Nama : Sugesty Pramana Aditya

NIM : 1415015042

Macam-macam Keyword pada Java

1. **abstrak**

Abstract class adalah kelas yang terletak pada posisi tertinggi pada hierarki kelas. Kelas ini digunakan sebagai basis bagi penurunan kelas lainnya, sehingga abstract class tidak dapat diinstansiasi secara langsung menjadi object. Suatu abstract class bisa mengandung method kongkrit dan/atau method abstract. Abstract method meruapakan suatu method yang tidak mempunyai badan method. Hanya berupa nama method dan parameter inputan method. Ketika suatu abstract class diturunkan/ di-**extends**, maka subclass harus mengimplementasikan semua method abstract pada parent/ base class. Jika tidak mengimplementasikan semua method abstract maka subclass harus dideklarasikan sebagai abstract class.﻿

Contoh :

       public abstract class MyClass

       {

       }

       public abstract String myMethod();

**2.**      **assert**

Digunakan untuk membuat pernyataan-pernyataan yang selalu benar. Jika pernyataan akan diaktifkan ketika program dijalankan dan ternyata bahwa pernyataan adalah salah, maka AssertionError ditampilkan dan program akan berakhir. Keyword ini dimaksudkan untuk membantu dalam debugging.

Contoh :

assert expression1 [: expression2];

**3.**      **boolean**

boolean digunakan untuk menyatakan variable yang dapat menyimpan nilai boolean yaitu benar atau salah.

Contoh :

       -boolean valid = true;

       if (valid)

       {

       }

       -boolean isComplete = false;

**4.**      **break**

Digunakan untuk melanjutkan eksekusi program ke statement selanjutnya segera setelah selesai mengeksekusi suatu statement.

Contoh :

       for (i=0; i

       {

              if ()

              {

                     break;

              }

       }

       int type = ;

       switch (type)

       {

              case 1:

       break;

              case 2:

       break;

              default:

       }

**5.**      **byte**

Kata kunci byte digunakan untuk menyatakan variable dapat menyimpan data integer sebanyak 8-bit.

Contoh :

byte b = 124;

**6.**      **case**

Digunakan untuk membuat case-case individu dalam pernyataan switch.

Contoh :

       int arg = ;

       switch (arg)

       {

              case 1:

                     break;

              case 2:

                     break;

              default:

                     break;

       }

**7.**      **catch**

sebuah blok satement yang dieksekusi jika pengecualian dijalankan pada blok pendefinisian oleh kata kunci try sebelumnya.

Contoh :

try

{

}

catch ( e)

{

}

try

{

}

catch (FooException e)

{

}

catch (BarException e)

{

}

try

{

}

catch ( e)

{

}

finally

{

}

**8.**      **char**

Kata kunci char digunakan untuk menyatakan variable yang dapat menyimpan data karakter sebanyak 16-bit.

Contoh :

char delimiter = ';';

**9.**      **class**

digunakan untuk mendefinisikan dan mengimplementasikan berbagai macam objek didalamnya.

Contoh :

public class Rectangle

{

       float width;

       float height;

       public Rectangle(float w, float h)

       {

       width = w;

       height = h;

       }

       public float getWidth()

       {

       return width;

       }

       public float getHeight()

       {

       return height;

       }

}

**10.**  **continue**

Digunakan untuk melanjutkan eksekusi program pada akhir pengulangan.

Contoh :

for (i=0; i

{

       if ()

       {

       continue;

       }

}

**11.**  **default**

Default dapat digunakan dalam pernyataan switch untuk label blok pernyataan yang akan dieksekusi jika case tidak sesuai dengan nilai tertentu.

Contoh :

int arg = ;

switch (arg)

       {

              case 1:

                     break;

              case 2:

                     break;

              default:

                     break;

       }

**12.**  **do**

Digunakan bersama dengan while untuk membuat do-while pengulangan, yang mengeksekusi suatu blok statemen berkaitan dengan pengulangan dan kemudian tes ekspresi boolean, jika pernyataan bernilai true, pemgulangan dijalankan lagi, ini terus sampai ekspresi bernilai false.

Contoh :

do

{

}

while (!found);

**13.**  **double**

Digunakan untuk menyatakan variable dapat menampung 64-bit tipe data float.

Contoh :

double ratio = .01;

double diameter = 6.15;

double height = 1.35E03; // 1.35 \* 103 or 1350.0

double height = 1e−2; // 1.0 \* 10−2 or 0.01

**14.**  **else**

Kata kunci lain yang digunakan bersama dengan if untuk membuat pernyataan jika-lain, yang menguji suatu ekspresi boolean, jika ekspresi bernilai true, blok pernyataan terkait dengan jika dievaluasi, jika mengevaluasi ke false blok, dari laporan terkait dengan lain dievaluasi .

Contoh :

if (condition)

{

}

else

{

}

**15.**  **enum** (J2SE 5.0)

Sebuah Java kata kunci yang digunakan untuk mendeklarasikan tipe enumerasi. Enumerations memperpanjang kelas dasar Enum.

