第六章

|  |  |
| --- | --- |
|  | 复习内容 |
| 6.1 类的基本概念 |  |
| 6.2 定义类 | 掌握类的结构  理解表6.1、表6.2、表6.3  了解成员变量与局部变量p83 |
| 6.3 对象的创建与使用 | 理解图6.3，了解表6.4，  掌握例题6.1, 例题6.2, 例题6.3  掌握this关键字p89 |
| 6.4 参数传递 | 掌握例题6.4 |
| 6.5 匿名对象 | 了解匿名对象的创建与使用 |

第七章

|  |  |
| --- | --- |
|  | 复习内容 |
| 7.1 类的私有成员与公有成员 | ？ |
| 7.2 方法的重载 | 掌握方法重载的概念，掌握例题7.3 |
| 7.3 构造方法 | 理解构造方法的作用与定义，掌握例题7.4  掌握构造方法的特殊性p100  掌握默认构造方法 p101  理解构造方法重载，掌握例题7.5  掌握从一个构造方法调用另一个构造方法 理解例题7.6 |
| 7.4 静态成员 | ？ |
| 7.5 对象的应用 | ？ |

|  |  |
| --- | --- |
| 答案A | 1. 以下关于构造方法的描述错误的是( A )。 2. 构造方法的返回值类型只能是void型 3. 构造方法是类的一种特殊方法，它的方法名必须与类名相同 4. 构造方法的主要作用是完成对象的成员变量的初始化工作 5. 用new关键字可以调用构造方法 6. 以下关于方法的形式参数的说法中正确的是( A )。 7. 方法的形式参数是方法的局部变量，但不能在定义时赋默认值 8. 若多个形式参数的类型相同，则可以只写一个类型说明符 9. 方法若没有形式参数，则方法名后的()可以省略，或者在()内写void 10. 方法的形式参数不可以和成员变量同名 //见p90-91 11. 以下关于方法的参数的说法中错误的是( A )。 12. 类的方法的形式参数可以是各种类型，但不能与该类相同 //见p112例题7.11-行号12 13. 调用方法时的实际参数可以是变量、常量和表达式，也可以是其他方法的返回值 14. 调用方法时将实际参数的值赋给形式参数 15. 在方法体内给形式参数赋值不会影响实际参数的值 16. 关于一个有返回值的方法，以下说法不正确的是( A )。 17. 方法体里不能出现一条以上return语句 //见p113例题7.13-行号14，行号16 18. 即使有return语句，也可能有编译错误 19. return后只能是一个常量、变量或表达式 20. 方法体里只能有一条return语句被执行 21. 以下关于构造方法重载的描述中，错误的是(A )。//要符合构造方法+方法重载（p98概念） 22. 构造方法是在创建对象时被自动调用的，只能产生一个实例对象，所以不能重载 23. 如果希望可以用多种方式创建类的对象，可以重载构造方法 24. 重载的构造方法应该具有不同的参数列表 25. 没有参数的构造方法和有参数的构造方法，可以构成重载 26. 以下关于方法调用的说法中错误的是( A )。   //说明：实际参数的类型可以与形式参数的类型兼容，  //例如，int型的实际参数可以传递给double 型的形式参数。   1. 调用有形式参数的方法必须按照形式参数的类型、个数和顺序提供实际参数，且实际参数的类型只能和形式参数的类型完全一致 2. 调用没有形式参数的方法不能提供实际参数，但方法后的()不能省略 3. 一个方法调用同类中构造方法之外其他方法的格式为"方法名(实际参数列表)" 4. 访问一个对象的成员方法的格式为"对象名.方法名(实际参数列表)" 5. 以下关于方法调用的说法中错误的是( A )。 6. 调用有返回值的方法不能独立作为一条语句 7. 调用没有返回值的方法只能独立作为一条语句 8. 执行一个方法调用时程序控制流转向该方法 9. 执行一个方法调用结束时程序控制流转回方法调用者 |
|  | 以下答案选B |
| 答案B | 1. 以下关于成员变量的说法中错误的是( B )。 2. 成员变量描述对象的属性 3. 成员变量只能是基本类型//见例题7.3 行号7（private String color;） 4. 成员变量可有可无 5. 成员变量在定义时可以初始化//见例题7.3 行号6（private double pi=3.14;） 6. this关键字的含义是表示(B )。 7. 指针 8. 当前对象 / 9. 当前类 10. 当前方法 11. 构造方法在( B )时被调用。 12. 类定义时 13. 创建对象时 14. 调用对象方法时 15. 使用对象的变量时 16. 以下关于默认构造方法的描述错误的是( B ) // 见p101 17. 如果类定义的构造方法都是有参数的，则类没有默认构造方法   //指在这种情况下，系统不再提供默认构造方法   1. 如果类没有定义任何构造方法，则一定不能创建类的对象 2. 通常定义默认构造方法以提高创建对象的灵活性 3. 默认构造方法没有参数，方法体可以为空 4. 以下关于构造方法重载的描述中，错误的是( B )。 5. 一个构造方法调用同类其他构造方法使用"this(实际参数列表)"而不是方法名   //见p103 this（）语句的用法   1. 一个构造方法不可以调用同类其他构造方法 2. 一个构造方法对同类另一个构造方法的调用必须是方法体中第一条语句 3. 除了带参数的构造方法，通常还需要定义没有参数的默认构造方法   // 见p122例题8.3 行号6   1. 