注解、注释、Junit

1. (单选题)用于指定注解存活时间的元注解为（              ）

A. `@Target`：定义注解可以用于哪些Java元素（类、方法等）。

B. `@Retention`：定义注解保留的阶段（源代码、类文件、运行时）。

C. `@Documented`：指示注解信息应包含在javadoc中。

D. `@Inherited`：允许注解被继承，子类继承父类的注解。

2.(单选题)为了使得注解在程序运行时被反射读取，则应设置@Retention取值为（     ）

A. RetentionPolicy.SOURCE

B. RetentionPolicy.CLASS

C. RetentionPolicy.RUNTIME：

D. RetentionPolicy.JVM

3. (单选题)@Override注解只在源码阶段保留，它的@Retention取值为（            ）

A. RetentionPolicy.SOURCE

B. RetentionPolicy.CLASS

C. RetentionPolicy.RUNTIME

D. RetentionPolicy.JVM

4. (单选题)用于指定注解运用地方的元注解为（              ）

A. @Target

B. @Retention

C. @Documented

D. @Inherited

5. (单选题)设置一个注解只能修饰方法和类型（包括：类、接口、枚举）的元注解代码应为

A. @Target(ElementType.METHOD,ElementType.TYPE)

B. @Target({ElementType.METHOD,ElementType.TYPE})

C. @Retention(ElementType.METHOD,ElementType.TYPE)

D.@Retention({ElementType.METHOD,ElementType.TYPE})

6. (单选题)以下不属于Java注释的为（ ）

A. //这是单行注释

B./\*

这是多行注释

\*/

C./\*\*

 \* 这是文档注释

 \* \*/

1. <!这是一个单行注释 >

7. (多选题)抑制一个类上的所有警告，则能在该类上使用的代码为（ ）

A. @SuppressWarnings("all")

B. @SuppressWarnings(value="all")

C. @SuppressWarnings(value={"all"})

D. @SuppressWarnings(["all"])

E. @SuppressWarnings(value=["all"])

8. (多选题)关于Junit说法正确的是（ ）

A. Junit框架，可以独立于main方法完成单元测试

B. @Before注解修饰的方法，只在所有的测试方法前运行一次，而不会在每一个测试方法前都运行一次

C. Junit框架中，不能使用断言判断测试结果是否正确

D. 使用@Test注解的方法，必须是public void类型，且不能有任何参数。

E. Junit支持一个项目中所有方法的自动化测试

9. (填空题)已知Student的类定义为：

package fafu.jsxy;

public class Student {

    public void study() {

        System.out.println("studying");

    }

    public void speak(){

        System.out.println("speaking");

    }

    private void sleep(){  //私有方法

        System.out.println("sleeping");

    }

}

TestAnnotation类的功能为：通过反射，执行Student类中所有方法。使用反射的好处为：不需要知道Student类中定义的方法的名字，就能够执行Student类中的所有的方法。

请在下划线处填写正确的代码。

package fafu.jsxy;

import java.lang.reflect.Method;

public class TestAnnotation {

    public static void main(String[] args) throws Exception{

        Class \_\_\_studentClass\_\_\_\_ = Student.class; //获得Student的字节码对象

        Student student = (\_\_Student\_\_\_)studentClass.newInstance(); //通过字节码对象，创建Student对象

        \_\_\_Method\_\_\_\_[] methods = studentClass.getDeclaredMethods(); //通过字节码对象，获得Student类中的所有方法

        for (Method m:methods){

            m.setAccessible(true);  //暴力反射

            m.\_\_\_invoke\_\_(student); //执行方法

        }

    }

}

集合与泛型

1. (单选题)以下不属于Java的集合框架的组成部分是（）

A. ArrayList

B. HashMap

C. TreeSet

D. Arrays

2. (单选题)Java中提供了对集合中的元素进行迭代访问的接口为（）

A. Iterator

B. Collection

C. List

D. Set

3. (单选题)以下代码执行时，将引发的异常为（ ）

Plain Text

import java.util.\*;

class Test {

   public static void main(String[] args) {

       List<String> list = new ArrayList<>();

       for(int i=0;i<5;i++){

           list.add(String.valueOf(i));

       }

       for (String element:list){

           if (element.equals("0")){

               list.remove(element);

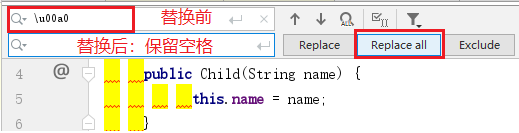
           }

       }

   }

}

注：拷贝代码到idea，如果报“illegal character U+00A0”错误，可使用快捷键“ctrl+r”，弹出“替换”对话框，将“\u00a0”全部替换为半角空格。



