**,l;JURUSAN INFORMATIKA**

**UNIVERSITAS SANATA DHARMA**

**PRAKTIKUM PBO I**

**MODUL IX**

**Polimorfisme dan Abstrak**

1. **TUJUAN PRAKTIKUM**
2. Mahasiswa mampu memahami konsep polimorfisme.
3. Mahasiswa mampu mengimplementasikan polimorfisme dalam kelas.
4. **DASAR TEORI**

Sebuah method dengan nama yang sama dapat menghasilkan output yang berbeda tergantung dimana lingkungannya berada.

Polimorfisme (*polymorphism*) memungkinkan sebuah variabel tunggal merujuk ke obyek dari sub-kelas yang berbeda-beda dalam hirarki pewarisan tertentu.

Tujuan polimorfisme

* Fleksibilitas dalam pemrograman
* Efisiensi dalam pengembangan program

1. **LANGKAH PERCOBAAN**

1. Ketik dan jalankan program dibawah ini

class Hewan

{

void suara()

{

System.out.println("suara hewan-hewan");

}

}

class Anjing extends Hewan

{

void suara()

{

System.out.println("gug gug");

}

}

class Kucing extends Hewan

{

void suara()

{

System.out.println("meong-meong");

}

}

class Bebek extends Hewan

{

void suara()

{

System.out.println("kwek-kwek");

}

}

class Iseng {

public static void main(String[] args) {

Hewan kewan = new Hewan();

kewan = new Anjing();

kewan.suara();

kewan = new Kucing();

kewan.suara();

kewan = new Bebek();

kewan.suara();

}

}

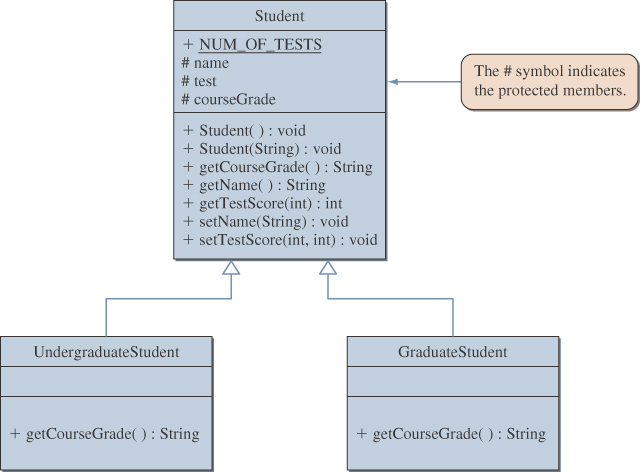
**Analisa dan soal**

1. Buat diagram UML
2. Jelaskan konsep polimorfisme dalam program tersebut
3. Tambahkan method polymorf untuk suara ”harimau” atau hewan favorit anda

2. Lihat diagram UML dibawah ini yang diimplementasikan dalam program

**Diagram kelas UML**

1



**Gambar 9.1 Diagram Kelas UML untuk Kasus Polimorfisme**

**Program**

**Kelas Student**

package polymorf;

public class Student {

protected final static int NUM\_OF\_TESTS = 3;

protected String name;

protected int[] test;

protected String courseGrade;

public Student( ) {

this("No Name");

}

public Student(String studentName) {

name = studentName;

test = new int[NUM\_OF\_TESTS];

courseGrade = "\*\*\*\*";

}

public String getCourseGrade( ) {

return courseGrade;

}

public String getName( ) {

return name;

}

public int getTestScore(int testNumber) {

return test[testNumber-1];

}

public void setName(String newName) {

name = newName;

}

public void setTestScore(int testNumber, int testScore) {

test[testNumber-1] = testScore;

}

}

**Kelas GraduateStudent**

package polymorf;

public class GraduateStudent extends Student {

public String getCourseGrade() {

int total = 0;

for (int i = 0; i < NUM\_OF\_TESTS; i++) {

total += test[i];

}

if (total / NUM\_OF\_TESTS >= 85) {

courseGrade = "Pass";

} else {

courseGrade = "No Pass";

}

return courseGrade;

}

}

**Kelas UndergraduateStudent**

package polymorf;

public class UndergraduateStudent extends Student {

public String getCourseGrade() {

int total = 0;

for (int i = 0; i < NUM\_OF\_TESTS; i++) {

total += test[i];

