Rekursive Variante

marc-oliver pahl mmiii

Pseudocode

```
sum_rek(n){
  if (n == 0) then return 0; fi
  return ( n + summe_rek( n-1 ) );
}
```

Java

```
public static int sum_rek(int n){
  if ( n == 0 ) return 0;
  return ( n + sum_rek( n-1 ) )
}
```

Iterative Variante

marc-oliver pahl mmiii

Pseudocode

```
sum_iter(n){
  k := n;
  erg := 0;
  while ( k >= 0 ) do {
    erg := k + erg;
    k := k - 1;
  } od
  return erg;
}
```

Java

```
public static int sum_iter(int n){
  int k = n;
  int erg = 0;
  while ( k >= 0) {
    erg = k + erg;
    k = k - 1;
  }
  return erg;
}
```

Kontext der Rekursion in Eclipse

marc-oliver pahl mmiii

sum_rek(3) => return (3 + sum_rek(2))

=> return (3 + (return (2 + sum_rek(1))))

=> return (3 + (return (2 + (return (1 + sum_rek(0))))))

Debug hier

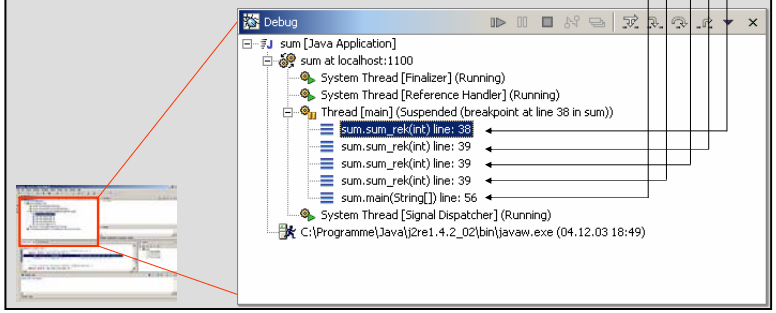
=> return (3 + (return (2 + (return (1 + (return 0))))))

=> return (3 + (return (2 + (return (1 + 0)))))

=> return (3 + (return (2 + 1)))

=> return (3 + 3)

=> 6



Ablauf der Iteration in Eclipse

marc-oliver pahl mmiii

sum_iter(3) => [erg = 0; k = 3;]

=> < (k >= 0)? > Ja =>=> [erg = 3; k =2;]

=> < (k >= 0)? > Ja =>=> [erg = 5; k =1;]

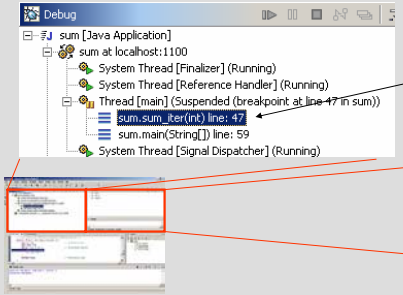
=> < (k >= 0)? > Ja =>=> [erg = 6; k =0;]

Debug hier

=> < (k >= 0)? > Ja =>=> [erg = 6; k =-1;]

=> < (k >= 0)? > Nein => return erg;

=> 6



Nur ein aktiver Aufruf von sum_iter