

第13章 ネットワーク通信

(1) 次の空欄を埋めよ。

ネットワーク通信を簡単に実現するためのクラスが存在し、細かな制御は_____クラスで行ってくれるため、我々は簡単に Web ページの取得が行うことができる。

[java.net.URL]

(2) 次の空欄を埋めよ。

<java.net.URL クラスの一般的な利用手順>

①URL クラスを_____する。

```
URL url = new URL("http://www.yahoo.co.jp/");
```

[インスタンス化]

②openStream() を呼び出して、データを取り出す_____を取得する。

```
InputStream is = url.openStream();
```

[ストリーム]

③read() を呼び出して、ストリームから 1 バイトずつ情報を取り出す。

```
int data = is._____;
```

[read()]

(3) 次の空欄を埋めよ。

①read() を呼び出して、ストリームから最初の 1 バイトを取り出し、終わりか判定する

```
int c = isr.read();
```

```
while(c != _____) {
```

[-1]

```
System.out.print((char) c);
```

```
//read() を呼び出して、ストリームから次の 1 バイトを取り出す
```

```
c = isr.read();
```

```
}
```

```
//ストリームを閉じる
```

```
isr._____;
```

[close()]

J a v a II 第 13 章小テスト

(4) 次の空欄を埋めよ。

<Socket を用いた接続と切断の一般的な利用手順>

①IP アドレスまたはサーバー名と_____を指定して Socket をインスタンス化する。

```
Socket sock = new Socket("yahoo.com", 80);
```

[**ポート番号**]

②Socket から入カストリームと_____を取得する。

```
InputStream is = sock.getInputStream();
```

```
OutputStream os = sock.getOutputStream();
```

[**出力ストリーム**]

③2 つのストリームを読み書きする。

```
int data = is.read();
```

```
os._____ ("HELLO");
```

[**write**]

④ソケットを閉じる。(Socket の close() を呼べば、入力と出力の 2 つのストリームも自動的に閉じられる。

```
sock._____;
```

[**close()**]