以下关于成员方法的说法中错误的是( B )。//见p82成员方法的定义及说明 2. 成员方法描述对象的行为 3. 成员方法的返回值和形式参数只能是基本类型 4. 成员方法可有可无 5. 除了构造方法之外的成员方法都要有返回值类型说明符 6. 以下关于方法返回值的说法中正确的是( B )。 7. 如果方法没有返回值，则返回值类型说明符可以省略 8. 方法的返回值必须和返回值类型(方法类型)一致，或者能够进行自动类型转换 9. 返回值类型为void的方法不能有return语句 10. 构造方法的返回值类型可以省略也可以写void 11. 下列关于作用域的说法中，正确的一个是( B )。 12. 一个方法可以访问另外一个方法中定义的变量 //局部变量只能在定义它的方法中被使用 13. 不同方法中可以声明相同类型的同名局部变量 14. 在一个类中可以声明不同类型的同名成员变量 15. 在不同语句块中不能声明相同类型的同名变量 16. 以下关于main()方法的说法中错误的是( B )。 17. main()也是类的成员方法，但不能被其他方法调用 18. main()方法中定义的变量都是类的成员变量 //方法中定义的变量都是局部变量 19. 一个类可以只有main()方法而没有其他方法或变量 20. main()方法没有返回值 |
|  | 以下答案选C |
| 答案C | 1. 以下关于构造方法的描述错误的是( C )。 2. 类中可以定义0个或多个构造方法 //见例题6.1，例题7.5，7.6 3. 如果在一个类中没有定义构造方法，则系统一般会提供默认构造方法 //见例题6.1 4. 默认构造方法至少要包含一个参数 //见p101中间 5. 如果在类中定义了构造方法，则系统不再生成默认构造方法 6. 当成员变量与局部变量同名时，若要在方法内使用成员变量，要使用关键字( )。//p89 this的用法 7. super 8. import 9. this 10. return 11. 语句fun(1, 3.14);调用的不可能是同类中的方法( C )。 12. public int fun(int a, double b){…} 13. public void fun(double a, double b){…} 14. public void fun(int a, char b){…} 15. char fun(long a, double b){…} |
|  | 以下答案选D |
| 答案D | 1. 以下关于类和对象说法中错误的是( D )。 2. 编写类就是自定义一种数据类型，对象是这种类型的数据 3. 类是抽象的，对象是具体的、类的实例 4. 一个类的对象名就是这种类类型的变量 5. 数组不是对象而是基本类型变量   //见62 数组名是引用类型的变量，用来保存数组在堆内存中的首地址   1. 以下关于类和对象关系的描述中，正确的是( D )。 2. 同一个类只能创建一个对象，如果创建多个对象，则编译出错 3. 同一个类可以创建多个对象，但多个对象共享同一个内存空间 4. 同一个类只能创建一个对象，因为类是同一类事物的抽象描述 5. 同一个类可以创建多个对象，这些对象被分配不同的内存空间 6. 对于类的说法中，不正确的一项是( D )。 7. 类的属性可以包括常量、变量和数组等 8. 类中的每个方法都由方法头和方法体构成 // 除去抽象方法 9. 一个.java源文件中可以定义多个类，但是公共类至多有一个 10. 一个.java源文件中可以定义多个公共类 //只能有一个public类，见p16页 11. 类的成员的访问限定符不包括( D )。 12. public 13. private 14. protected 15. int 16. 类的访问限定符符可以是( D )。 17. public 18. 默认(无限定符) 19. private 20. A和B都可以 21. new操作符的作用不包括( D )。 22. 为对象分配内存空间 23. 调用类的构造方法 24. 为对象返回一个引用 25. 定义一个类 26. 语句int t=a.fun(1, 3.14);调用的可能是方法( D )。 27. public String fun(int a, double b){…} 28. public void fun(double a, double b){…} 29. public int fun(int a, char b){…} 30. char fun(long a, double b){…}  * **数值型不同类型数据的转换**   + **自动类型转换**   **①转换前的数据类型与转换后的类型兼容。**  **②转换后的数据类型的表示范围比转换前的类型大。**  **条件②说明不同类型的数据进行运算时，需先转换为同一类型，然后进行运算。转换从“短”到“长”的优先关系为：byte→short→char→int→long→float→double**   1. 关于以下类，说法错误的是( D )。   class A {  int a;  void test(int t) {  double p = t \* a;  }  }   1. a是成员变量 2. p是局部变量 3. test()方法外无法访问变量t //见p83局部变量t只能在定义它的方法内使用 4. test()方法内的a不可以写作this.a //见p91，只有特指成员变量时，可以用this关键字。 |