# Java Bronze　後期中間試験　令和1年12月10日　実施

## カプセル化の説明として、正しいものを一つ選びなさい。 （　C　 ）

1. 同種の異なるインスタンスを同じ型として扱う
2. 公開すべきものと非公開にすうべきものを区別して扱う
3. 関係するものをひとまとめにする
4. クラスのフィールドを隠蔽し、他のクラスからは直接使えないようにする。

## 従業員の情報を表すEmployeeくらすがある。正しくカプセル化されるようにこのクラスを修正したい。修正内容として、正しいものを2つ選びなさい。 （　C,D　 ）

|  |
| --- |
| public class Employee{  String corporateName;  String corporateAddress;  int employeeNo;  String name;  } |

1. すべてのフィールドのアクセス修飾子をprivateにする
2. フィールドにアクセするためのgetter、setterメソッドを追加する
3. corporateNameとcorporateAddressフィールドを他のクラスに移動する
4. 名前を名乗ってあいさつするメソッドを追加する
5. 給与計算のメソッドを追加する

## カプセル化の維持に欠かせない原則として、正しいものを1つ選びなさい。 （　A　 ）

1. データ隠蔽
2. 情報隠蔽
3. 抽象化
4. データ抽象

## データ隠蔽を実現するためには、フィールドをどのように修飾すればよいか。正しいものを1つ選びなさい。 （　C　 ）

1. public final
2. public static
3. private
4. private static
5. private final

## データ隠蔽の説明として、最も適切なものを1つえらびなさい。 （　B　 ）

1. 関係するデータとそのデータを扱う処理をひとまとめにする
2. 属性の公開範囲を制限する
3. getterやsetterなどのアクセサメソッドを提供する
4. 実装の詳細を隠蔽する

## 共通部分だけを抽出し、それ以外を無視して扱うことを何と呼ぶか。正しい用語を選びなさい。 （　A　 ）

1. 抽象化
2. カプセル化
3. データ隠蔽
4. 情報隠蔽

## 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。 （　B　 ）

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

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

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

## 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。(2つ選択） （　A、D　 ）

|  |
| --- |
| public class Sample{  void methodA(){}  void methodB(int a ){}  void methodC( int a, int b){  return 0;  }  voiod methodD( int a ){  return 1;  }  } |

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 ; }

## クラス定義の記述として、正しいものを１つ選びなさい。 （　　 ）

1. class ClassA ( int a ){  
    // クラス内定義  
   }
2. ClassA {  
    // クラス内定義  
   }
3. class ClassA ( ) {  
    // クラス内定義  
   }
4. class ClassA {  
    // クラス内定義  
   }

## 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） （　247-3　 ）

|  |
| --- |
| 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。正しいものを一つ選びなさい。 （　248-4　 ）

1. Java　SE
2. Java　EE
3. Java　ME
4. Java　DB

## 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１つ選びなさい。 （　248-5　 ）

|  |
| --- |
| 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 ){  s.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1つ選びなさい。 （　249-6　 ）

|  |
| --- |
| public class Main{  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で就職できるものを選びなさい。 （　249-7　 ）

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

## クラスの宣言として有効なものを二つ選びなさい。 （　252-10　 ）

1. public class Test extends java.lang.\*;
2. public class