

Judith Koch - Übung 2

2.1 ggT

Um den ggT zu berechnen, überprüfe ich erst, welche der zwei Zahlen grösser ist. Dann benutze ich den Modulooperator, um das eigentliche Ergebnis zu finden:

```
private int calculateGcd(int c, int d){
    int result = 1;
    for (int i=1; i<=d; i++){
        if (c%i == 0 && d%i ==0){
            result = i;
        }
    }
    return result;
}
```

2.2 n über k (Binomialkoeffizient)

Bei der Berechnung des Binomialkoeffizienten muss n grösser/gleich k sein. Des Weiteren muss ich eine Hilfsfunktion implementieren, die die Fakultätswerte berechnet.

```
public int binCoefficient(int a, int b){
    if(a>=b){
        return factorial(a)/(factorial(a-b)*factorial(b));
    } else {
        return -1;
    }
}
```

2.3 Liste/Knoten

Das Interface Liste (List) liefert den Rahmen fuer die Klasse Knoten (Node). Die getLenth()-Methode ist die interessanteste Methode dieser Klasse, da das „Stille-Post-Verfahren“ die Anzahl der verknüpften Knoten erfragt.

```
public int getLength(){
    int count = 1;
    if(next != null){
        count += next.getLength();
    }
    return count;
}
```

2.4 Stapel

Die wichtigste Methode der Klasse Stapel (Stack) ist getLatest():

```
public Node getLatest(){
    Node last = root;
    if(last != null){
        boolean isLast = false;
        while(!isLast){
            if(last.next == null){
                isLast=true;
            } else {
                last = last.next;
            }
        }
    }
    return last;
}
```

Basierend darauf kann ich push() und pop() implementieren.

2.5 Klammern prüfen

```
public class Brackets {  
  
    private Scanner scanner;  
  
    public void checkBrackets(){  
        Stack stack = new Stack();  
        scanner = new Scanner(System.in);  
        System.out.print("Input: ");  
        String string = scanner.next();  
        StringTokenizer st = new StringTokenizer(string);  
  
        while(st.hasMoreTokens()){  
            String nextString = st.nextToken();  
            if(){} else {}  
  
            System.out.println("Success");  
        }  
    }  
}
```

Die oben aufgeführten Uebersetzungen sind vor Info (SU) entstanden...

Fuer diese Klasse musste ich rausfinden, wie ich die Tastatureingabe verwenden kann. Dies hab ich mit Scanner und StringTokenizer geloest. Im Nachhinein habe ich festgestellt, dass der Scanner die ganze Zeile lesen muss, sprich scanner.nextLine() anstelle von scanner.next() gehoert.