**深圳大学随堂测试试卷**

学院 专业 姓名 学号 座号

( 密 封 线 内 不 答 题；学院或任课教师可根据需要选择是否密封。 )

……………………………………………………密………………………………………………封………………………………………线……………………………………线………………………………………

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_

…

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 开/闭卷 | 开卷 | A/B卷 | | | A |
| 课程编号 | 1502850001 | 课程名称 | 互联网编程 | 学分 | 2+2 |

命题人(签字) 审题人(签字) 2022 年 5 月 11 日

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 题号 | 一 | 二 | 三 | 四 | 五 | 六 | 七 | 八 | 九 | 十 | 基本题总分 | 附加题 |
| 得分 |  |  |  |  |  |  |  |  |  |  |  |  |
| 评卷人 |  |  |  |  |  |  |  |  |  |  |  |  |

一．单选题（每空3分），共30分。在以下每小题给出的A 、B、C、D四个选项中，只有一个选项正确，请选出正确答案，并将答案填在题中括号内）。

1. Web的底层协议HTTP一般使用端口（B）进行通信。
2. 20
3. 80
4. 21
5. 81
6. 下面那个类可以用于创建直接能访问URL资源的流对象（C），读取网络数据流。
7. BufferedInputStream
8. BufferedReader
9. InputStream
10. DataInputStream
11. 下面不是互联网应用层协议的是（D）
12. FTP
13. HTTP
14. TELNET
15. UDP
16. 下列不属于TCP/IP参考模型的是（C）
17. 网际层
18. 应用层
19. 会话层
20. 主机-网络层
21. 以下IPv6地址不正确的是（C）？
22. AD80::ABAA:0000:00C2:0002
23. 2001:0:0:0:0:0:0:1
24. ::FFFF.192.168.89.9
25. 2000:DB0:2DE::E13
26. 从哪里可以找到描述TCP/IP的具体文档（D）？
27. JDK的JavaDoc文档
28. NIC的官方网站
29. 国际标准化组织（ISO）的官方网站
30. RFC的官方网站
31. 使用Executor线程池编程框架时，定义计算任务的类应该实现（C）接口。
32. Executor
33. Runnable
34. Callable
35. Future
36. 调用join方法的线程是（C）？
37. 主线程
38. 被连接的线程
39. 连接线程
40. 副线程
41. 以下关于线程相关内容说法正确的是（B）？
42. 同步块越大越好
43. 轮询和回调都是解决静态条件的有效方法
44. 对于Thread的派生子类，应该在编程时覆盖Thread的所有方法。
45. 线程优先级中，1表示优先级最高。
46. 有一种协议规定：如果客户端发送一行字符串“date”,服务器端就返回当前日期信息，如果客户端发送一行字符串“exit”，服务器端就结束与客户端的通信。这种协议应该是哪一层的协议（A）？
47. 应用层
48. 网际层
49. 主机-网络层
50. 传输层

二．判断题 （每小题2分，共10分，请给正确的打“√”，错误的打“×”，并填写题中括号内）。

1. 假定一个进程已经占用TCP的80端口，它就不可以占用UDP的80端口（ F ）
2. DataOutputStream类提供了8字节的写入方法，并且以big-endian格式写入 （ T ）
3. SMTP是发送电子邮件的协议（ T ）
4. 在关闭流之前不需要进行刷新操作（ F ）
5. 避免死锁可以利用同步方法（ F ）

三．综合题（每小题20分，共60分）

1. 请结合实例说明什么是网络地址转换（NAT），其主要作用是什么？

网络地址转换（Network Address Translation，NAT）是一种网络协议转换技术，用于在不同网络之间转换IP地址。它的主要作用是解决IPv4地址短缺问题，并提供了一种有效的方式来连接私有网络与公共网络。

NAT的主要作用如下：

1. IP地址转换：NAT允许将私有IP地址转换为公共IP地址，以便在私有网络中的设备能够与公共网络进行通信。这样，私有网络中的设备可以共享有限的公共IP地址，从而解决了IPv4地址短缺的问题。

2. 网络隔离：通过使用NAT，私有网络中的设备可以使用非路由的私有IP地址，这些地址在公共互联网上是不可路由的。这提供了一定程度的网络隔离和安全性，使得私有网络中的设备对外部网络的直接访问受到限制。

