本科教学工程项目

湖北经济学院国际化课程（培育）

Java程序设计

年

度

报

告

项目名称： 国际化课程培育

学 院： 信息工程学院

负 责 人： 关培超

时 间： 2016年12月

# 项目的执行情况

湖北经济学院理工科专业立足于培养应用型的复合人才。Java程序设计是软件类专业的重要必修课程，也是其他理工科专业的重要专业选修课程，同时本课程对专业的后续课程有良好的支持作用，也是后续Java EE课程的先行课。随着国际化信息类相关课程的不断开展，非理工科专业学生的也将Java程序设计的学习列为重点（我校中美合作办学的信息管理与信息系统专业规划中，Java占108课时，二个阶段完成）。2016年下半年，学校通识选修Java程序设计也第一次被选上开课，说明该课程对非专业学生也具备一定吸引力和使用性。

课程在专业培养计划中定位于学科专业基础技能课程。所以课程建设目标在于系统地向学生讲述Java面向对象的编程思想、使学生可熟练运用面向对象技术，可以使用Java编写简单的各种类型的面向对象的应用程序，培养解决实际问题的能力，培养专业学生和非理工科学生的综合素质、技术能力、学习热情。

Java程序设计课程在去年进入国际化课程培育阶段，目前项目各个建设环节按照去年规划正常进行，目前主要完成的工作包括课程教学大纲中英文（各类），部分英文课程试题库，部分英文教学课件。结合课程性质，培养学生动手实践能力，目前已集合了部分课程主讲教师和大三该课程专业学生，开始了课程教学网站也在设计建设。

# 项目建设取得的标志性成果

本项目近一年取得的标准性成果包括（详见第六部分附件）：

1、课程建设教师，在该门课程基础上指导专业学生进行的实践性Java全国竞赛中取得了不错的成绩。课程结束后、学生应可达到在此基础上进行扩展式的学习，大部分可以以Java技术基础进入软件企业的学生，可以获得良好评价和广泛的认同。

2、2016年截止到12月份，已结合现有课程实际教学，完成了分专业，分课程性质的课程中文大纲、构建了部分习题库、课程实验案例库（见附件），具体完成的任务如下：

（1）9月份前，完成对应中文教学大纲、实验大纲的英文翻译工作，完成对应英文习题库、课件的初步构建工作。

（2）12月底之前，完成中文Java课程教学过程的基本整理，并收集现有班级学生对Java课程教学内容、教学过程的反馈意见，为明年确定Java整个教学过程打下基础。

3、11月份开始，针对Java程序设计教学特点，开始Java课程双语教学网站的设计与具体建设。目前完成的部分主要是根据现有资料和学生课程过程中的主要重难点，进行中文网站建设工作。

# 项目建设主要经验、成效及示范带动作用

经过一年的课程培育建设，Java程序设计国际化课程相关的四位专职教师已经形成固定的教学团队，通过日常教研活动以及课程教学经验的交流，对课程建设已经达成初步共识，相关建设经验如下：

1、课程教学一定要根据学生具体情况制定教学计划与内容体系，Java程序设计本身教学内容多，对于没有任何编程经验的学生学习难度大（一些有兴趣编程的学生，在没有C语言基本编程锻炼的前提下，没有面向对象编程思想的情况下就选修了该门课程），更要根据学生具体情况，选择难度不大，好入门的内容进行教学过程。

2、课程专业学生可以进行传递性教学，专业学生一般具备专业前面编程基础，对Java课程体系在专业培养过程中的地位和相关课程的前后级关系比较清楚，由于同在信息工程学院，各个年级之间的传帮带，有利于课程学习的延续性（从专业技能上看，Java学习应该涉及、贯穿整个专业学习至少2年）。目前专业高年级学生（大三学生）已经可以由个别学习能力强，表达能力高的学生，在暑期利用IT协会活动给低年级学生传授Java学习经验，整个课程学习已经从专业课堂扩展到课外，形成了较好的良性循环。

# 项目建设存在的问题及对策

从目前项目建设情况来看，主要存在以下问题：

1、专业课程学生课程的学习时间过少，54个学时不能完全把Java程序设计的重点部分讲述清楚，真正学习好的同学都是平时在老师引导下，自觉学习的课后知识点，大量编码的同学，而不习惯自学，不喜欢问问题的学生一般只能完成课程最基本要求。

