**Soal Pertama**Toni ingin membuat satu Class yang dapat menghitung luas dari beberapa jenis bangun datar sekaligus, didalam Class Bangun Datar terdapat method-method yang ber-overload satu sama lain, seperti berikut:

public double luas(double r) // Lingkaran

public double luas(double a, double t,) // segitiga

public int luas (int s) // persegi

public int luas(int p, int l) //persegi panjang

Buatlah ke empat method diatas, kemudian cobalah panggil di Main Class!

**Jawaban**

**Source Code**

|  |
| --- |
| 1. **public** **class** BangunDatar { 2. **double** phi=3.14; 4. **public** **double** luas (**double** r){ 5. **return** phi\*r\*r; 6. } 7. **public** **double** luas (**double** a , **double** t){ 8. **return** 0.5\*a\*t; 9. } 10. **public** **int** luas(**int** s){ 11. **return** s\*s; 12. } 13. **public** **int** luas(**int** p, **int** l){ 14. **return** p\*l; 15. } 16. } 18. **public** **class** App { 19. **public** **static** **void** main(String[] args) **throws** Exception { 20. BangunDatar hitung = **new** BangunDatar(); 21. System.out.println("luas lingkaran = "+ hitung.luas(10.0)); 22. System.out.println("luas segitiga = "+ hitung.luas(4.0,2.0)); 23. System.out.println("luas Persegi = "+ hitung.luas(4)); 24. System.out.println("luas Persegi Panjang = "+ hitung.luas(4,2)); 25. } 26. } |

**Output Program**

|  |
| --- |
|  |

**Soal Kedua**Perhatikan Kode Program berikut ini :

|  |
| --- |
| 1. **public** **class** SuperClass { 2. **public** **static** **void** cetak1(){ 3. System.out.println("static method superclass"); 4. } 5. **public** **final** **void** cetak2 (){ 6. System.out.println("final method superclass"); 7. } 8. **public** **void** cetak3(){ 9. System.out.println("method dari superclass"); 10. }  13. } |

|  |
| --- |
| 1. **public** **class** AdaYangSalah **extends** SuperClass{ 2. **public** **static** **void** cetak1(){ 3. System.out.println("static method subclass"); 4. } 5. **public** **final** **void** cetak2(){ 6. System.out.println("final method "); 8. } 9. **public** **void** cetak3(){ 10. System.out.println("method subclass"); 11. } 12. } |

Cobalah kode program diatas, Jika ada error, Jelaskan apa yang menyebabkan error dari kode program diatas, kemudian berikan kode program yang sudah benar !

**Jawaban**

Yaitu terdapat kesalahan pada class SuperClass, yang mana pada public final void cetak2 harusnya yaitu public void cetak2

**Source Code**

Main

|  |
| --- |
| 1. **public** **class** Main { 2. **public** **static** **void** main(String[] args) { 3. SuperClass.cetak1(); 4. AdaYangSalah.cetak1(); 5. } 6. } |

SuperClass

|  |
| --- |
| 1. **public** **class** SuperClass { 2. **public** **static** **void** cetak1(){ 3. System.out.println("Static method superclass"); 4. } 5. **public** **void** cetak2(){ 6. System.out.println("final method super class"); 7. } 8. **public** **void** cetak3(){ 9. System.out.println("method dari superclass"); 10. } 11. } |

AdaYangSalah

|  |
| --- |
| 1. **public** **class** AdaYangSalah **extends** SuperClass{ 2. **public** **static** **void** cetak1(){ 3. System.out.println("static method subclass"); 4. } 5. @Override 6. **public** **final** **void** cetak2(){ 7. System.out.println("final method subclass"); 8. } 9. @Override 10. **public** **void** cetak3(){ 11. System.out.println("method dari subclass"); 12. } 13. } |

**Output Program**

|  |
| --- |
|  |

**Soal Ketiga**Buatlah sebuah Abstract Class dengan nama Kucing beserta method-methodnya kemudian buatlah beberapa Class jenis-jenis Kucing yang meng-extends ke Abstract Class Kucing dan buatlah interface dengan nama Kaki, Ekor dan Cakar ke Class Kucing!

