Nama : Muhammad Naufal Ariq

NPM : 0620101009

1. **Acces modifier**: tujuan nya adalah membatasi resource-resource mana saja yang dapat diakses oleh object tertentu

* Public: class atau variable yang dapat diakses oleh seluruh class
* Private: class atau variable yang hanya bisa di akses oleh class itu sendiri
* Protected: dapat diakses oleh subclass atau class lain asalkan didalam satu package yang sama

1. **Inheritence:** dalam inheritance dimana entitas dapat mempunyai banyak objek turunan
2. **Polymorphsm:** di mana class memiliki nama yang sama namun dapat memiliki method yang berbeda-beda.
3. **Interface:** digunakan secara metafora untuk perbatasan antara benda
4. **Exception Handling (try catch):** mekanisme yang paling diperlukan dalam menangani error.

|  |
| --- |
| import java.util.Scanner;  public class himp{  Scanner input = new Scanner(System.in);  int[] Array1= new int[100];  int[] Array2= new int[100];  int JumlahArray1;  int JumlahArray2;  public void dataAnggota() {  System.out.print("Masukkan Jumlah Array 1 : ");  JumlahArray1 = input.nextInt();  System.out.println("Anggota Array 1");  for(int i=1;i<=JumlahArray1;i++){  System.out.print("Anggota "+i+" : ");  Array1[i]=input.nextInt();  }  System.out.print("Masukkan Jumlah Array 2 : ");  JumlahArray2 = input.nextInt();  System.out.println("Anggota Array 2");  for(int i=1;i<=JumlahArray2;i++){  System.out.print("Anggota "+i+" : ");  Array2[i]=input.nextInt();  }  }  public void tampilAnggota(){  System.out.print("Anggota Array 1 = { ");  for(int i=1;i<=JumlahArray1;i++){  System.out.print(Array1[i]+" ");  }  System.out.println("}");  System.out.print("Anggota Array 2 = { ");  for(int i=1;i<=JumlahArray2;i++){  System.out.print(Array2[i]+ " ");  }  System.out.println("}");  }  public void himpunanIrisan(){  System.out.print("Himpunan Irisan Array 1 dan 2 : { ");  for( int i=1;i<=JumlahArray2;i++){  boolean adaSama = false;  for(int j=1;j<=JumlahArray1;j++){  if(Array2[i]==Array1[j])  adaSama=true;  }  if(adaSama==true){  System.out.print(Array2[i]+" ");  }  }  System.out.println(" }");  }  public static void main(String args[]){  himp bilangan = new himp();  bilangan.dataAnggota();  bilangan.tampilAnggota();  bilangan.himpunanIrisan();  }  } |



|  |
| --- |
| public class DataClass {  int numberData; String wordsData;  public DataClass(int numberData, String **wordsData** ){  this.numberData = **numberData**;  **this.wordsData** = wordsData;  }  Public int getNumberData(){**int[]numberData**;}  public String getWordsData(){**String[]wordsData**;}  }  public interface ParentInterface {  public DataClass initData(int number, String words);  **return new DataClass(number, words);**  }  public class ParentClass implements **ParentInterface**{  public DataClass initData(int number, String words){  return new DataClass(number, words);  }  public String getDataString() { return “” ; }  }  public class childClass extends **ParentClass**{  DataClass childData ;  public void setData() {  childData = initData(1, “firstChild”);  }  @override  public String getDataString() { return childData.getWordsData(); }  }  public class secondChild extends **ParentClass**{  DataClass secondChildData ;  public void setData() {  secondChildData = initData(2, “secondChild”);  }  @override  public String getDataString() { return **secondChildData**.getWordsData(); }  }  public MainClass{  public static void main (String [] args){  // do polymorphsm here  **childClass**.objek ;  objek = new childClass();  objek.setData();  System.out.println(objek.getDataString());  objek = **new secondChild();** // change to secondChild  objek.setData();  System.out.println(objek.getDataString());  }  } |