Freie Universität Berlin

Source codes and comprehension questions used in:

Busjahn, Teresa; Schulte, Carsten; Busjahn, Andreas: Analysis of code reading to gain more insight in program comprehension. In: Proceedings of the 11th Koli Calling International Conference on Computing Education Research (Koli, Finland), 2011. S. 1-9.

```
public class Bsp {
        public static int method ( int number ) {
                int result = 0 ;
                while ( number != 0 ) {
                        result = result + number % 10 ;
                        number = number / 10 ;
                }
                return result ;
        }
        public static void main ( String args[ ] ) {
                int result = method ( 123 ) ;
                System.out.println ( result ) ;
        }
}
```

This source code ...

- prints the mean value.
- **prints the checksum.**
- prints the factorial.
- prints nothing.
- I don't know.

Source code 1

```
public static void method ( int x , int y , char op ) {
        switch ( op ) {
                case '+' :
                        System.out.println ( x + y ) ;
                        break ;
                case '-' :
                        System.out.println ( x - y ) ;
                        break ;
                case '*' :
                        System.out.println ( x * y ) ;
                        break ;
                case '/' :
                        System.out.println ( x / y ) ;
                        break ;
        }
}
```

This source code ...

- tests, if an entered character is +, -, * or / and prints the character.
- **tests, if an entered character is +, -, * or / and prints the result of the operation with the input values.**
- tests, if the entered character is +, -, * or / and prints the input values.
- tests, if the entered character is +, -, * or / and prints the character.
- I don't know.

Source code 2

```
public class D {
	public static void main ( String [ ] args ) {
		System.out.println ( test ( "anna" ) ) ;
	}
	public static boolean test ( String s ) {
		int i = 0 ;
		int j = s.length ( ) - 1 ;
		while ( i <= j ) {
			if ( s.charAt ( i ) != s.charAt ( j ) )
				return false ;
			i ++ ;
			j -- ;
		}
		return true ;
	}
}
```

This source code ...

- **tests, if “anna” is a palindrome, a word that stays the same, when read invertedly, and returns true.**
- tests, if “anna” is an acronym, an abbreviation made from initial letters, and returns true.
- tests, if “anna” contains vowels, and returns true.
- I don't know.

Source code 3

```
public static boolean method1 ( String str ) {
        char ch ;
        for ( int i = 0 ; i < str.length ( ) ; i ++ ) {
                ch = str.charAt ( i ) ;
                if ( ! ( ch >= 'a' && ch <= 'z' ) && ! ( ch >= 'A' && ch <= 'Z' ) )
                        return false ;
        }
        return true ;
}
public static void method2 ( ) {
        String str = "HelloWorld" ;
        if ( method1 ( str ) )
                System.out.println ( str ) ;
}
```

This source code …

- **tests, if all characters of the strings „HelloWorld“ are letters and prints the string.**
- tests, if all characters of the strings „HelloWorld“ are letters, but doesn't print the string.
- tests, if all characters of the strings „HelloWorld“ are vowels, and prints the string.
- tests, if all characters of the strings „HelloWorld“ are vowels, but doesn't print the string.
- I don't know.

Source code 4

```
protected static boolean doSomething ( double [ ] nums ) {
        for ( int i = 0 ; i < ( nums.length - 1 ) ; i ++ ) {
                if ( nums [ i ] > nums [ i + 1 ] )
                        return false ;
        }
        return true ;
}
public static void main ( String [ ] args ) {
        double [ ] a1 = { 1.0 , 3.343434343 , 5.0 } ;
        System.out.println ( doSomething ( a1 ) ) ;
}
```

This source code …

- sorts the elements of an array in ascending order.
- sorts the elements of an array in descending order.
- **tests, if the input array is sorted in ascending order.**
- tests, if the input array is sorted in descending order.
- I don't know.

Source code 5

```
public static double method ( double [ ] list )
                                throws DivideByZeroException {
        if ( list.length == 0 )
                throw new DivideByZeroException ( ) ;
        double result = 0 ;
        for ( int i = 0 ; i < list.length ; i ++ ) {
                result = result + list [ i ] ;
        }
                return result / list.length ;
}
```

This source code …

- **returns the mean value.**
- prints the checksum.
- prints nothing.
- prints the factorial.
- I don't know.

Source code 6

```
package project_A;

public class B {
	private int a ;
	public B ( int a , int b ) {
		this.a = a ;
		while ( b > 0 ) {
			this.a ++ ;
			b -- ;
		}
	}
	public static void main ( String [ ] arguments ) {
		B obj = new B ( 3 , 5 ) ;
		System.out.print ( obj.a ) ;
	}
}
```

This source code …

- prints the greatest common divisor of a and b.
- **prints the sum of a and b.**
- prints the least common multiple of a and b.
- prints nothing.
- I don't know.

Source code 7

```
public class A {
	public int calculate ( int num1 , int num2 ) {
		int result = num1 ;
		for ( int i = 2 ; i <= num2 ; i += 1 ) {
			result = result * num1 ;
		}
		return result ;
	}
	public static void main ( String [ ] args ) {
		int result = new A ( ).calculate ( 2 , 4 ) ;
		System.out.println ( result ) ;
	}
}
```

This source code ...

- calculates num1+num2.
- **calculates num1^num2 .**
- calculates num1*num2 .
- calculates num1/num2 .
- I don't know.

Source code 8

```
public static int calc ( int a , int b ) {
        if ( a == b )
                return a ;
        if ( a > b )
                return calc ( b , a - b ) ;
        return calc ( b , b - a ) ;
}
public static void start_calc ( ) {
        int result = calc ( 14 , 7 ) ;
        System.out.println ( result ) ;
}
```

This source code …

- **prints the greatest common divisor of a and b.**
- prints the sum of a and b.
- prints the least common multiple of a and b.
- prints nothing.
- I don't know.

Source code 9

```
private static float [ ] fmm ( float [ ] num_array ) {
        float a = num_array [ 0 ] ;
        float b = num_array [ 0 ] ;
        for ( int i = 0 ; i < num_array.length ; i = i + 1 ) {
                if ( num_array [ i ] < a )
                        a = num_array [ i ] ;
                else if ( num_array [ i ] > b )
                        b = num_array [ i ] ;
        }
        float [ ] r = { a , b } ;
        return r ;
}
```

This source code …

- tests, if the input array is sorted in ascending order.
- returns nothing.
- returns the first and last value of an array.
- **returns the minimum and the maximum of an array.**
- I don't know.

Source code 10

```
dpublic static void method1 ( int [ ] nums ) {
        int length = nums.length ;
        if ( length > 1 ) {
                for ( int i = 0 ; i < ( length / 2 ) ; i ++ )
                        method2 ( nums , i , length ) ;
        }
        method3 ( nums ) ;
}
public static void method2 ( int [ ] nums , int i , int length ) {
        int temp = nums [ length - i - 1 ] ;
        nums [ length - i - 1 ] = nums [ i ] ;
        nums [ i ] = temp ;
}
public static void method3 ( int [ ] nums ) {
        for ( int i = 0 ; i < nums.length ; i ++ )
                System.out.print ( nums [ i ] + " " ) ;
}
public static void main ( String [ ] args ) {
        int [ ] nums = { 1 , 2 , 3 , 4 , 5 } ;
        method1 ( nums ) ;
}
```

This source code …

- prints an empty array.
- sorts the elements of an array in ascending order and prints them.
- **prints the elements of an array in inverse order.**
- swaps the first half of the elements with each other and prints them.
- I don't know.