A. NullPointerException

B. ConcurrentModificationException

C. ArrayIndexOutOfBoundsException

D. ClassCastException

4. (单选题)以下程序的执行结果为：

Plain Text

import java.util.\*;

class Child {

   public String name;

   public Child(String name) {

       this.name = name;

   }

   @Override

   public boolean equals(Object o) {

       Child child = (Child) o;

       return Objects.equals(name, child.name);

   }

   public static void main(String[] args) {

       HashSet<Child> set = new HashSet<>();

       set.add(new Child("小红"));

       set.add(new Child("小花"));

       set.add(new Child("小红"));

       set.add(new Child("小明"));

       System.out.println(set.size());

   }

}

A. 1

B. 2

C. 3

D. 4

5. (单选题)关于List的说法正确的是（）

A. 不能通过下标访问List中的元素

B. List中的元素不能重复

C. List的长度可以动态改变

D. ArrayList和HashList为List接口的实现类

6. (单选题)

已知List list=new ArrayList()，以下代码为遍历输出list中的每个元素，则下划线处的代码为

Plain Text

for(int i=0;i<\_\_\_\_\_\_\_\_\_\_;i++){

   System.out.println(list.get(i));

}

A. list.length()

B. list.size()

C. list.length

D. list.size

7. (单选题)以下程序的执行结果为：

Plain Text

String s1 = "hello";

String s2 = "hello";

String s3 = new String("hello");

System.out.println(s1==s2);

System.out.println(s1==s3);

System.out.println(s1.equals(s3));

A.false

false

true

B.true

true

true

C.true

false

true

D.true

false

false

8. (多选题)关于HashMap和HashSet的说法正确的是（）

A. HashSet底层使用 HashMap作为存储结构。

B. HashSet 的 add() 方法底层调用了 HashMap 的 put() 方法。

C. HashSet中的元素可以重复

D. HashMap以键值对的形式添加元素，键不能重复，值可以重复

E.HashSet不能保证插入元素的顺序和循环输出元素的顺序一定相同

9. (多选题)已知：String a=null;

则运行时，会引发NullPointerException的代码为（ ）

A. System.out.println(a.equals("hello"));

B. System.out.println(a.length());

C.  System.out.println(a==null);

D. System.out.println("hello".equals(a));

E. System.out.println(Objects.equals(a,"hello"));

10. (填空题)

以下代码的功能为：使用HashMap存储一个英中词典，其中，一个英文单词对应多个中文翻译。当接收控制台输入的英文单词，将给出对应的多个中文翻译。

Plain Text

import java.util.\*;

class Test {

   public static void main(String[] args) {

       //1、创建词典

       Map<\_\_String,String[]\_\_\_> map = new HashMap<>();

       map.put("excellent", new String[]{"杰出的", "非凡的"});

       map.put("annotation", new String[]{"注解", "评注"});

       map.put("sale", new String[]{"销售", "销量"});

       //修改 “excellent”对应的中文翻译

       map.\_\_\_\_put \_\_\_\_\_\_\_("excellent", new String[]{"极好的", "出色的", "杰出的"});

       //2、输出词典的内容

       Set<Map.\_\_\_\_\_Entry\_\_\_\_\_<String, String[]>> mapSet = map.entrySet();

       Iterator<Map.Entry<String, String[]>> iter = mapSet.iterator();

       while (iter.hasNext()) {

           Map.Entry<String, String[]> entry = iter.next();

           System.out.print(entry.getKey() + ":");        //输出键

           for (String \_\_value\_\_\_\_ : entry.getValue()) {    //输出值

               System.out.print(value + " ");

           }

           System.out.println();

       }

       //3、接收控制台输入的英文单词，给出中文翻译

       Scanner scanner = new Scanner(System.in); // 创建Scanner对象

       System.out.println("请输入英文单词：");

       String input = scanner.nextLine(); // 接收控制台输入的一行文本

       if (map.containsKey(input)) {

           String[] outputs = map.\_\_\_get\_\_\_\_(input);

           System.out.println("该单词的中文翻译为：");

           for (String output : outputs) {

               System.out.println(output);

           }

       } else {

           System.out.println("无该单词对应的中文");

       }

   }

}

11. (填空题)以下代码的功能为：遍历ArrayList集合，删除以“实验1”结尾的所有字符串。

Plain Text

List<String> list = new ArrayList<>();

list.add("2020001张三实验1");list.add("2020001张三实验2");

list.add("2020002李四实验1");list.add("2020002李四实验2");

for (Iterator<String> iterator = \_\_\_list\_\_\_\_\_.iterator();iterator.\_\_\_hasNext()\_\_\_ ;) {

