# 2019年度　留学生2年　前期単位認定試験　Java3、Java　Bronze

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。

|  |
| --- |
| public class Main{  public static void main( String[] args){  for( int i = 0 ; i < 5; i++){  for ( i = 5; i < 10 ; i++ ){  System.out.println( i);  }  }  }  } |

1. 何も表示されない。
2. 「56789」と1回表示される ○
3. 「56789」と5回表示される
4. コンパイルエラーになる。
5. 実行時に例外が発生する。

## 次のSampleクラスを継承したサブクラスを定義するときに、サブクラスに定義したメソッドのうち、Sampleクラスのメソッドを正しくオーバーライドしているものを選びなさい。

|  |
| --- |
| public class Sample {  void MethodA(){}  void MethodB( int a ){}  void MethodC( int a, int b){  return 0;  }  void MethodD( int a ){  return 0;  }  } |

1. public void methodA(){} ○
2. public void methodB( long a ){}
3. public int methodC( char a, int b){  
    return 0;  
   }
4. public int methodD( int i ){ ○  
    return 1;  
   }

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Train {  static String name = "none";  public Train( String name){  this.name = name;  }  public static void main( String[] args){  Train t1 = new Train();  Train t2 = new Train( "aline");  System.out.println( t1.name + " " + t2.name) ;  }  } |

1. 「none aline」と表示される
2. 「null aline」と表示される
3. 「aline aline」と表示される ○
4. コンパイルエラーになる
5. 実行時に例外が発生する

## ある企業は、GUIベースのアプリケーション開発を希望しており、将来的な拡張として、Webベースのアプリケーションへの移行を予定している。このアプリケーションを作成するには、どのJavaテクノロジを使用するのが良いか。正しいものを選びなさい。（1つ選択）

1. JavaSE ○
2. JavaME
3. JavaDB
4. JavaEE

## 次のプログラムをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public interface Sample {  void test();  } |

|  |
| --- |
| public class A implements Sample{  public void test(){  System.out.println( "A");  }  } |

|  |
| --- |
| public class B extends A {  public void test(){  System.out.println( "B");  }  } |

|  |
| --- |
| public class Main{  public static void main( String[] args){  Sample[] samples = { new A(), new B()};  for ( Sample s : samples){  a.test();  }  }  } |

1. 「A」「B」の順に表示される ○
2. 「B」「A」の順に表示される
3. 「A」「A」の順に表示される
4. 「B」「B」の順に表示される
5. Bクラスでコンパイルエラーが発生する
6. Mainクラスでコンパイルエラーが発生する

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択））

|  |
| --- |
| public class P249 {  public static void main( String[] args){  int i = 5;  System.out.println((i += 5 ) + ":" + ( i--));  }  } |

1. 「5:5」と表示される
2. 「5:4」と1回表示される
3. 「5:9」と表示される
4. 「10:10」と表示される ○
5. 「10:9」と表示される

## 以下の中から、アクセス修飾子privateで修飾できるものを選びなさい。（3つ選択）

1. インタフェースのフィールド
2. クラスのコンストラクタ ○
3. クラスのフィールド ○
4. クラスの抽象メソッド
5. インタフェースのメソッド
6. クラスの具象メソッド ○

## 次のコードのコンパイルを成功させ、実行結果が「Refresh L」となるようにしたい。空欄にあてはまるコードを選びなさい。（1つ選択）

|  |
| --- |
| public class P250{  private String size;  private String title;  P250 (){  }  P250( String title){  this.title = title;  }  void printInfo(){  System.out.println( title + " " + size);  }  public static void main( String[] args){  P250 p = new P250( );  p.printInfo();  }  } |

1. this.size = "L";  
   this( "refresh");
2. Photo( "refresh");  
   this.size = "L";
3. this( "refresh" ); ○

this.size = "L";

1. this.size = "L";  
   Photo( "refresh");
2. this.Photo( "refresh");

this.size = "L";

## 次のプログラムをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class SuperClass {  protected int num ;  public SuperClass(){  this.num = 1;  }  public SuperClass( int num){  this.num = num;  }  } |

|  |
| --- |
| public class SubClass extends SuperClass{  private int a;  private int b;  public SubClass( int a ){  this.a = a;  }  public SubClass( int a, int b ){  this( a);  this.b = b;  }  public static void main( String[] args){  SubClass sub = new SubClass( 2, 3);  System.out.println( sub.num + ":" + sub.a + ":" + sub.b);  }  } |

1. 「1:2:3」と表示される ○
2. 「0:2:3」と表示される
3. コンパイルエラーになる
4. 実行時に例外がスローされる

## クラスの宣言として、有効なものを選びなさい。（３つ選択）

1. public class Test extends java.lang.\* { }
2. public class Test extends java.lang.Object{} ○
3. final class Test{} ○
4. public class Test{} ○
5. public class Test implements Object{}

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Main{  public static void main( String[] args){  int a = 12;  int b = 8;  if ( a >= 10 || b >= 10 ){  a /= 2;  b += a;  } else  a /= 2;  b += a;  System.out.println( a + ":" + b );  }  } |

1. 「6:14」と表示される
2. 「6:20」と表示される ○
3. 「12:20」と表示される
4. コンパイルエラーとなる

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Main{  public static void main( String[] args){  for ( int i = 0 ; ; i ++ ){  int j = 0;  while ( i <= 3 ){  System.out.println( j++ );  }  }  }  } |

1. 「123」と1回表示される
2. 「0123」と1回表示される
3. 「1234」と1回表示される
4. 「1234」が無限に表示される
5. 「1234」が無限に表示される ○
6. コンパイルエラーとなる

