## 第一章

**1、C 2、B 3、A 4、BD**

**5、CE 6、BD 7、AC 8、ABCD 9、D**

## 第二章

**1、C** **2、D** **3、B 4、BD 5、D 6、ABCD**  **7、A 8、B**

## 第三章

**1、ABCD 2、D 3、B 4、A 5、C**

**6、D 7、AB**

## 第四章

**1、B 2、DE** **3、C** **4、A** **5、D**

**6、C** **7、D** 8**、B**

## 第五章

**1、B** **2、A** **3、D** **4、C 5、ABCD**

**6、C** **7、C 8、AB**

## 第六章

**1、A** **2、D 3、C 4、B 5、A**

**6、B 7、AD**

## 第七章

**1、CD** **2、A 3、java.text** **4、SimpleDateFormat** **5、native2ascii**

**6、/^[\d]{1,2}$/**

**7、能不能直接通过new来创建某种类型的Class对象？为什么？**

**不能；类加载是指将类的class文件读入内存，并为之创建一个java.lang.Class对象，即当程序使用任何一个类时，系统都会为之创建一个java.lang.Class对象。**

**8、简述Class.forName的作用?为什么要用?**

**Class.forName的作用是在Java程序中获取Class对象；**

**每个类被加载之后，系统都会为该类生成一个对应的Class对象，通过Class对象就可以访问到JVM中该类的信息。一旦类被载入JVM中，同一个类将不会被再次载入。被载入JVM的类都有一个唯一标识，该标识是该类的全限定类名（包括包名和类名）。**

9、简述类在什么情况下被加载？

当程序主动使用某个类时，如果该类还未被加载到内存中，系统将通过加载、连接和初始化三个步骤对类进行初始化。如果没有意外出现，JVM会连续完成这三个步骤，因此将加载、连接和初始化三个步骤统称为类的加载。

类加载器通常无需在首次使用类时加载该类，JVM允许系统预先加载某些类。

10、简要描述一下Field、Method、Constructor的功能。

Field类用于封装属性的信息，调用Class对象的getFields()方法或getField()可以获取当前类的所有属性或指定属性。

Method类用于封装方法的信息，调用Class对象的getMethods()方法或getMethod()可以获取当前类的所有方法或指定方法。

Constructor类用于表示类的构造方法，通过调用Class对象的getConstructors()方法可以获取当前类的构造方法的集合。

## 第八章

**1、C** **2、A** **3、D 4、C** **5、ACD 6、B**