2、对非专业学生的课堂讲授还是存在课时少，教学难度大的问题。从本学期Java程序设计第一次A类选修课的教学情况看，学生的层次参差不齐，部分偏理工类学生，有编码基础的学生学习过程中吃不饱，只能靠教师课后单独引导或传授。而部分完全没有编码基础的学生反映课程难度大，基本的代码都无法理解，主要原因还是在于这些学生的学习习惯造成（一些非理工学生连基本的英文打字都无法过关，学习编程思想不灵活）

2、课程实验时间过少，Java程序设计，无论是必修还是选修，都存在这个问题，课程本身的实验性质决定了。

# 总结反思与建议

结合上述项目建设过程中存在的问题，目前课程组各位教师经过不断研讨，提出建议如下：

1、对于必修课程，可以尝试加大课时量，特别是上实验的课时量。

2、对于非专业的学生，必须设置一个基本的门槛，限制完全没有基础的学生选修该门课程。

# 附件

**1、课程英文大纲（部分）**

**Information Engineering College, Java Programming**

**I．Basic Course Information**

Course Name: Java Programming Course Code: ZB5215

Course Type: Major Required Course Credits: 3

Hours: 54 Theoretical Hours: 30 Experimental Practice Hours: 24

Object - oriented: Computer and related majors

Prerequisites: Introduction to Computers, Advanced Language Programming, Data Structures

**II．The purpose and requirement of course teaching**

This course is a compulsory course for computer and related majors. It mainly introduces the basic knowledge of Java programming, Java object-oriented programming principle and application, Java SE core application module and API document. This course is the computer introductory, high-level language program design courses follow-up courses for computer and related majors in Java technology direction lay the foundation for learning.

After this course, students are required to have the following knowledge and skills:

1．Understand the characteristics of Java language and application range, can use the mainstream Java development environment for Java software project design and development.

2．Master Java basic grammar norms, constants, variables, operators, expressions, statements and so on.

3．Master Java object-oriented programming methods, classes, objects, interfaces, inheritance and so on

4．Master the common data structure of Java implementation, familiar with the array, string, collection, basic input and output, packaging and other commonly used structure or the use of commonly used class.

5．Understand the exception handling in Java, skilled use of common Java exception class, to understand the custom exception class;

6．Understand the Java graphical interface programming, according to practical problems using Java design to achieve a simple graphical interface program.

7．Understand the basic methods of Java database programming, familiar with the basic principles of JDBC and use.

8．Understand the basic idea of Java network programming, according to practical problems using Java network programming technology.

9．Understand the Java thread processing mechanism, can be simple to write multi-threaded Java program to solve practical problems.

Through this course, students should be able to master the above listed Java technology, based on a certain degree of programming, coding and debugging troubleshooting capabilities, can use Java graphical interface, database, network, multi-threaded technology integration Application and design of small Java applications, according to the actual situation design and programming procedures to solve the problem.

**III．Course Examination Requirements**

The course of assessment methods currently closed-book examination in the form of conditions in the case can be used on the form of machine examination. The test questions focus on examining students' mastery of basic knowledge and comprehensive analysis and application ability. Assessment is divided into the process of assessment and completion assessment, the overall results by the two parts, the process of examination results accounted for 30%, 70% completion examination results. Process assessment results can be based on time and attendance, operations, laboratory reports, tests, classroom discussions and on the results of the form of comprehensive evaluation and other forms of acceptance. All scores are given in percentile

**IV．Basic Contents of Teaching, Distribution of Hours and Arrangement of Teaching**

"Java programming" hours allocation

|  |  |  |
| --- | --- | --- |
| Content | Theory Hours | Experiment (Practice) Hours |
| Chapter 1 Overview of Java | 2 |  |
| Chapter 2 Data Types, Expressions and Control Structures | 2 |  |
| The third chapter Java object-oriented implementation | 4 | 4 |
| The fourth chapter commonly used to use | 4 | 4 |
| Exception handling of Java of the 5th chapter | 2 | 2 |
| Chapter 6 Java graphical interface design | 4 | 4 |
| Chapter 7 Java Database Programming | 4 | 4 |
| Chapter 8 Java Network Programming | 2 | 2 |
| Chapter 9 Java multithreading mechanism | 2 | 2 |
| The 10th chapter Java case synthesis design | 4 | 2 |
| total | 30 | 24 |
|  |  |  |

**Chapter 1 Overview of Java**

The purpose and requirements of this chapter: to enable students to understand the historical background of Java, features and application scope, understand JDK, JVM and other related concepts, and from a simple Java program to enable students to master the Java program development steps and development tools.