**Jawaban**

**Source Code**

|  |
| --- |
| 1. **public** **abstract** **class** Kucing { 2. **public** **abstract** **int** kaki(); 3. **public** **abstract** **boolean** ekor(); 4. **public** **abstract** **boolean** cakar(); 5. } 7. **public** **interface** Ciri { 8. **int** kaki = 4; 9. **boolean** cakar = **true**; 10. **boolean** ekor = **true**; 12. **void** kaki(); 13. **void** cakar(); 14. **void** ekor(); 15. } 17. **public** **class** Anggora **implements** Ciri{ 18. **public** **void** kaki(){ 19. System.out.println("Jumlah Kaki = " + kaki); 20. } 22. **public** **void** cakar(){ 23. System.out.println("Bercakar Tajam = " + cakar); 24. } 26. **public** **void** ekor(){ 27. System.out.println("Berekor Lebat = " + ekor); 28. } 29. } |

**Soal Keempat**Perhatikan Kode Program Berikut :

public interface AdaYangSalah {

default void iniMethod(int iniParam){

System.out.println(iniParam);

}

}

Apa yang salah dari penerapan interface diatas ? coba tuliskan baris kode yang benar !

**Jawaban**

Tidak bisa karena di Class Interface hanya boleh memangill method tanpa parameter, cara mengatasi nya buat class satu lagi yaitu class main yang implements ke class interface AdaYangSalah dan di dalam class main terdapat method dengan parameter dan untuk memanggil nya harus membuat obyek.

**Source Code**

Main

|  |
| --- |
| 1. **public** **class** ad **implements** AdaYangSalah { 2. **int** iniParam = 1; 4. @Override 5. **public** **void** iniMethod() { 6. System.out.println("nm"+iniParam); 7. } 9. **public** **static** **void** main(String[] args) { 10. Ad da = **new** Ad(); 11. da.iniMethod(); 13. } 14. } |

AdaYangSalah

|  |
| --- |
| 1. **public** **interface** AdaYangSalah 2. { 3. **public** **void** iniMethod(); 4. } |

**Output Program**

|  |
| --- |
|  |

**Soal Ke Lima  
*PROGRES STUDI KASUS PROJECT AKHIR :***

Berdasarkan studi kasus kalian cobalah untuk buat/terapkan penggunaan Compile-Time Polymorphism dan Run-Time

Polymorphism serta tambahkan Abstraction tau Interface kedalam project Studi Kasus kalian!

**Jawaban**

**Source Code**

AkunController

|  |
| --- |
| 1. **package** controller; 3. **import** entity.DaftarAkunEntity; 5. **public** **class** AkunController{ 6. **int** indexlogin = 0; 7. **public** AkunController(){ 9. } 10. **public** DaftarAkunEntity getData(){ 11. **return** AllObjectModel.daftarAkunModel.showDataAKun(indexlogin); 12. } 13. **public** **void** daftarKursus(String idakun, String nama, String alamat, String notlp, String pw, String kursus){ 14. AllObjectModel.daftarAkunModel.insertDataDaftarAkun(**new** DaftarAkunEntity(idakun,nama,alamat,notlp,pw,kursus)); 15. } 17. **public** **void** login(String idakun, String pw){ 18. indexlogin = AllObjectModel.daftarAkunModel.cekData(idakun,pw); 19. } 20. **public** **void** viewAkun(){ 21. AllObjectModel.daftarAkunModel.view(); 22. } 24. **public** **void** hapusbuku(**int** index){ 25. AllObjectModel.daftarAkunModel.delete(index-1); 26. } 27. **public** **void** editakun(**int** index, **int** pilih, String data){ 28. AllObjectModel.daftarAkunModel.update(index-1,pilih,data); 29. } 31. } |

AllObjectModel

|  |
| --- |
| 1. **package** controller; 3. **import** model.DaftarAkunModel; 5. **public** **class** AllObjectModel { 6. **public** **static** DaftarAkunModel daftarAkunModel = **new** DaftarAkunModel(); 7. } |

DaftarAkunEntity

|  |
| --- |
| 1. **package** entity; 3. **public** **class** DaftarAkunEntity { 4. **private** String idakun; 5. **private** String nama , alamat , notlp , pw , kursus; 7. **public** String getKursus() { 8. **return** kursus; 9. } 11. **public** **void** setKursus(String kursus) { 12. **this**.kursus = kursus; 13. } 15. **public** DaftarAkunEntity(String idakun, String nama, String alamat, String notlp, String pw, String kursus) { 16. **this**.idakun = idakun; 17. **this**.nama = nama; 18. **this**.alamat = alamat; 19. **this**.notlp = notlp; 20. **this**.pw = pw; 21. **this**.kursus = kursus; 22. } 24. **public** String getIdakun() { 25. **return** idakun; 26. } 28. **public** **void** setIdakun(String idakun) { 29. **this**.idakun = idakun; 30. } 32. **public** String getNama() { 33. **return** nama; 34. } 36. **public** **void** setNama(String nama) { 37. **this**.nama = nama; 38. } 40. **public** String getAlamat() { 41. **return** alamat; 42. } 44. **public** **void** setAlamat(String alamat) { 45. **this**.alamat = alamat; 46. } 48. **public** String getNotlp() { 49. **return** notlp; 50. } 52. **public** **void** setNotlp(String notlp) { 53. **this**.notlp = notlp; 54. } 56. **public** String getPw() { 57. **return** pw; 58. } 60. **public** **void** setPw(String pw) { 61. **this**.pw = pw; 63. } 64. } |

