*\*&---------------------------------------------------------------------\**  
*\*& Report ZVAISH\_INHERITANCE*  
*\*&---------------------------------------------------------------------\**  
*\*&*  
*\*&---------------------------------------------------------------------\**  
REPORT ZVAISH\_INHERITANCE.  
  
*\*class superclass definition.*  
*\*  public section.*  
*\*  methods supcmethod1.*  
*\*  data : supcnum1 type I value 6.*  
*\**  
*\*  PROTECTED SECTION.*  
*\*  methods supcmethod2.*  
*\*  data: supcnum2 type i value 10.*  
*\*  endclass.*  
*\**  
*\*  class superclass implementation.*  
*\*    method supcmethod1.*  
*\*      write ' i am public method of super class'.*  
*\*      endmethod.*  
*\*    method supcmethod2.*  
*\*      write 'i am protected method of super class'.*  
*\*    endmethod.*  
*\*    endclass.*  
*\**  
*\**  
*\*    class subclass DEFINITION inheriting from superclass.*  
*\*      public section.*  
*\*      data subc type i value 40.*  
*\*      methods submethod.*  
*\**  
*\*      endclass.*  
*\**  
*\*      class subclass IMPLEMENTATION.*  
*\*        method submethod.*  
*\*              write 'i am public method of subclass'.*  
*\*              call method supcmethod2. "protected method of superclass can only be accessed in subclass method.*  
*\*              write supcnum2.*  
*\*          ENDMETHOD.*  
*\*        endclass.*  
*\*START-OF-SELECTION.*  
*\*\*data supclsobj type REF TO superclass.*  
*\*\*create object supclsobj.*  
*\*\*supclsobj->supcmethod1( ).*  
*\*\*\*supclsobj->supcmethod2( ).*  
*\*\*write supclsobj->supcnum1.*  
*\*\*\*write supclsobj->supcnum2.*  
*\**  
*\*data subclsobj type REF TO subclass.*  
*\*create object subclsobj.*  
*\*subclsobj->supcmethod1( ).*  
*\*\*subclsobj->supcmethod2( ).*  
*\*subclsobj->submethod( ).*  
*\*write subclsobj->supcnum1.*  
*\*\*write supclsobj->supcnum2.*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*# 2 subclass methods can re-implement inherited methods from superclass*  
  
*\*class superclass DEFINITION.*  
*\*  public section.*  
*\*  methods: supp1.*  
*\*  protected section.*  
*\*  methods: supp2.*  
*\*  endclass.*  
*\**  
*\*  class superclass IMPLEMENTATION.*  
*\*    method: supp1.*  
*\*      write 'im superclass method of public visibility'.*  
*\*      endmethod.*  
*\*      method supp2.*  
*\*        write 'im superclass method of private visibility'.*  
*\*        endmethod.*  
*\*    ENDCLASS.*  
*\**  
*\*    class subclass DEFINITION INHERITING FROM superclass.*  
*\*      public section.*  
*\*      methods supp1 REDEFINITION. "overriding concept from java same method name different application or implementation.*  
*\**  
*\*      PROTECTED SECTION.*  
*\*        methods supp2 REDEFINITION.*  
*\*      endclass.*  
*\**  
*\*      class subclass IMPLEMENTATION.*  
*\*        method supp1.*  
*\*          write 'i am subclass method of public visibility'.*  
*\*          endmethod.*  
*\*          method supp2.*  
*\*            write 'i am subclass method of protected visibility'.*  
*\*            endmethod.*  
*\*        endclass.*  
*\*        START-OF-SELECTION.*  
*\*        data :supobj type REF TO superclass,*  
*\*              subobj type ref to subclass.*  
*\**  
*\*        create object: supobj, subobj.*  
*\*        supobj->supp1( ). "superclass method  of public visibility*  
*\*\*supobj->supp2( ). "cannot acces from here as it is protected.*  
*\**  
*\**  
*\*\*subobj->subp1( ). "subclass method of public visibility*  
*\*\*subobj->subp2( ). "caanot as its protected.*  
*\*        subobj->supp1( ).*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*#3. objects are not created in abstract class.*  
  
*\*class superclass DEFINITION ABSTRACT.*  
*\*  public section.*  
*\*  methods supmethod1.*  
*\**  
*\*  ENDCLASS.*  
*\**  
*\*\*\*\*\*\*\*\*\*\*\*  class superclass IMPLEMENTATION.*  
*\*\*\*\*\*\*\*\*\*\*\*    method supmethod1.*  
*\*\*\*\*\*\*\*\*\*\*\*      write 'abstarct method'.*  
*\*\*\*\*\*\*\*\*\*\*\*      endmethod.*  
*\*\*\*\*\*\*\*\*\*\*\*    endclass.*  
*\**  
*\*    class subclass DEFINITION INHERITING FROM superclass.*  
*\*      public section.*  
*\*      methods submethod.*  
*\**  
*\*      ENDCLASS.*  
*\**  
*\*\*\*      class subclass IMPLEMENTATION.*  
*\*\*\*        method submethod.*  
*\*\*\*          write 'subclass method'.*  
*\*\*\*          call method supmethod1( ).*  
*\*\*\*          endmethod.*  
*\**  
*\*        endclass.*  
*\*        START-OF-SELECTION.*  
*\*\*data supobj type REF TO superclass.*  
*\*\*create object supobj. "abstarct class obeject cannot be created.*  
*\**  
*\*        data subobj type REF TO subclass.*  
*\*        create object subobj.*  
*\*        subobj->submethod( ).*  
*\*        subobj->supmethod1( ).*  
  