This chapter focuses on teaching: Java program structure, Java program development steps

This chapter is difficult to teach: Java program development steps

**Section 1 Java history, status and application**

**Section 2 features of the Java language**

**Section 3 Java program development steps**

(I) JDK, JRE installation

(II) Development environment configuration and tool installation

(III) The first Java program and structure

(IV) Program compilation, compilation and execution

**Chapter 2 Data Types, Expressions and Control Structures**

Objectives and requirements of this chapter: To enable students to understand the basic data types of Java, the basic concepts of constants and variables, integer and real data encoding and computing, and common operators, expressions, basic input and output statements. Learn about the basic control structures such as if, switch, for, while, do ... while in Java, and the use of keywords such as break, continue, and so on.

This chapter focuses on teaching: data types, the use of control structures

This chapter difficult teaching: the expression, the basic input and output statements, some control statements

**Section 1 Data Types**

(I) Numeric types

(II) The character type

(III) Boolean type

(IV) Reference type

**Section 2 Constants and Variables**

(I) Constants are defined and used

(II) Variables, variable definition and initialization

**Section 3 Operators, Expressions, Statements, and Blocks**

**Section 4 basic input and output and type conversion**

**Section 5 Select the structure**

(I) Standard if and minimal if statements, nested selection

(II) Multi-branch switch structure

**Section 6 cycle structure**

(I) Standard for loop, for loop traversal

(II) The standard while loop, do ... while loop

(III) Loop control statements break and continue

**Chapter 3 Java object-oriented implementation**

Objectives and requirements of this chapter: Students understand the basic principles of object-oriented programming theory, the concepts of classes and objects, the syntax of classes, object references, properties and methods, constructors, class variables and class methods, overloading, access control, packages . Understand and master the principles of interface and inheritance in object-oriented programming, master the basic syntax of interface and inheritance in Java, and be familiar with the concept and meaning of polymorphism and coverage.

The teaching focus of this chapter: class and object concepts and syntax composition, inheritance and interface concepts and basic grammar

Difficulty in teaching in this chapter: Concepts and syntax of classes and objects, inheritance and interface Application principles, polymorphism, and coverage

**Section 1 Classes and Objects**

(I) Object - Oriented Programming Theory

(II) The basic concepts of classes and objects

(III) Object - Oriented Programming Example

**Section 2 of the basic structure**

(I) Class declaration

(II) Member properties, and member methods

(III) Construction method

(IV) Method overloading

(V) package

**Section 3 of the object creation and reference**

**Section 4 Inheritance Relationship**

(I) The concept of inheritance, the extends keyword

(II) The keywords this and super

(III) Package and class access control

**Section 5 Interface**

(I) The interface concept and the implements keyword

(II) Polymorphism, the interface as a type

**Chapter 4 Commonly used to use**

The purpose and requirements of this chapter: to enable students to master the basic API class of basic use, including System, String, Math, packaging and so on.

This chapter focuses on teaching: all

Teaching difficulties in this chapter: string, packaging class

**Section 1** **System**

**Section 2 String**

**Section 3 Math**

**Section 4 Packaging category**

(I) The concept of packaging classes

(II) Wrapper classes, and primitive data types

**Chapter 5 Exception handling of Java**

The purpose and requirements of this chapter: to enable students to understand the basic concepts of exceptions, understanding of Java exception handling mechanism, familiar with the common Java exception class, learn to use try ... catch ... finally structure to catch exceptions, and actively throw an exception .

This chapter focuses on teaching: all

Teaching difficulties in this chapter: exception handling mechanism, abnormal catch and throw

**Section 1 The first section of the concept of anomaly, type and processing mechanism**

**Section 2 abnormal object**

Throwable and its subclasses

**Section 3 exception handling**

(I) Methods and procedures for capturing anomalies

(II) Throws an exception and a custom exception

**Chapter 6 Java graphical interface design**

The purpose and requirements of this chapter: Use students to understand the awt and swing graphical interface design framework, familiar with Java graphical interface design components and related classes, layout manager, and GUI event handling.

Teaching Difficulties in This Chapter: Event Handling

**Section 1 Java GUI design framework**

(I) swing and awt

(II) Component

(III) Layout and layout manager

(IV) Event handling and listener

**Section 2 Silicon Components**

**Section 3 Layout Manager**

**Section 4 Event Handling and Listeners**

**Chapter 7 Java Database Programming**

This chapter aims and requirements: to enable students to understand the basic concept of JDBC, you can use Java to connect to the mainstream database, and can control the database for basic add, delete, change, check and other operations.

