基础题

1.B 2.C 3.D 4.B 5.D 6.B 7.AC 8.BCD 9.A 10.A

提高题

1.

（1）Java1.7中永久代被合并到堆中，在1.8中永久代被元空间替代，元空间位于本地内存

(2)静态变量和常量池从位于堆中的永久代转移到元空间中

2.

a=a-b;

b=a+b;

a=b-a;

3.

public class Car {

int speed;

public int getSpeed() {

return speed;

}

public void setSpeed(int speed) {

this.speed = speed;

}

public void testSpeed(){

if(speed>120){

System.out.println("当车速超过120公里/小时，罚款50元");

}

System.out.println("车速正常");

}

}

4.

public void doPost () {

CloseableHttpClient httpClient = HttpClientBuilder.create().build();

HttpPost httpPost = new HttpPost("http://localhost/PostController");

User user = new User();

user.setName("小明");

user.setAge(18);

user.setGender("女");

String jsonString = JSON.toJSONString(user);

StringEntity entity = new StringEntity(jsonString, "UTF-8");

httpPost.setEntity(entity);

httpPost.setHeader("Content-Type", "application/json;charset=utf8");

CloseableHttpResponse response = null;

try {

response = httpClient.execute(httpPost);

HttpEntity responseEntity = response.getEntity();

System.out.println("响应状态为:" + response.getStatusLine());

if (responseEntity != null) {

System.out.println("响应内容长度为:" + responseEntity.getContentLength());

System.out.println("响应内容为:" + EntityUtils.toString(responseEntity));

}

} catch (ClientProtocolException e) {

e.printStackTrace();

} catch (ParseException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

} finally {

try {

// 释放资源

if (httpClient != null) {

httpClient.close();

}

if (response != null) {

response.close();

}

} catch (IOException e) {

e.printStackTrace();

}

}

}

5.IOC是Inverse of Controller,即控制反转的缩写，把对象的创建，初始化和销毁都交给Bean容器管理，可以有效降低代码之间的耦合度。

AOP，面向切面编程，可以在切点执行业务前后配置通知，能够使代码更加灵活

6.开闭原则和单一职责原则

开放题