*\*\*\*\*\*\*\*\*\*\*# 4  abstract method should be redefined then implemented in\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**  
*\*class superclass DEFINITION ABSTRACT.*  
*\*  public section.*  
*\*  methods supmethod1 abstract.*  
*\*  methods supmethod2.*  
*\**  
*\*  ENDCLASS.*  
*\**  
*\*  class superclass IMPLEMENTATION.*  
*\*\*\*\*\*\*\*\*\*    method supmethod1.  "abstract method cannot be implemented until its redifined or overridden.*  
*\*\*\*\*\*\*\*\*\*      write 'abstarct method'.*  
*\*\*\*\*\*\*\*\*\*      endmethod.*  
*\*      method supmethod2.*  
*\*        write 'non abstarct method'.*  
*\*        endmethod.*  
*\*    endclass.*  
*\**  
*\*    class subclass DEFINITION INHERITING FROM superclass.*  
*\*      public section.*  
*\*      methods submethod.*  
*\*  methods supmethod1 REDEFINITION.*  
*\*      ENDCLASS.*  
*\**  
*\*      class subclass IMPLEMENTATION.*  
*\*        method submethod.*  
*\*          write 'subclass method'.*  
*\*\*call method supmethod1( ).*  
*\*          endmethod.*  
*\*          method supmethod1.*  
*\*            write 'abstarct method of super class'.*  
*\*            ENDMETHOD.*  
*\**  
*\*        endclass.*  
*\*        START-OF-SELECTION.*  
*\*\*data supobj type REF TO superclass.*  
*\*\*create object supobj. "abstarct class obeject cannot be created.*  
*\**  
*\*        data subobj type REF TO subclass.*  
*\*        create object subobj.*  
*\*        subobj->submethod( ).*  
*\*        subobj->supmethod1( ).*  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*\*5.\*\*\*\*\*\*\*\*\*\*\**  
*\*CLASS superclass DEFINITION FINAL.*  
*\*  public section.*  
*\*  methods finalmeth.*  
*\*ENDCLASS.*  
*\**  
*\*CLASS superclass IMPLEMENTATION .*  
*\*  method finalmeth.*  
*\*    write 'im final method of superclass'.*  
*\*    endmethod.*  
*\*ENDCLASS.*  
*\**  
*\*\*CLASS subclass DEFINITION INHERITING FROM superclass. "final class cannot be inherited*  
*\*class subclass DEFINITION.*  
*\*  public section.*  
*\*  methods submethod.*  
*\*ENDCLASS.*  
*\**  
*\*CLASS subclass IMPLEMENTATION.*  
*\*  method submethod.*  
*\*    write 'i am submethod of subclass'.*  
*\*    endmethod.*  
*\*ENDCLASS.*  
*\**  
*\*START-OF-SELECTION.*  
*\*DATA : subobj TYPE REF TO subclass.*  
*\*CREATE OBJECT subobj.*  
*\*subobj->submethod( ).*  
*\*\*\*subobj->finalmeth( ). "cannot access as not inherited*  
*\**  
*\*data supobj type REF TO superclass.*  
*\*create object supobj.*  
*\*supobj->finalmeth( ).*  
  
  
  
*\*\*\*\*\*\*\*\*\*\*#6. final methods cannot be redefined*  
  
CLASS C1 DEFINITION .  
  PUBLIC SECTION.  
  METHODS : METH1.  *"FINAL.*  
  METHODS : METH2 FINAL.  
ENDCLASS.  
  
CLASS C1 IMPLEMENTATION .  
  METHOD meth1.  
    WRITE:/5 'I am method meth1'.  
  ENDMETHOD.  
  
  METHOD meth2.  
    WRITE:/5 'I am method meth1'.  
  ENDMETHOD.  
  
ENDCLASS.  
  
CLASS C2 DEFINITION INHERITING FROM C1.  
  PUBLIC SECTION.  
  METHODS : meth1 REDEFINITION.  
*\*\*\*\*METHODS : meth2 REDEFINITION.*  
ENDCLASS.  
  
CLASS C2 IMPLEMENTATION.  
  METHOD : meth1.  
    WRITE:/5 ' I am meth1,modified in class C2'.  
  ENDMETHOD.  
  
 ENDCLASS.  
  
START-OF-SELECTION.  
DATA : OREF2 TYPE REF TO C2.  
CREATE OBJECT oref2.  
CALL METHOD OREF2->meth1.  
  
  
  
*\*\*\*\*\*\*\*\*\*\*\*\*for more practice\*\*\*\*\*\*\*\**  
class employee DEFINITION.  
  public section.  
  methods: setinfo IMPORTING lv\_empno type i lv\_empname type string,  
          displayinfo.  
  
  PRIVATE SECTION.  
  data: empno type i,  
        empname type string.  
  
  endclass.  
  
  class employee IMPLEMENTATION.  
    method setinfo.  
      empno = lv\_empno.  
      empname = lv\_empname.  
      ENDMETHOD.  
      method displayinfo.  
        write :/ 'EMPLOYEE DETAILS',empno,empname.  
        ENDMETHOD.  
    endclass.  
  
    class bluecollaremp DEFINITION.  
      public SECTION.  
      methods: setinfo importing lv\_empno type i lv\_empname type string,  
        displayinfo REDEFINITION.  
      ENDCLASS.  
      class bluecollaremp IMPLEMENTATION.  
  
        ENDCLASS.  
  
    data empobj type REF TO employee.  
    create object empobj.  
  
    data bempobj type REF TO bluecollaremp.  
    create object bempobj.