This chapter focuses on teaching: JDBC concepts and Java basic control methods

Difficulties in teaching this chapter: Java database basic operations

**Section 1 Overview of JDBC**

**Section 2 relational database**

**Section 3 of the database connection**

**Section 4 of the basic control of the relational database**

**Chapter 8 Java Network Programming**

The purpose and requirements of this chapter: understanding of network IP, port and other related concepts and the use of Java programming to understand the basic principles of Socket Socket in Java, you can use Socket to write a simple network program.

This chapter focuses on teaching: the use of network-related Java classes, Socket programming the basic process.

Teaching difficulties in this chapter: synchronization of network programs.

**Section 1 IP and port concept**

**Section 2 the network-related Java class**

**Section 3 Java network program**

(I) The Fundamentals of Socket

(II) Server and client

(III) Analysis of Network Program

**Chapter 9 Java multithreading mechanism**

The objectives and requirements of this chapter: to enable students to master the basic concepts of Java concurrency and thread control to master the basic method of creating threads to understand the thread control problems, the use of API method for simple thread scheduling.

This chapter focuses on teaching: thread-related concepts and thread creation

This chapter difficult teaching: thread scheduling

**Section 1 Concurrency and threading**

(I) Thread and process concepts

(II) The concept of multithreading in a program

**Section 2 Create and run threads**

(I) Inherits the Thread class

(II) Implement the Runnable interface

(III) Method of running threads

**Section 3 thread life cycle**

**Section 4 thread scheduling**

(I) Multi-threading, sleep and join use

(II) Priority scheduling, the use of synchronized

**Chapter 10 Java case synthesis design**

The purpose and requirements of this chapter: This chapter is a supplement to the chapter, teachers have a basic knowledge of the basis of Java and the actual coding capabilities, design cases to guide students to achieve a specific Java project, requiring students to comprehensive application of the sixth to ninth chapters Java related technologies.

This chapter focuses on teaching: the use of related Java class linkage.

Teaching difficulties in this chapter: Java multiple classes of synchronous design and implementation.

**V．Curriculum Guidance and Suggestions**

The teaching content of Java Programming is based on Java SE, which is based on Java language foundation, Java object-oriented implementation, the use of common Java class libraries and the application of Java in specific technical direction. Teaching methods mainly to multimedia presentation and a small amount of blackboard writing combination of teaching methods, teaching effectiveness and efficiency can be greater protection. Experimental teaching should be appropriately increased, with the experimental environment using live webcast and individual counseling combined with the way to strengthen the students understanding of the core knowledge and use.

Suggest that students in the course of this course, pre-class should be well prepared and found that problems in the classroom to listen to relevant content and actively communicate with teachers to solve, after the completion of homework and on-machine practice, master the relevant Knowledge and skills to improve practical ability, but also should read the relevant extra-curricular reference materials to enhance the comprehensive application of Java capabilities.

**VI．Recommended textbooks and reading lists**

1. "Introduction to Java Foundation," Chuanzhi Podcast Higher Education Product Research Department, Tsinghua University Press, 2014.

2. "Java language programming", phase clean Hok Ke-You, the People's Posts and Telecommunications Press, 2013

3. "Java Foundation Tutorial," Geng Xiangyi edited, Tsinghua University Press, 2012.

4. "Java How to Program" by Paul Deitel. Harvey Deitel, Prentice Hall, 2012

5. "Beginning Java 7", by Jeff Friesen, Apress, 2011

6. "Java programming thinking", edited by Bruce Eckel, Chen Haopeng translation, Mechanical Industry Press, 2012.

**2、课程英文课件（部分）**





















**3、课程试题库（部分）**

**Ⅰ. The Java Basic Grammar & Data Structure**

1.Which of the following lines will compile without warning or error.   D

A. float f=1.3;

B. char c="a";

C. byte b=257;

D. int i=10;

2.Which of the following are keywords or reserved words in Java?   D

A. if

B. then

C. goto

D. while

3.If you run the code below, what gets printed out?   B

 String s=new String("Bicycle");

  int iBegin=1;

 char iEnd=3;

 System.out.println(s.substring(iBegin,iEnd));

A. Bic

B. ic

C. icy

D. error: no method matching substring(int,char)

4. What will happen when you compile and run the following code? D

public class MyClass{

static int i;

public static void main(String argv[]){

System.out.println(i);