Contoh :

Enum Grade { A, B, C, D, F };

//…

Private Grade gradeA = Grade.A;

**16.**  **extends**

Keyword yang digunakan bila terjadi pewarisan, kelas yang mewariskan method dan attributenya disebut kelas super, sedangkan yang diwariskan disebut subkelas.

Contoh :

public class Rectangle extends Polygon

{

}

**17.**  **final**

Dengan keyword ini kelas tidak dapat menurunkan kelas lain, method tidak dapat dioverride oleh method lain, membentuk suatu attribute menjadi konstanta.

Contoh :

public final class MyFinalClass

{

}

public class MyClass

{

       public final String myFinalMethod()

              {

              }

}

**18.**  **finally**

Digunakan untuk mendefinisikan suatu blok statemen untuk blok yang ditetapkan sebelumnya dengan kata kunci try.

Contoh :

try

{

}

catch ( e)

{

}

finally

{

}

**19.**  **float**

Digunakan untuk menyatakan variable dapat menampung 32-bit tipe data decimal.

Contoh :

float ratio = .01;

float diameter = 6.15;

float height = 1.35E03; // 1.35 \* 103 or 1350.0

float height = 1e−2; // 1.0 \* 10−2 or 0.01

**20.**  **for**

Kata kunci untuk digunakan untuk membuat untuk pengulangan, yang menetapkan inisialisasi variabel, ekspresi boolean, dan suatu incrementation. Inisialisasi variabel dilakukan terlebih dahulu, kemudian ekspresi boolean dievaluasi. Jika ekspresi bernilai true, blok pernyataan yang berhubungan dengan pengulangan dijalankan, dan kemudian incrementation yang dilakukan. Ekspresi boolean kemudian dievaluasi lagi, ini terus sampai ekspresi bernilai false .

Contoh :

int i;

for (i=0; i

{

}

**21.**  **goto**

Meskipun tercantum sebagai kata kunci di Java, goto tidak digunakan dan tidak memiliki fungsi.

**22.**  **if**

if digunakan untuk menguji suatu ekspresi boolean, jika ekspresi bernilai true, blok statement yang berhubungan akan dieksekusi. Kata kunci ini juga dapat digunakan untuk membuat pernyataan if-else.

Contoh :

if (condition)

{

}

if (condition)

{

}

else

{

}

**23.**  **implements**

Termasuk dalam sebuah deklarasi kelas untuk menentukan satu atau lebih interface yang diimplementasikan oleh kelas saat ini. Kelas A mewarisi jenis dan metode abstrak dinyatakan oleh interface.

Contoh :

public class Truck implements IVehicle

{

}

**24.**  **import**

Digunakan pada awal sebuah file sumber untuk menentukan kelas atau seluruh paket JAVA untuk disebut kemudian tanpa termasuk paket-paket mereka nama-nama dalam referensi. Sejak J2SE 5.0, pernyataan impor dapat mengimpor anggota statis dari sebuah kelas.

Contoh :

import java.io.File;

import java.net.\*;

**25.**  **instanceof**

Sebuah operator biner yang menggunakan sebuah referensi obyek sebagai operan pertama dan kelas atau antarmuka sebagai operan kedua dan menghasilkan hasil boolean. Operator instanceof bernilai true jika dan hanya jika jenis objek runtime adalah tugas kompatibel dengan kelas atau interface.

Contoh :

if (node instanceof TreeNode)

{

}

**26.**  **int**

Digunakan untuk menyatakan variable dapat menampung 32-bit tipe data bilangan asli.

Contoh :

int number = 5;

int octalNumber = 0377;

int hexNumber = 0xff;

**27.**  **interface**

**Merupakan**kumpulan method yang hanya memuat deklarasi dan struktur method, tanpa detail implementasinya. Sedangkan detail dari method berada pada class yang mengimplementasikan interface tersebut. Interface digunakan bila Anda ingin mengaplikasikan suatu method yang spesifik, yang tidak diperoleh dari proses inheritance. Tipe data yang boleh pada interface hanya tipe data **konstan.**Setelah mengetahui apa itu interface.**long**

Contoh :

public interface IPolygon

{

public float getArea();

public int getNumberOfSides();

public int getCircumference();

}

**28.**  **long**

Long is a Java primitive type. A long variable may store a 64−bit signed integer.

Contoh :

long number = 5;

long anotherNumber = 34590L;

long octalNumber = 0377;

long hexNumber = 0xffl;

**29.**  **native**

Digunakan dalam deklarasi metode untuk menentukan bahwa metode ini tidak diimplementasikan dalam file sumber Java yang sama, melainkan dalam bahasa lain .

Contoh :

native String getProcessorType();

**30.**  **new**

Digunakan untuk membuat sebuah instance dari kelas atau array / obyek.

Contoh :

String sName = new String();

Float fVal = new Float(0.15);

**31.**  **null**

null is a Java reserved word representing no value.

Contoh :

Integer i;

i = null;

String s;

if (s != null)

{

}

**32.**  **package**

Sekelompok jenis. Paket dinyatakan dengan kata kunci paket.

