Exceptions - Klausuraufgaben

(1.) Exception ist ….!

(a.)eine Klasse
(b.)eine Schnittstelle
(c.)abstrakte Klasse
(d.) Keines davon

(2.) Exception is in dem Java Package!

(a.) java.lang
(b.) java.util
(c.) java.io
(d.) java

(3.) Die höchste Klasse in der Java Hierarchie ist

(a.) java.lang.Exception
(b.) java.lang.Error
(c.) java.lang.Throwable
(d.) java.lang.Object

(4.) Welches Schlüsselwort wird verwendet um eine Exception auszulösen?

(a.) catch
(b.) throw
(c.) throws
(d.) raise

(5.) Welcher Block wird unabhängig davon, ob eine Exception ausgelöst wird, ausgeführt?

(a.)throws
(b.)finally
(c.)catch
(d.)throw

(6.) Unter welchen Bedinungen wird der finally Block nicht ausgeführt?

(a.)Wenn ein Fehler auftritt
(b.)Wenn eine Exception aufgerufen wird
(c.)When System.Exit(1) aufgerufen wird
(d.)In allen Fällen

(7.) Welche Arten von Exceptions werden zur Kompilierungszeit ignoriert?

(a.)Laufzeit
(b.)Geprüfte
(c.)Beide
(d.)Keine

(8.) Was gibt folgendes Programm aus:

```
public class Overwelming08 {
  static String str = "a";

  public static void main(String[] args) {
    new Overwelming08().method1();
    System.out.println(str);
  }

  void method1() {
    try {
      method2();
    } catch (Exception e) {
      str += "b";
    }
  }

  void method2() throws Exception {
    try{
      method3();
      str += "c";
    }catch(Exception e){
      throw new Exception();
    }finally{
      str += "d";
    }
    method3();
    str += "e";
  }

  void method3() throws Exception {
    throw new Exception();
  }
}
```

(a.) ade

(b) adb

(c.) bcde

(d.) adbe

(9.) Was gibt folgendes Programm aus:

```
public class Overwelming09 {
  public static void main(String[] args) {
    m(); //call recursive method m()
    System.out.println("Nach der Fehlerbehandlung!");
  }

  static void m() {
    try {
      m();
    } catch (StackOverflowError e) {
      System.out.println("Stackoverflow Exception");
    }
  }
}
```

(10.)Was passiert, wenn catch und finally ein „return“ enthalten?

(a.)Es wird der Wert von dem catch zurückgegeben
(b.)Es wird der Wert von dem finally zurückgegeben
(c.)finally wird nicht ausgeführt
(d.)keins von den drei Möglichkeiten

(11.) Was ist die Ausgabe?

```
public class Overwelming11 {
  public static void main(String[] args) {
    System.out.println("method return -> "+m());
  }

  static String m(){
    try{
      int i=10/0;
    }catch(ArithmeticException e){
      return "catch";
    }finally{
      return "finally";
    }
  }
}
```

(a.) Laufzeitfehler

(b) „finally“

(c.) „catch“

(d.) Kompilerfehler

(12.)

```
public class Overwelming12 {
 static String str = "";
 public static void main(String[] args){
   try{
     str += "a";
     throw new Exception();
   } catch (Exception e) {
     str += "b";
   } finally  {
     str += "c";
     method();
     str += "d";
   }
   System.out.println(str);
 }
 static void method(){
   throw new NullPointerException();
 }
}
```

(a.) abc

(b) abcd

(c.) Fehler zur Laufzeit(da method diesen generiert)

(d.) Kompilerfehler

(13.)

```
class UserDefinedException extends RuntimeException{
  UserDefinedException(String s){
    super(s);
  }
}
public class Overwelming13 {
  public static void main(String... arg) {
    int i=1;
    if(i==1)
      throw new UserDefinedException("user defined exception");
    System.out.println("end");
  }
}
```

(a.) Kompilerfehler

(b) UserDefinedException wird ausgelöst

(c.) UserDefinedException wird ausgelöst und end ausgegeben

(d.) Nur wenn i=1 ist wird der Fehler ausgelöst

(14.)

```
class SuperClass{
  void method() throws NullPointerException{
    System.out.println("superClass method");
}
}
class SubClass extends SuperClass{
  void method() throws RuntimeException{
    System.out.println("SubClass method");
  }
}

public class Overwelming14 {
  public static void main(String[] args) {
    SuperClass obj=new SubClass();
    obj.method();
  }
}
```

(a.) Laufzeitfehler

(b) Kompilerfehler

(c.) superClass method

(d.) subClass method

(15.) Wie ist Ausgabe?

```
public class Overwelming15 {
 public static void main(String[] args)
 {
   methodl();
   System.out.println("after calling m()");
 }

 static void methodl(){
   method2();
 }

 static void method2(){
   method3();
 }

 static void method3(){
   throw new NullPointerException();
 }
}
```

(a.) „after calling m()“

(b) Laufzeitfehler

(c.) Kompilerfehler

(d.) Keine der drei obigen Antworten

(16.)

```
 import java.io.FileNotFoundException;
import java.io.IOException;
class SuperClass{
  void method() throws IOException{
    System.out.println("superClass method");
  }
}
class SubClass extends SuperClass{
  void method() throws FileNotFoundException{
    System.out.println("SubClass method");
  }
}
public class Overwelming16 {
  public static void main(String[] args) throws Exception {
    SuperClass obj=new SubClass();
    obj.method();
  }
}
```

(a.) Laufzeitfehler

(b) Kompilerfehler

(c.) superClass method

(d.) subClass method

(17.)

```
//import java.io.FileNotFoundException;
import java.io.IOException;
class SuperClass{
  void method() throws IOException{
    System.out.println("superClass method");
  }
}
class SubClass extends SuperClass{
  void method() throws FileNotFoundException{
    System.out.println("SubClass method");
  }
}
public class Overwelming16 {
  public static void main(String[] args) throws Exception {
    SuperClass obj=new SubClass();
    obj.method();
  }
}
```

(a.) Laufzeitfehler

(b) Kompilerfehler

(c.) superClass method

(d.) subClass method

(18.)

```
import java.io.IOException;
class SuperClass{
  void method() throws IOException{
    System.out.println("superClass method");
  }
}
class SubClass extends SuperClass{
  void method() throws NullPointerException{
    System.out.println("SubClass method");
  }
}
public class Overwelming18 {
  public static void main(String[] args) throws Exception {
    SuperClass obj=new SubClass();
    obj.method();
  }
}
```

(a.) Laufzeitfehler

(b) Kompilerfehler

(c.) superClass method

(d.) subClass method