3. 网络连接共享：NAT还可以实现多个设备共享单个公共IP地址的功能。通过将私有网络中的设备映射到不同的端口号，NAT可以将多个设备的通信路由到公共IP地址上，实现多个设备共享一个公共IP地址的连接。

下面是一个示例，说明NAT的工作原理：

假设有一个私有网络，其中有多台设备使用私有IP地址（如192.168.0.x）进行通信。当其中一台设备需要与公共互联网上的服务器进行通信时，NAT会将该设备的私有IP地址转换为公共IP地址，并在转换过程中修改IP头部的源IP地址和目标IP地址。这样，设备与服务器之间的通信就可以顺利进行。

1. 在程序中直接将读入的网络数据流转换为等长字节数组存在安全问题，因为在网络获取数据不流畅，数据流的传送会断断续续，不能保证一次能读取全部数据。特别是在读取大容量网络数据时问题更严重。假设有以下场景，应用程序servelet当前收到http（协议）请求流，请设计程序怎么保证当前程序接收到所有请求流数据？请写出主要代码和思路。

已知http请求流的获得指令：

ServletInputStream inStream = request.getInputStream(); //取HTTP请求流

int size = request.getContentLength(); //取HTTP请求流长度

参考思路：

在读取数据时检测实际读到的长度，如果没有读完已知长度的数据就应该再次读取，以此循环检测，直到实际读取的长度累加与已知的长度相等。

ServletInputStream inStream = request.getInputStream(); //取HTTP请求流

int size = request.getContentLength(); //取HTTP请求流长度

byte[] buffer = new byte[size]; //用于缓存每次读取的数据

byte[] in\_b = new byte[size]; //用于存放结果的数组

int count = 0;

int rbyte = 0;

while (count < size) { //循环读取

　rbyte = inStream.read(buffer); //每次实际读取长度存于rbyte中

　for(int i=0;i;)

　　in\_b[count + i] = buffer[i];

}

count += rbyte;

}

1. 关于URI与URL的区别，以下描述不正确的是（ C ）。
2. URI 类与URL 类相比，它更能适应相关的规范
3. URI 类只关心资源的标识和对URI的解析
4. 如果要获取内容，则使用URI;如果只是标识资源，则使用URL
5. URL 类在存放之前，就已经对所有的URI进行了“绝对化”的处理
6. 以下哪种编码方式经常会导致中文乱码（C）？
7. GBK
8. ASCII
9. UTF-8
10. GB2312
11. http服务器的响应码以“4”开头表示（ A ）。
12. 客户端错误
13. 服务器错误
14. 重定位或重定向
15. 响应成功
16. 下面关于URLConnection的内容，不正确的是（ B ）。
17. URLConnection可向服务器发送数据
18. URLConnection是URL的子类
19. URLConnection提供了对HTTP头信息访问
20. URLConnection可以配置发给服务器的参数
21. 下面关于DNS缓存，描述不正确的是（ B ）。
22. 修改域名指向可能需要几个小时才能生效
23. 按IP地址使用getByName查找，需要检查DNS
24. DNS查询可能需要几秒
25. 本机、本地DNS、其他DNS会缓存DNS查询
26. URL中的符号#的作用是（ D ）。
27. 传递参数
28. 标记一个新的网页地址
29. 清除缓存
30. 标记网页中的一个位置
31. 使用请求方法（ A ）需要设置 setDoOutput(true) 。
32. POST
33. HEAD
34. OPTIONS
35. GET
36. 下面关于GET 与 POST 的区别, 错误的是（ D ）。
37. get 是从服务器上获取数据，post 是向服务器传送数据
38. 在做数据查询时，建议用Get方式；而在做数据添加、修改或删除时，建议用Post方式
39. get 传送的数据量较小，不能大于2KB。post 传送的数据量较大，一般被默认为不受限制
40. get 安全性较高，post 安全性相对低。
41. 关于Cookie 与 Session的说法，错误的是（ C ）。
42. Cookie保存在客户端
43. Session在服务器端创建
44. Cookie在客户端创建
45. Session保存在服务器端
46. 通过getCanonicalHostName方法能得到（ D ）。
47. 域名
48. IP地址
49. IP地址和域名
50. IP地址或域名