}

}

A. Error Variable i may not have been initialized

B. null

C. 1

D. 0

5. What will happen if you try to compile and run the following code?   C

public class Q {

public static void main(String argv[]){

int anar[]=new int[]{1,2,3};

 System.out.println(anar[1]);

}

}

A. 1

B. Error anar is referenced before it is initialized

C. 2

D. Error: size of array must be defined

6. What will be printed out if you attempt to compile and run the following code ?   C

int i=1;

 switch (i) {

case 0:

 System.out.println("zero");

break;

 case 1:

System.out.println("one");

 case 2:

System.out.println("two");

 default:

 System.out.println("default");

}

A. one

B. one, default

C. one, two, default

 D. default

7. What will be printed out if you attempt to compile and run the following code?   B

int i=9;

switch (i) {

default:

System.out.println("default");

case 0:

System.out.println("zero");

break;

 case 1:

System.out.println("one");

case 2:

System.out.println("two"); }

A. default

B. default, zero

C. error default clause not defined

D. no output displayed

8. Which of the following lines of code will compile without error B C

A. int i=0;

 if(i) {

System.out.println("Hello");

 }

B. boolean b=true;

boolean b2=true;

if(b==bB. {

System.out.println("So true");

}

C. int i=1;

 int j=2;

if(i==1|| j==B.

System.out.println("OK");

D. int i=1;

int j=2;

f(i==1 &| j==B.

System.out.println("OK");

**Ⅱ. Basic Mechanisms Of The OOP Paradigm**

1.What will happen if you try to compile and run the following code   A

public class MyClass {

 public static void main(String arguments[]) {

amethod(arguments);

 }

public void amethod(String[] arguments) {

System.out.println(arguments);

System.out.println(arguments[1]);

}

A. error Can't make static reference to void amethod.

B. error method main not correct

C. error array must include parameter

D. amethod must be declared with String

2. What will be the result of attempting to compile and run the following code?   C

abstract class MineBase {

abstract void amethod();

static int i;

 }

public class Mine extends MineBase {

 public static void main(String argv[]){

int[] ar=new int[5];

 for(i=0;i < ar.length;i++)

System.out.println(ar[i]);

 }

}

A. a sequence of 5 0's will be printed

B. Error: ar is used before it is initialized

C. Error Mine must be declared abstract

D. IndexOutOfBoundes Error

3. What will happen if you attempt to compile and run the following code?   C

 class Base {}

 class Sub extends Base {}

 class Sub2 extends Base {}

 public class CEx{

 public static void main(String argv[]){

 Base b=new Base();

 Sub s=(Sub) b;

  }

}

A. Compile and run without error

B. Compile time Exception

C. Runtime Exception

4.What will happen if you attempt to compile and run the following code?   C

 Integer ten=new Integer(10);

 Long nine=new Long (9);

 System.out.println(ten + nine); int i=1;

 System.out.println(i + ten);

A. 19 followed by 20

B. 19 followed by 11

C. Compile time error

D. 10 followed by 1

5. What will happen when you attempt to compile and run the following code?.  C

 class Background implements Runnable{

 int i=0;

public int run(){

while(true){

 i++;

System.out.println("i="+i);

 } //End while

return 1;

 }//End run

}//End class

A. It will compile and the run method will print out the increasing value of i.

B. It will compile and calling start will print out the increasing value of i.

C. The code will cause an error at compile time.

D. Compilation will cause an error because while cannot take a parameter of true.

6. Given the following class definition, which of the following methods could be legally placed after the comment with the commented word "//Here"? A D

public class Rid{

 public void amethod(int i, String s){}

 //Here

}

A. public void amethod(String s, int i){}

B. public int amethod(int i, String s){}

C. public void amethod(int i, String mystring){}

D. public void Amethod(int i, String s) {}

7. Given the following class definition which of the following can be legally placed after the comment line D

//Here ?

class Base{

public Base(int i){}

}

public class MyOver extends Base{

public static void main(String arg[]){

 MyOver m = new MyOver(10);

 }

 MyOver(int i){

 super(i);

 }

 MyOver(String s, int i){

 this(i);

 //Here

 }

}

A. MyOver m = new MyOver();

B. super();

C. this("Hello",10);

D. Base b = new Base(10);

8. What will happen when you attempt to compile and run the following code? A

class Base{

 public void amethod(int i, String s){

 System.out.println("Base amethod");

 }

 Base(){

 System.out.println("Base Constructor");