   String str = iterator.next();

   if(str.endsWith("实验1")){

      \_\_\_\_iterator.remove()\_\_\_\_\_ ; //删除迭代器指向的元素

   }

}

System.out.println(list);

Java Web开发入门

1. (单选题)关于B/S的说法错误的是（    ）

A. 客户端无需安装，有Web浏览器即可

B. B/S架构无需升级多个客户端，升级Web服务器即可

C. B/S表现要达到C/S程序的程度，需要较高的设计成本

D. B/S在速度和安全性上的设计成本较低

2. (单选题)要在浏览器的地址栏上隐藏传输给服务器的参数，需要将<form>标签的method属性设置为（         ）

A. hide

B. get

C. post

D. Submit

3. (单选题)已知register.html中的部分代码为：

Plain Text

<form action="register" method="post">

   用户名：<input name="username" type="text"/><br/>

   密码：<input name="userpass" type="password"/><br/>

   <input type="submit" value="注册"/>

   <input type="reset" value="重置"/>

</form>

RegisterServlet.java的部分代码为：

Plain Text

@WebServlet("/register")

public class RegisterServlet extends HttpServlet {

   @Override

   protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

       String username = req.getParameter("userName");

       String userpass = req.getParameter("userpass");

   }

}

如果用户在register.html页面中，不输入任何内容，直接点击“注册”按钮，则RegisterServlet的doPost方法中的变量username和userpass的值分别为（  ）

A. ""  ""

B. ""  null

C. null  null

D. null  ""

4. (填空题)在如图2-1所示的下载页面download.html中输入正确的密码“123456”，则DownloadServlet生成的页面中显示“idea下载”的链接，如图2-2所示；否则显示如图2-3所示的出错提示。

请在下划线处填写正确代码。

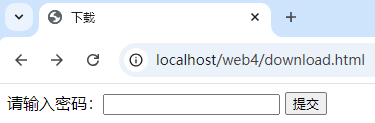


                          图2-1 下载页面

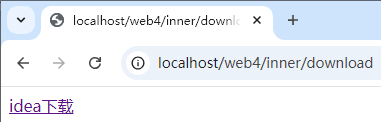


                    图2-2 密码正确后进入的下载页面

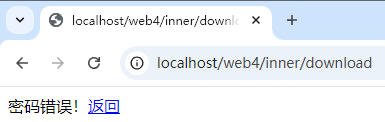


                    图2-3 密码错误

download.html文件（该文件直接位于项目的web节点下）：

Plain Text

<!DOCTYPE html>

<html lang="en">

<head>

   <meta charset="UTF-8">

   <title>下载</title>

</head>

<body>

<form action="inner/download" method="post">

   请输入密码：<input type="password" name="userpass"/>

   <input type="submit" value="提交">

</form>

</body>

</html>

DownloadServlet.java文件：

Plain Text

package it.serlvet;

import javax.servlet.ServletException;

import javax.servlet.annotation.WebServlet;

import javax.servlet.http.HttpServlet;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import java.io.IOException;

import java.util.Objects;

@WebServlet(value="\_/inner/download\_")

public class DownloadServlet extends HttpServlet {

   @Override

   protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

       \_\_\_resp\_\_\_.setContentType("text/html;charset=utf-8");

       String userpass = \_req\_\_\_.getParameter("userpass");

       if (\_\_\_\_Objects\_\_\_\_.equals("123456",userpass)){

           resp.getWriter().println("<a href='https://www.jetbrains.com.cn/idea/download/?section=windows'>idea下载</a>");

       }else{

           resp.getWriter().println("密码错误！"+"<a href='download.html'>返回</a>");

       }

   }

}

5. (多选题)属于向服务器发送get请求的情况包括（      ）

A. 在浏览器地址栏上直接输入URL，回车

B. 在浏览器上点击超链接

C. 使用form表单提交数据时，form标签中，没有写method属性

D. 在form标签中，指定method属性为：method=“get”

6. (多选题)与@WebServlet(urlPatterns="/inner/random") 作用等同的代码包括（）

A. @WebServlet(name="/inner/random")

B. @WebServlet(value="/inner/random")

C. @WebServlet(WebInitParam="/inner/random")

D. @WebServlet("/inner/random")

7. (多选题)关于POST和GET请求，说法错误的是（）

A. POST在请求行中提交数据，GET在请求体中提交数据

B. 向服务器传送大量数据，应使用POST

C. 涉及敏感数据的提交，应使用GET

D. 向服务器上传一张图片，应使用POST