensions:");
    rect.displayDimensions();

    rect.resizeWidth(150);
    rect.resizeHeight(75);
```

```
        System.out.println("After resizing:");
        rect.displayDimensions();
      }  }
```

Program 8: Develop a JAVA program to raise a custom exception (user defined exception) for DivisionByZero using try, catch, throw and finally.

```
class DivisionByZeroException extends Exception {
  public DivisionByZeroException(String message)
  { super(message); }
}

public class Main {
   public static int divide(int numerator, int denominator)
                          throws DivisionByZeroException
   {
     if (denominator == 0)
     throw new DivisionByZeroException("Error: Cannot
                                divide by zero!");
        return numerator / denominator;
}
 public static void main(String[] args)
 {
    int a = 10;
    int b = 0;
//change the 'b' value other than 0 in the 2nd time execution

    try {
       int result = divide(a, b);
       System.out.println("Result: "+result);
    }
    catch (DivisionByZeroException e)
    {System.out.println("Caught Exception: "+e.getMessage());}
    finally
    {System.out.println("Division operation attempted."); }
 }
}
```

Program9:Write a program to illustrate creation of threads using a runnable class. (start
method start each of the newly created thread. Inside the run method there is sleep()
for suspend the thread for 500 milliseconds).

```
package Week10;

class MyRunnable implements Runnable {
private String threadName;

public MyRunnable(String name) {
this.threadName = name;
}

public void run() {
for (int i = 1; i <= 5; i++) {
System.out.println(threadName + " - Count: " + i);

try {
Thread.sleep(500); // Pause for 500 ms
}
catch (InterruptedException e) {
System.out.println(threadName + " was interrupted.");
}
}

System.out.println(threadName + " has finished execution.");
}
}

public class Main {
public static void main(String[] args) {
```

```
MyRunnable task1 = new MyRunnable("Thread-1");
MyRunnable task2 = new MyRunnable("Thread-2");

// Create Thread objects and pass Runnable instances
Thread t1 = new Thread(task1);
Thread t2 = new Thread(task2);

// Start the threads
t1.start();
t2.start();
}
}
```

Program 10: Implement a JAVA program to illustrate the use of different types of character extraction, string comparison, string search and string modification methods.

```
package Strings;
public class Example1
{
public static void main(String[] args) {
String str1 = "Hello, World!";
String str2 = "hello, world!";

// 1. Character Extraction
System.out.println("Character Extraction:");
System.out.println("Character at index 7 in str1: " +
str1.charAt(7));
System.out.println();

// 2. String Comparison
System.out.println("String Comparison:");
System.out.println("str1 equals str2? "+
str1.equals(str2));
System.out.println("str1 equalsIgnoreCase str2? "+
```

```
str1.equalsIgnoreCase(str2));
System.out.println("str1 compareTo str2: "+
str1.compareTo(str2));
System.out.println();


// 3. String Searching
System.out.println("String Searching:");
System.out.println("Does str1 contain 'World'? "+
str1.contains("World"));
System.out.println("Index of 'o' in str1: "+
str1.indexOf('o'));

System.out.println("Last index of 'o' in str1: "+
str1.lastIndexOf('o'));
System.out.println("Does str1 start with 'Hello'? "+
str1.startsWith("Hello"));
System.out.println("Does str1 end with '!'? "+
str1.endsWith("!"));
System.out.println();

// 4. String Modification
System.out.println("String Modification:");
System.out.println("str1 to upper case: "+
str1.toUpperCase());
System.out.println("str2 to lower case: "+
str2.toLowerCase());
System.out.println("Replace 'World' with 'Java': "+
str1.replace("World", "Java"));
System.out.println("Substring from index 7: "+
str1.substring(7));
System.out.println("Substring from index 0 to 5: "+
str1.substring(0, 5));
System.out.println("Trimmed string: " + (" Hello Java! ".trim()));
}
```

```
}
```

Program 11: Develop a Java application that checks whether a given string is a palindrome,
designed using the SOLID principles of object-oriented programming.

```
class PalindromeChecker {

    public boolean isPalindrome(String text) {
        String clean = text.replaceAll("\\s+", "").toLowerCase();
        String reversed = new StringBuilder(clean).reverse().toString();
        return clean.equals(reversed);
    }

    // Moved here (allowed & simpler)
    public void check(String text) {
        if (isPalindrome(text)) {
            System.out.println(text + " → Palindrome");
        } else {
            System.out.println(text + " → Not a palindrome");
        }
    }
}

public class PalindromeApp {

    public static void main(String[] args) {

        PalindromeChecker checker = new PalindromeChecker();

        checker.check("madam");
        checker.check("hello");
        checker.check("A man a plan a canal Panama");
    }
}
```