 }

}

public class Child extends Base{

int i;

String Parm="Hello";

public static void main(String argv[]){

 Child c = new Child();

 c.amethod();

}

void amethod(int i, String Parm){

 super.amethod(i,Parm);

 }

public void amethod(){}

}

A. Compile time error

B. Error caused by illegal syntax super.amethod(i,Parm)

C. Output of "Base Constructor"

D. Error caused by incorrect parameter names in call to super.amethod

**Ⅲ. Flow Control & Exception Handling**

1. What will happen when you attempt to compile and run the following code? D

public class MyIf{

boolean b;

public static void main(String argv[]){

 MyIf mi = new MyIf();

}

MyIf(){

 if(b){

 System.out.println("The value of b was true");

 }

 else{

 System.out.println("The value of b was false");

 }

 }

}

A. Compile time error variable b was not initialised

B. Compile time error the parameter to the if operator must evaluate to a boolean

C. Compile time error, cannot simultaneously create and assign value for boolean value

D. Compilation and run with output of false

2.What will happen when you attempt to compile and run this code? D

public class MyIf{

public static void main(String argv[]){

 MyIf mi = new MyIf();

 }

MyIf(){

 boolean b = false;

 if(b=false){

 System.out.println("The value of b is"+b);

 }

 }

}

A. Run time error, a boolean cannot be appended using the + operator

B. Compile time error the parameter to the if operator must evaluate

 to a boolean

C. Compile time error, cannot simultaneously create and assign value for boolean

 value

D. Compilation and run with no output

3.What will happen when you attempt to compile and run this code? D

public class MySwitch{

public static void main(String argv[]){

 MySwitch ms= new MySwitch();

 ms.amethod();

 }

public void amethod(){

 char k=10;

 switch(k){

 default:

 System.out.println("This is the default output");

 break;

 case 10:

 System.out.println("ten");

 break;

 case 20:

 System.out.println("twenty");

 break;

 }

 }

}

A. None of these options

B. Compile time errror target of switch must be an integral type

C. Compile and run with output "This is the default output"

D. Compile and run with output "ten"

4.What will happen when you attempt to compile and run the following code? A

public class MySwitch{

public static void main(String argv[]){

 MySwitch ms= new MySwitch();

 ms.amethod();

 }

public void amethod(){

 int k=10;

 switch(k){

 default: //Put the default at the bottom, not here

 System.out.println("This is the default output");

 break;

 case 10:

 System.out.println("ten");

 case 20:

 System.out.println("twenty");

 break;

 }

 }

}

A. None of these options

B. Compile time errror target of switch must be an integral type

C. Compile and run with output "This is the default output"

D. Compile and run with output "ten"

5.What will happen when you attempt to compile and run the following code? D

import java.io.\*;

class Base{

public static void amethod()throws FileNotFoundException{}

}

public class ExcepDemo extends Base{

public static void main(String argv[]){

 ExcepDemo e = new ExcepDemo();

}

public static void amethod(){}

protected ExcepDemo(){

 try{

 DataInputStream din = new DataInputStream(System.in);

 System.out.println("Pausing");

 din.readChar();

 System.out.println("Continuing");

 this.amethod();

 }catch(IOException ioe) {}

 }

}

A. Compile time error caused by protected constructor

B. Compile time error caused by amethod not declaring Exception

C. Runtime error caused by amethod not declaring Exception

D. Compile and run with output of "Pausing" and "Continuing" after a key is hit

6. What will happen when you attempt to compile and run the following code? D

import java.io.\*;

class Base{

public static void amethod()throws FileNotFoundException{}

}

public class ExcepDemo extends Base{

public static void main(String argv[]){

 ExcepDemo e = new ExcepDemo();

}

public static void amethod(int i)throws IOException{}

private ExcepDemo(){

 try{

 DataInputStream din = new DataInputStream(System.in);

 System.out.println("Pausing");

 din.readChar();

 System.out.println("Continuing");

 this.amethod();

 }catch(IOException ioe) {}

 }

}

A. Compile error caused by private constructor

B. Compile error caused by amethod declaring Exception not in base version

C. Runtime error caused by amethod declaring Exception not in base version

D. Compile and run with output of "Pausing" and "Continuing" after a key is hit

7.What will happen when you attempt to compile and run the following code? C

import java.io.\*;

class Base{

public static void amethod()throws FileNotFoundException{}

}

public class ExcepDemo extends Base{

public static void main(String argv[]){

 ExcepDemo e = new ExcepDemo();