DaftarAkunModel

|  |
| --- |
| 1. **package** model; 3. **import** entity.DaftarAkunEntity; 5. **import** java.util.ArrayList; 7. **public** **class** DaftarAkunModel { 8. **private** ArrayList<DaftarAkunEntity> daftarAkunEntityArrayList; 9. **public** DaftarAkunModel(){ 10. daftarAkunEntityArrayList= **new** ArrayList<DaftarAkunEntity>(); 11. } 12. **public** **void** insertDataDaftarAkun(DaftarAkunEntity daftarakun){ 13. daftarAkunEntityArrayList.add(daftarakun); 14. } 15. **public** ArrayList<DaftarAkunEntity> getDaftarAkunEntityArrayList(){ 16. **return** daftarAkunEntityArrayList; 17. } 18. **public** ArrayList<DaftarAkunEntity> alldatakursus(){ 19. **return** daftarAkunEntityArrayList; 20. } 22. **public** **void** view(){ 23. **for**(DaftarAkunEntity daftarAkun : daftarAkunEntityArrayList){ 24. System.out.println("======================"); 25. System.out.println("ID Akun :"+daftarAkun.getIdakun()); 26. System.out.println("Nama depan : "+daftarAkun.getNama()); 27. System.out.println("Alamat : "+daftarAkun.getAlamat()); 28. System.out.println("Nomer Telephone : "+daftarAkun.getNotlp()); 29. System.out.println("Password: "+daftarAkun.getPw()); 30. System.out.println("Kursus: "+daftarAkun.getKursus()); 31. } 32. } 34. **public** **int** cekData(String idakun, String pw){ 35. **int** loop = 0; 36. **for**(DaftarAkunEntity daftarAkunEntity : daftarAkunEntityArrayList){ 37. **if** (daftarAkunEntity.getIdakun().equals(idakun) && daftarAkunEntity.getPw().equals(pw)){ 38. **break**; 39. } **else** { 40. loop++; 41. } 42. } 43. **return** loop; 44. } 46. **public** **int** search (**int** index) 47. { 48. **if** (index < **this**.daftarAkunEntityArrayList.size()) 49. **return** index; 50. **else** 51. **return** -1; 52. } 54. **public** **void** delete(**int** index) 55. { 56. **if** (search(index)==-1) 57. { 58. System.out.println("data tidak ditemukan"); 59. } 60. **if** (index == -1) 61. { 62. System.out.println("data tidak ditemukan"); 63. } 64. **else** 65. { 66. daftarAkunEntityArrayList.remove(index); 67. System.out.println("berhasil dihapus"); 68. } 69. } 71. **public** **void** update( **int** index, **int** pilih, String data){ 72. **if** (search(index)==-1){ 73. System.out.println("Data tidak ditemukan"); 74. }**else**{ 75. **switch** (pilih){ 76. **case** 1: 77. daftarAkunEntityArrayList.get(index).setIdakun(data); 78. **break**; 79. **case** 2: 80. daftarAkunEntityArrayList.get(index).setNama(data); 81. **break**; 82. **case** 3: 83. daftarAkunEntityArrayList.get(index).setAlamat(data); 84. **break**; 85. **case** 4: 86. daftarAkunEntityArrayList.get(index).setNotlp(data); 87. **break**; 88. **case** 5: 89. daftarAkunEntityArrayList.get(index).setPw(data); 90. **break**; 91. **case** 6: 92. daftarAkunEntityArrayList.get(index).setKursus(data); 93. **break**; 94. **default**: 95. } 96. } 97. } 98. **public** DaftarAkunEntity showDataAKun (**int** index){ 99. **return** daftarAkunEntityArrayList.get(index); 100. } 101. } |