二．判断题 （每小题2分，共10分，请给正确的打“√”，错误的打“×”，并填写题中括号内）。

1. InetAddress类的getLocalHost( )方法不会进行DNS查询，将直接返回主机名和回送地址。（ F ）
2. 不同服务器响应Http首部的属性数量是一样的 （ F ）
3. getHeaderFieldKey(int n), n=1时表示请求方法的字段名（ F ）
4. POST方法主要面向不能重复的不安全操作（ T ）
5. URI中必须使用ASCII字符（ F ）

试简析InetAddress类、URI类、URL类和URLConnection类的作用，分别在什么编程需求下可以选用？

InetAddress类、URI类、URL类和URLConnection类是Java中用于处理网络相关操作的类。它们在不同的编程需求下可以选用，具体作用如下：

1. InetAddress类：

- 作用：InetAddress类用于表示IP地址和主机名，并提供了与网络通信相关的方法。

- 编程需求：当需要获取主机名、IP地址、进行主机名解析、进行网络通信等操作时，可以选用InetAddress类。

2. URI类：

- 作用：URI类用于表示统一资源标识符（Uniform Resource Identifier），包括URL和URN等。

- 编程需求：当需要解析、构建和操作URI，获取URI的各个组成部分（如协议、主机、路径等）时，可以选用URI类。

3. URL类：

- 作用：URL类用于表示统一资源定位符（Uniform Resource Locator），即网络上的资源地址。

- 编程需求：当需要处理URL，包括解析URL、获取URL的各个组成部分（如协议、主机、路径等）、进行网络连接和数据传输等操作时，可以选用URL类。

4. URLConnection类：

- 作用：URLConnection类用于建立和管理与URL之间的连接，并提供了发送和接收HTTP请求和响应的功能。

- 编程需求：当需要发送HTTP请求、接收HTTP响应、设置请求头、获取响应头、处理Cookies等操作时，可以选用URLConnection类。

1. 客户端编程：创建一个简单的HTTP客户端程序：EasyHTTPClient 类，它访问www.szu.edu.cn/xxgk/xxjj.htm, 把得到的HTTP响应结果保存到本地文件系统的一个文件中。

文本

描述已自动生成

1. 关于缓存及属性相关的描述，不正确的是（ C ）。
2. 如果缓存没超过过期时间，则无需与服务器交互，直接使用
3. Etag首部用于资源改变时唯一的标识符
4. no-cache表示不能缓存
5. Last-modified表示资源最后一次修改的日期
6. 关于socket.setOOBInline的用法描述，不正确的是（ B ）。
7. 在flush方法前，能优先发送数据
8. 只需客户端设置，服务器端不需要设置
9. OOB 是 Out-of-band的缩写，发送具有特殊意义的紧急数据
10. 每次只能发送一个字节数据
11. 关于保护通信相关的描述，不正确的是（ A ）。
12. 实际应用中，使用对称加密传输非对称加密的密钥
13. 对于对称加密，加密方和解密方使用相同的密钥
14. 对于非对称加密，加密方和解密方使用不同的密钥
15. 为了避免中间人攻击，公开密钥放在可信任的第三方认证机构
16. 方法（ C ）用于判断远端服务器是否已经断开连接最合适。
17. isClosed
18. isConnected
19. sendUrgentData
20. isInputStreamShutdown
21. 为了防止客户端不知服务器已经崩溃，客户端可以采用方法（ B ）在空闲时持续检查服务器的响应情况。
22. setTcpNoDelay
23. setKeepAlive
24. setSoTimeout
25. setSoLinger
26. Buffer的flip() 方法的作用是（ A ）。
27. 将限度limit设置为当前位置position，位置position设置为0；
28. 将限度limit设置为capacity，位置position设置为0；
29. 将限度limit设置为0，位置position设置为当前位置position
30. 将限度limit设置为0，位置position设置为capacity；
31. 关于服务器日志的描述，正确的是（ C ）。
32. 日志可有可无
33. 一般不用记录服务器错误
34. 系统上线后，不应输出调试信息
35. 尽可能多的记录信息
36. 关于非阻塞I/O的优缺点描述，不正确的是（ B ）。
37. 能使单线程处理多个连接
38. 性能超过多线程+阻塞IO
39. 适用于很多连接，但每个连接不活跃的情况
40. 非阻塞IO处理连接是异步的
41. 对于安全Socket加密算法SSL\_DH\_anon\_EXPORT\_WITH\_DES40\_CBC\_SHA，DES40表示（ C ）。
42. 密码协商的方法
43. 密钥交换算法
44. 加密算法
45. 校验和
46. 下面函数（ B ）的作用是将 position 与 limit之间的数据复制到buffer的开始位置，并分别将limit –position，capacity赋值给position与limit。
47. reset
48. compact
49. rewind
50. flip