}

if (total / NUM\_OF\_TESTS >= 80) {

courseGrade = "Pass";

} else {

courseGrade = "No Pass";

}

return courseGrade;

}

}

**Kelas StudentTest**

package polymorf;

public class StudentTest {

public static void main(String[] args)

{

Student[] roster = new Student[40];

roster[0] = new GraduateStudent();

roster[0].setTestScore(1, 70);

roster[0].setTestScore(2, 70);

roster[0].setTestScore(3, 70);

roster[1] = new UndergraduateStudent();

roster[1].setTestScore(1, 70);

roster[1].setTestScore(2, 70);

roster[1].setTestScore(3, 70);

roster[2] = new UndergraduateStudent();

roster[2].setTestScore(1, 82);

roster[2].setTestScore(2, 82);

roster[2].setTestScore(3, 82);

int undergradCount = 0;

for (int i = 0; i < 3; i++)

{

System.out.println(roster[i].getCourseGrade());

}

}

}

**Analisa**

a. Apa output program tersebut

b. Jelaskan KONSEP POLIMORFISME dalam program diatas

1. **TUGAS**

Dengan memakai **sudut pandang polimorfisme** dan **mempelajari kasus program di atas** kerjakan soal latihan berikut ini:

a. Dalam suatu perusahaan PEGAWAI terdiri dari MANAJER, MARKETING dan HONORER.

Penentuan gaji untuk setiap jenis pegawai tersebut berbeda-beda.

* Manajer memiliki gaji pokok + tunjangan+gaji lembur.
* Marketing memiliki gaji pokok + gaji bonus.
* Sedangkan gaji honorer hanya terdiri dari gaji pokok +gaji lembur saja.

Main program pada kelas TESTPEGAWAI

Keterangan

1. Definisikan masing masing kelas beserta atribut dan methodnya.
2. Buat diagram kelas UML.
3. Tunjukkan method polimorfismenya, misal: nama method yang sama untuk menghitung total gaji: HitungGaji().
4. Pada main program terdapat atribut: array of pegawai .
5. Array of pegawai terdiri dari manajer, marketing dan honorer
6. Masukkan data: nama pegawai, jabatan dan komponen gaji ( terdiri dari manajer, marketing dan honorer).
7. Tampilkan data nama pegawai, jabatan dan Hitung total gaji masing-masing obyek dari jabatan manajer, marketing dan honorer.

b. Memakai konsep Polymorphisma, buat program sederhana untuk merekam 3 jenis bentuk bangun datar yakni segi empat, segitiga dan lingkaran dalam sebuah array of BangunDatar. Polymorphisma digunakan untuk menghitung keliling dan luas bangun datar. Bangun datar segi empat beratribut panjang dan lebar, lingkaran beratribut jari-jari dan segitiga beratribut panjang 3 sisinya. Atribut setiap bangun datar dimasukkan lewat keyboard. Output dari program Anda sebaiknya memakai format berikut:

=========================================================

No. Bangun datar Atribut Keliling Luas

-------------------------------------------------------------------------------------------------

1. Lingkaran r = 10 62.8... 314….

2. Segiempat p = 10, l = 5 30.00 50.0

3. Segitiga a = ..., b= ..., c =.... ........ .........

dst….

--------------------------------------------------------------------------------------------------

1. **LAPORAN PRAKTIKUM**

Buatlah laporan dari hasil menjalankan dan menganalisa program Anda dalam modul 9, yang mencakup bagian **LANGKAH PERCOBAAN** dan **TUGAS**. Laporan memuat:

1. Tujuan praktikum (5%)
2. Listing program (hasil *capture screen*) (10%)
3. Output program (hasil *capture screen*) (10%)
4. Algoritma dari main method. (15%)
5. Analisa program yang berisi:
   * Maksud/tujuan umum program (10%)
   * Diagram kelas (10%)
   * Struktur hubungan antara kelas(-kelas) dan main program (10%)
   * Penjelasan atas tiap baris perintah program dalam tiap kelas (10%)
   * Penjelasan keseluruhan jalannya program (10%)
6. Daftar pustaka yang dipakai untuk membuat laporan (10%)

**Daftar Pustaka**

Wu, C. Thomas. 2006. *An Introduction to Object-Oriented Programming with Java*. Mc Graw-Hill International Edition. 4th edition.

👍👍 Seize the day !! Carpe Diem 👍👍