Akun

|  |
| --- |
| 1. **package** view; 3. **import** controller.AllObjectModel; 4. **import** model.DaftarAkunModel; 6. **import** java.util.Scanner; 8. **public** **class** Akun { 10. **private** **static** Scanner input = **new** Scanner(System.in); 12. **public** **static** **void** Daftar(){ 13. String idakun; 14. String nama , alamat , notlp , pw; 15. **int** indexkursus = 0; 16. System.out.print("ID : "); 17. idakun = input.nextLine(); 18. System.out.print("Nama : "); 19. nama = input.nextLine(); 20. System.out.print("Alamat : "); 21. alamat = input.nextLine(); 22. System.out.print("Nomer telephone : "); 23. notlp = input.nextLine(); 24. System.out.print("password : "); 25. pw = input.nextLine(); 26. System.out.println("Menu Kursus : "); 27. System.out.println("- Matematika "); 28. System.out.println("- Biologi "); 29. System.out.println("- Fisika "); 30. System.out.print("Pilih Kursus : "); 31. String kursus = input.next(); 32. AllObjectController.akunController.daftarKursus(idakun,nama,alamat,notlp,pw,kursus); 33. *// AllObjectController.kursusController.dataKursus(kursus);* 35. } 37. **public** **static** **void** login(){ 38. System.out.print("id : "); 39. String idakun = input.next(); 40. System.out.print("Passowrd : "); 41. String pw = input.next(); 42. **try**{ 43. AllObjectController.akunController.login(idakun, pw); 44. menudalam(); 45. } **catch** (Exception e){ 46. System.out.println("Daftar Kursus tidak ada"); 47. } 49. } 50. **public** **static** **void** menu(){ 51. **int** pilih; 52. **do** { 53. System.out.println("Menu Pilihan"); 54. System.out.println("1. Daftar Akun"); 55. System.out.println("2. Login "); 56. System.out.println("3. Exit"); 57. System.out.print("pilih : "); 58. pilih = input.nextInt(); 59. input.nextLine(); 60. **switch** (pilih){ 61. **case** 1: 62. Daftar(); 63. **break**; 64. **case** 2: 65. login(); 66. **break**; 67. } 68. } **while** (pilih != 3); 69. } 71. **public** **static** **void** menudalam(){ 72. **int** pilih; 73. **do** { 74. System.out.println("Menu Pilihan"); 75. System.out.println("1. Edit "); 76. System.out.println("2. Hapus"); 77. System.out.println("3. lihat "); 78. System.out.println("4. Exit"); 79. System.out.print("pilih : "); 80. pilih = input.nextInt(); 81. input.nextLine(); 82. **switch** (pilih){ 83. **case** 1: 84. editbuku(); 85. **break**; 86. **case** 2: 87. deletebuku(); 88. **break**; 89. **case** 3: 90. AllObjectController.akunController.viewAkun(); 91. **break**; 92. } 93. } **while** (pilih != 4); 94. } 96. **public** **static** **void** deletebuku(){ 97. AllObjectController.akunController.viewAkun(); 98. System.out.println("Masukkan nomor akun yang ingin di hapus : "); 99. **int** nomor = input.nextInt(); 100. input.nextLine(); 101. AllObjectController.akunController.hapusbuku(nomor); 102. } 103. **public** **static** **void** editbuku(){ 104. **int** pilih; 106. System.out.print("nomor : "); 107. **int** nomor = input.nextInt(); 108. input.nextLine(); 109. System.out.println("Menu Pilihan"); 110. System.out.println("1. ID Akun "); 111. System.out.println("2. Nama"); 112. System.out.println("3. alamat"); 113. System.out.println("4. nomer telephone "); 114. System.out.println("5. password"); 115. System.out.println("6. Kursus"); 116. System.out.print("pilih : "); 117. pilih = input.nextInt(); 118. input.nextLine(); 119. System.out.println("Masukan Data Baru"); 120. String data = input.nextLine(); 121. **switch** (pilih){ 122. **case** 1: 123. AllObjectController.akunController.editakun(nomor,1,data); 124. **break**; 125. **case** 2: 126. AllObjectController.akunController.editakun(nomor,2,data); 127. **break**; 128. **case** 3: 129. AllObjectController.akunController.editakun(nomor,3,data); 130. **break**; 131. **case** 4: 132. AllObjectController.akunController.editakun(nomor,4,data); 133. **break**; 134. **case** 5: 135. AllObjectController.akunController.editakun(nomor,5,data); 136. **break**; 137. **case** 6: 138. AllObjectController.akunController.editakun(nomor,6,data); 139. **default**:  142. }  145. } 146. } |

AllObjectController

|  |
| --- |
| 1. **package** view; 3. **import** controller.AkunController;  6. **public** **class** AllObjectController { 7. **public** **static** AkunController akunController = **new** AkunController(); 8. } |

**Output Program**

|  |
| --- |
|  |