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public static Main{  public static void main( String[] args){  for( int i = 0; ++ i < 5 ;){  System.out.println( i + " ");  }  }  } |

1. 何も表示されない
2. 「01234」と表示される
3. 「012345と表示される
4. 「1234と表示される ○
5. 「12345と表示される
6. コンパイルエラーになる

## 次のプログラムの6行目「// inser code here 」に入るコードとして、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Sample {  private String name;  private int price;  public String TMP = "sample";  public Sample(){  // insert code here  }  public Sample( String name){  this.name = name;  }  public void print(){  System.out.println( name + ":" + price);  }  public static void main( String[] args){  Sample s = new Sample();  s.print();  }  } |

1. this.price = 100 ;  
   this ( "sample");
2. this( "sample"); ○  
   this.price = 100;
3. this( TMP);  
   this.price 100;
4. this.price = 100;  
   this( TMP);
5. Sample( "sample");  
   this.price = 100;

## 次のコードをコンパイル、実行した時の結果として、正しいものを選択しなさい。（1つ選択）

|  |
| --- |
| public class Main{  public static void main( String[] args ){  int x = 6;  int y = x++;  int z = ++y;  System.out.println( x + " " + y + " " + z);  }  } |

1. 「6 6 7」と表示される
2. 「6 7 7」と表示される
3. 「6 6 6」と表示される
4. 「7 7 7」と表示される ○
5. コンパイルエラーとなる

## 次のインタフェースのコンパイルを成功させるには、どのコードを2行目の「//insert code here」に入れればよいか。正しいものを選びなさい。（２つ選択）

|  |
| --- |
| public interface Sample{  // insert code here  } |

1. public void setVal( String val); ○
2. private void setVal( String val );
3. String val;
4. void setVal(String val); ○
5. public static void setVal(String val);

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Bridge{  String name;  public static void main( String[] args){  Bridge b = new Bridge();  if ( b.name == "")  b.name = "Brookly";  System.out.println( b.name);  }  } |

何も表示されない

1. 「null」と表示される
2. 「Brooklyn」と表示される ○
3. コンパイルエラーになる
4. 実行時に例外が発生する

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Main{  public static void main( String[] args ){  int x = 5;  int y = ( x = 2 ) + x;  System.out.println( y + ":" + x );  }  } |

1. 「7:5」と表示される
2. 「2:5」と表示される
3. 「7:20」と表示される
4. 「4:5」と表示される
5. 「4:20」と表示される ○
6. コンパイルエラーになる。

## xxxというパッケージにアクセスできるMainというクラスを、xxx.hogeというパッケージ内に作成するには、どのようにクラスを宣言すればよいか。正しいものを選びなさい。（1つ選択）

1. import xxx.\*;  
   package xxx.hoge;  
   public class Main {

// any code

}

1. package xxx.hoge; ○  
   import xxx.\*;  
   public class Main{

// any code

}

1. import xxx;  
   package xxx.hoge;  
   public class Main{

// any code

}

1. package xxx.hoge;  
   import xxx;  
   public class Main{

// any code

}

1. package xxx.\*;  
   public class Main{

// any code

}

## 次のコードをコンパイル、実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Main{  public static void main(String[] args) {  String str1 = "abcdefghijk";  String str2 = null;  switch (str1.charAt(5)) {  case 'e':  str2 = "char E";  break;  case 'f':  str2 = "char F";  break ;  default:  str2 = "other";  break;  }  }  } |

1. 「char E」と表示される
2. 「char F」と表示される
3. 「other」と表示される
4. コンパイルエラーになる
5. 実行時に例外が発生する。

## 以下の中からポリモーフィズムに関係が深いものを選びなさい。

1. インタフェースの継承
2. インタフェースの実装 ○
3. メソッドのオーバーロード
4. メソッドのオーバーライド
5. アクセス修飾子protectedの利用

## クラスブロック内に定義できる要素として正しいものを選びなさい。(3つ選択）

1. フィールド
2. メソッド
3. コンストラクタ
4. パッケージ文

## 次のコードをコンパイルし、実行方法の通りに実行した時の結果として、正しいものを選びなさい。（1つ選択）

|  |
| --- |
| public class Main{  public static void main( String[] args){  System.out.println( args[1] + "hello");  }  } |

【実行方法】

java Main hoge fuga

1. Main hello
2. hoge hello
3. fuga hello
4. Main fuga hello

## クラスの定義として正しいものを選びなさい。（3つ選択）

1. public class $Item { } ○
2. public class Book%{ }
3. public class Employee#{ }
4. class Shape5{ } ○
5. class Sample-Test{　}
6. class Book\_Stop{　} ○

## コンストラクタの定義に関する説明として正しいものを選びなさい。（1つ選択）

1. コンストラクタの戻り値型はvoidにする
2. コンストラクタには、引数を定義できない
3. コンストラクタ名は、クラスメイト同じ値にする
4. コンストラクタには、戻り値型を指定できる。

# 2019年度　留学生2年　Java、　JavaBronze　前期期末試験問題　解答用紙

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 問題１ | 問題２ | 問題３ | 問題４ | 問題５ |
|  |  |  |  |  |
| 問題６ | 問題７ | 問題８ | 問題９ | 問題１０ |
|  |  |  |  |  |
| 問題１１ | 問題１２ | 問題１３ | 問題１４ | 問題１５ |
|  |  |  |  |  |
| 問題１６ | 問題１７ | 問題１８ | 問題１９ | 問題２０ |
|  |  |  |  |  |
| 問題２１ | 問題２２ | 問題２３ | 問題２４ | 問題２５ |
|  |  |  |  |  |

学績番号　　　　　　　　　　名前