二．判断题 （每小题2分，共10分，请给正确的打“√”，错误的打“×”，并填写题中括号内）。

1. 使用Java.net.Socket可以直接创建一个安全socket对象（ F ）
2. 默认情况下，SocketChannel对象处于阻塞模式（ T ）
3. 在SSL协议中， 当一个通信端无须向对方证实自己的身份，就称它处于客户模式（ T ）
4. UDPSocket和TCPsocket的SO\_REUSEADDR选项作用相同（ F ）
5. 下列两段代码是等价的 （ F ）

//第一段

socket.shutdownInput()

socket.shutdownOutput()

//第二段

socket.close

三．综合题（每小题30分，共60分）

1. 请简述什么是中间人攻击，如何有效解决？

中间人攻击（Man-in-the-Middle Attack）是一种网络安全攻击方式，攻击者在通信的两端之间插入自己的设备或程序，以窃取、篡改或伪造通信数据。

中间人攻击的工作原理如下：

1. 攻击者通过某种方式，例如ARP欺骗或DNS欺骗，将自己的设备或程序插入到通信的两端之间，使得通信数据经过攻击者的设备或程序。

2. 当通信的两端进行数据传输时，攻击者可以拦截通信数据，并对其进行篡改、窃取或伪造。

3. 攻击者可以窃取敏感信息，如用户名、密码、信用卡信息等，用于进行身份盗窃或其他恶意活动。

4. 攻击者还可以篡改通信数据，以修改交易内容、劫持会话或传播恶意软件等。

为了防止中间人攻击，可以采取以下措施：

1. 使用加密通信：通过使用SSL/TLS等加密协议，可以确保通信数据在传输过程中是加密的，从而防止攻击者窃取敏感信息。

2. 验证证书：在使用HTTPS等加密通信时，要验证服务器的证书，确保通信的对方是合法的服务器，而不是攻击者伪造的。

3. 使用安全的网络环境：确保在进行敏感操作时，使用受信任的网络环境，避免使用公共无线网络等容易受到中间人攻击的网络。

4. 定期更新软件和系统：及时更新操作系统、浏览器和其他软件，以修复已知的安全漏洞，减少中间人攻击的风险。

1. 使用Java编程实现web缓存可以加快客户端访问web的相应速度，需要使用系统级缓存。如果在你编程实现的HTTP客户端需具有缓存功能，你该如何去编程实现，需要使用哪些类，需要自定义哪些类，试给出编程实现的例子程序，例子程序可以直接给出源代码及说明，也可只给出详细类图，并指出类图中每一个类的作用。

文本

描述已自动生成

文本

描述已自动生成

在上面的代码中，我们使用HashMap类来实现一个简单的缓存，将URL作为键，将HTTP响应作为值。在getCachedResponse方法中，我们从缓存中获取URL对应的响应。在cacheResponse方法中，我们将URL和响应存储到缓存中。

在main方法中，我们首先尝试从缓存中获取响应，如果缓存中不存在，则发送HTTP请求获取响应，并将将响应存储到缓存中。

通过这样的实现，当客户端再次请求相同的URL时，首先会尝试从缓存中获取响应，如果缓存中存在，则直接使用缓存的响应，从而加快客户端的访问速度。