}

public boolean amethod(int i){

 try{

 DataInputStream din = new DataInputStream(System.in);

 System.out.println("Pausing");

 din.readChar();

 System.out.println("Continuing");

 this.amethod();

 return true;

 }catch(IOException ioe) {}

 finally{

 System.out.println("Doing finally");

 }

 return false;

 }

 ExcepDemo(){

 amethod(99);

 }

}

A. Compile time error amethod does not throw FileNotFoundException

B. Compile, run and output of Pausing and Continuing

C. Compile, run and output of Pausing, Continuing, Doing Finally

D. Compile time error finally clause never reached

8. Which of the following will distinguish between a directory and a file C

A. FileType()

B. isDir()

C. isDirectory()

D. getDirectory()Ⅳ. Graphical Programming

**Ⅳ.Graphical Programming**

1.What best describes the apprearance of an applet with the following code? C

import java.awt.\*;

public class FlowAp extends Frame{

public static void main(String argv[]){

 FlowAp fa=new FlowAp();

 fa.setSize(400,300);

 fa.setVisible(true);

}

FlowAp(){

 add(new Button("One"));

 add(new Button("Two"));

 add(new Button("Three"));

 add(new Button("Four"));

 }//End of constructor

}//End of Application

A. A Frame with buttons marked One to Four placed on each edge.

B. A Frame with buutons Makred One to four running from the top to bottom

C. A Frame with one large button marked Four in the Centre

D. An Error at run time indicating you have not set a LayoutManager

2.How do you indicate where a component will be positioned using Flowlayout? D

A. North, South,East,West

B. Assign a row/column grid reference

C. Pass a X/Y percentage parameter to the add method

D. Do nothing, the FlowLayout will position the component

3.How do you change the current layout manager for a container A

A. Use the setLayout method

B. Once created you cannot change the current layout manager of a component

C. Use the setLayoutManager method

D. Use the updateLayout method

4. What will happen if you add a vertical scroll bar to the North of a Frame? B

A. The Frame will enlarge to allow the scrollbar to become its preferred size

B. It will be wide, fat and not very useful

C. You cannot add a vertical scroll bar to the North of a frame, only the East or West

D. The scrollbar will stretch from the top to the bottom of the Frame

5. What happens if you add more buttons to a GridLayout than can fit and and fully display the button labels? B

A. The size of the container is increased to allow the button labels to fully display

B. The GridLayout ignores the size of the label and the labels will be truncated

C. A compile time error indicating the Buttons cannot be the preferred size

D. A run time error indicating the buttons cannot be the preffered size.

6. Which of the following statements are true? D

A. You can control component placing by calling setLayout(new GridBagConstraints())

B. The FlowLayout manager can be used to control component placing of the GridBagLayout

C. The GridBagLayout manager takes constraints of North, South, East, West and Center

D. None of these answers is true

7. Which of the following are fields of the GridBagConstraints class? A B C

A. ipadx

B. fill

C. insets

D.width

8. Which statements are correct about the anchor field? B C

A. It is a field of the GridBagLayout manager for controlling component placement

B. It is a field of the GridBagConstraints class for controlling component placement

C. A valid settting for the anchor field is GridBagConstraints.NORTH

D. The anchor field controls the height of components added to a container

9.Which of the following statements are true? C D

A. For a given component events will be processed in the order that the listeners were added

B. Using the Adapter approach to event handling means creating blank method bodies for all event methods

C. A component may have multiple listeners associated with it

D. Listeners may be removed once added

**Ⅴ. The Development Of Database And Web Application**

1. What is the correct choice of JDBC? In this case,

A.JDBC is designed to be a common database connection technology, JDBC technology can be applied not only in the Java process

Order inside, but also can be used in C + + such procedures inside. In this case,

B. JDBC technology is designed to connect SUN company designed to connect to the Oracle database technology, connecting the other

The database can only use Microsoft's ODBC solution

C. Microsoft's ODBC and SUN's JDBC solution can achieve cross-platform use, but the performance of JDBC

To be higher than ODBC

D. JDBC is just an abstract call specification, the underlying program actually depends on each database driver file

1. Select the statements that JDBC can execute (multiple choice)

A. DDL

B. DCL

C. DML

D. All of the above

1. Select the Java program recommended in the development of commonly used database (multiple choice)

A. Orancle

B. SQL Server 2000

C. MySQL

D. DB2

1. Which is not used JDBC interface and class?

A. System

B. Class

C. Connection

D. ResultSet

5. Which method of Connection can be used to create a PreparedStatement interface? In this case,

A. createPrepareStatement ()

B. prepareStatement ()

C. createPreparedStatement ()

D. preparedStatement ()

6. What is the correct description below? In this case,

A. PreparedStatement inherits from Statement

B. Statement inherits from PreparedStatement

C. ResultSet inherits from Statement

D. CallableStatement inherited from PreparedStatement

7. What's wrong with the description below? In this case,

A. The executeQuery () method of a Statement returns a result set

B. Statement of the executeUpdate () method will return whether to update the success of the boolean value C. Use ResultSet getString () can be obtained in a database corresponding to the type of charValue

D. The next () method in a ResultSet causes the next row in the result set to be the current row

8. If a field in the database is numeric, which of the methods in the result set can be obtained?

A. getNumberic ()

B. getDouble ()

C. getBigDecimal ()

D. getFloat ()

**Ⅵ. The Principle Of Multi-Thread Mechanism And Simple Application**

1. What will happen when you attempt to compile and run this code? C

public class Runt implements Runnable{

public static void main(String argv[]){

 Runt r = new Runt();

 Thread t = new Thread(r);

 t.start();

 }

 public void start(){

 for(int i=0;i<100;i++)

 System.out.println(i);

 }

}

A. Compilation and output of count from 0 to 99

B. Compilation and no output

C. Compile time error: class Runt is an abstract class. It can't be instantiated.

D. Compile time error, method start cannot be called directly

2. Which of the following statements are true? C D

A. Directly sub classing Thread gives you access to more functionality of the Java threading capability than using the Runnable interface

B. Using the Runnable interface means you do not have to create an instance of the Thread class and can call run directly

C. Both using the Runnable interface and subclassing of Thread require calling start to begin execution of a Thread

D. The Runnable interface requires only one method to be implemented, this is called run

3. What will happen when you attempt to compile and run the following code? A

public class Runt extends Thread{

public static void main(String argv[]){

 Runt r = new Runt();

 r.run();

 }

 public void run(){

 for(int i=0;i<100;i++)

 System.out.println(i);

 }

}

A. Compilation and output of count from 0 to 99

B. Compilation and no output

C. Compile time error: class Runt is an abstract class. It can't be instantiated.

D. Compile time error, method start has not been defined

4. Which of the following statements are true? B C D

A. To implement threading in a program you must import the class java.io.Thread

B. The code that actually runs when you start a thread is placed in the run method

C. Threads may share data between one another

D. To start a Thread executing you call the start method and not the run method

5. What will happen when you attempt to compile and run this code? D

public class TGo implements Runnable{

public static void main(String argv[]){

 TGo tg = new TGo();

 Thread t = new Thread(tg);

 t.start();

 }

 public void run(){

 while(true){

 Thread.currentThread().sleep(1000);

 System.out.println("looping while");

 }

 }

}

A. Compilation and no output

B. Compilation and repeated output of "looping while"

C. Compilation and single output of "looping while"

D. Compile time error

6. Which of the following are recommended ways a Thread may be blocked? A B

A. sleep()

B. wait/notify

C. suspend

D. pause

7. Which of the following statements are true? C D

A. The sleep method takes parameters of the Thread and the number of seconds it should sleep

B. The sleep method takes a single parameter that indicates the number of seconds it should sleep

C. The sleep method takes a single parameter that indicates the number of milliseconds it should sleep

D. The sleep method is a static member of the Thread class

8. Which of the following keywords indicates a thread is releasing its Object lock? B

A. release

B. wait

C. continue

D. notifyAll

**Ⅶ. The Graphical Interface Database And Network Integrated Application**

1. Instant messaging tool based on C/S.

Requirements:

1）C / S mode, based on Socket implementation, the server-side use multi-threaded to handle multiple client connections.

2）Interface: GUI interface, friendly and intuitive, reasonable layout, convenient function entrance.

3）Data storage: server-side storage one or more classes’ information and information of students in classes. Use the database to store.

Feature：

1. Login: enter the user name and password; user name is the student number, with the initial password, for example, 1111
2. Login verification: According to the server-side storage of class information, verify whether the user is in the class or a student of classes.
3. The main interface lists the student buddy list (you can delete or append friends).
4. Instant chat, select a friend, double-click to open the window chat. If not online, the server-side will forwarded after friend online(this feature is extended to achieve).
5. Students can modify their login password.

**4、课程网站建设（节选）**