Contoh :

package com.mycompany;

public class MyClass

{

}

**33.**  **private**

Dengan keyword ini kelas/method/attribute tidak bisa diakses oleh kelas lain bahkan juga tidak dapat diturunkan.

Contoh :

package com.mycompany;

public class MyClass

{

}

**34.**  **protected**

Dengan keyword ini kelas/method/attribute dapat diakses oleh kelas lain yang satu package atau kelas lain tersebut merupakan turunannya.

Contoh :

public class MyPublicClass

{

protected class MyPrivateClass

{

}

protected int i;

protected String myMethod()

{

}

}

**35.**  **public**

Merupakan keyword dimana dengan keyword ini kelas/method/attribute dapat diakses oleh kelas lain dimanapun.

Contoh :

public class MyPublicClass

{

public class MyPrivateClass

{

}

public int i;

public String myMethod()

{

}

}

**36.**  **return**

Keyword ini digunakan untuk keluar dari suatu method. Baris – baris program setelah keyword ini yang berada dalam blok method tersebut akan diabaikan. Kemudian eksekusi dilanjutkan ke pernyataan setelah blok method tersebut.

Contoh :

public void myVoidMethod()

{

return;

}

public String myStringMethod()

{

       String s = "my response";

       return s;

}

public int myIntMethod()

{

       int i = 5;

       return(i);

}

**37.**  **short**

Digunakan untuk menyatakan variable dapat menampung 16-bit tipe data integer.

Contoh :

short number = 5;

short octalNumber = 0077;

short hexNumber = 0xff;

**38.**  **static**

Dengan keyword ini method dan attribute milik kelas menjadi sifat bersama dari semua objek dalam kelas tersebut (tidak memerlukan instansiasi dan tidak bisa dioverride).

Contoh :

public class MyPublicClass

{

public final static int MAX\_OBJECTS = 100;

static int \_numObjects = 0;

static class MyStaticClass

{

}

static int getNumObjects()

{

}

}

**39.**  **strictfp** (J2SE 1.2)

Digunakan untuk membatasi presisi dan pembulatan perhitungan floating point untuk memastikan portabilitas.

Contoh :

Public strictfp class Myclass

{

//..

}

Public strictfp void method()

{

...

}

**40.**  **super**

Digunakan untuk mengakses anggota kelas diwariskan oleh kelas di mana ia muncul. Memungkinkan subclass untuk mengakses ditimpa metode dan anggota tersembunyi dari superclassnya. Kata kunci super juga digunakan untuk meneruskan panggilan dari konstruktor ke konstruktor di superclass.

Contoh :

public class MyClass

{

       public MyClass(String arg)

       {

       super(arg);

       }

       public String myStringMethod()

       {

       return super.otherStringMethod();

       }

**41.**  **switch**

Kata kunci switch digunakan dengan case dan default, yang akan memeriksa sebuah variabel, dan mengeksekusi blok pernyataan yang sama dengan case tersebut.

Contoh :

int arg = ;

switch (arg)

{

       case 1:

              break;

       case 2:

              break;

       default:

              break;

}

char arg = ;

switch (arg)

{

       case 'y':

       case 'Y':

              break;

       case 'n':

       case 'N':

              break;

       default:

              break;

}

**42.**  **synchronized**

Digunakan dalam deklarasi metode atau blok kode untuk memperoleh kunci mutex untuk objek while thread saat mengeksekusi kode . Untuk metode statis, objek terkunci adalah Kelas kelas. Jaminan bahwa paling banyak satu thread pada waktu operasi pada obyek yang sama mengeksekusi kode tersebut. Kunci mutex secara otomatis dilepaskan ketika keluar eksekusi kode disinkronisasi. Fields, kelas dan interface tidak dapat dinyatakan sebagai disinkronisasi.

Contoh :

public class MyClass

{

public synchronized static String mySyncStaticMethod()

{

}

public synchronized String mySyncMethod()

{

}

{

public class MyOtherClass

{

Object someObj;

public String myMethod()

{

synchronized (someObj)

{

}

}

}

**43.**  **this**

Keyword yang digunakan untuk menyatakan objek sekarang dan tanpa instansiasi.

Contoh :

public class MyClass

{

int number;

public MyClass(int number)

{

this.number = number;

}

}

**44.**  **throw**

Menyebabkan deklarasi pengecualian. Hal ini menyebabkan eksekusi untuk melanjutkan dengan pengecualian penangan melampirkan pertama dinyatakan oleh kata kunci catch untuk menangani tipe pengecualian tugas kompatibel. Jika tidak ada handler pengecualian tersebut ditemukan dalam metode saat ini, maka kembali metode dan proses ini diulang dalam metode panggilan. Jika tidak ada handler pengecualian ditemukan dalam setiap pemanggilan metode di stack, maka pengecualian dilewatkan ke handler pengecualian tidak tertangkap thread ini.

Contoh :

import java.io.IOException;

public class MyClass

{

public method readFile(String filename) throws IOException

{

if (error)

{

throw new IOException("error reading file");

}

}

}

**45.**  **throws**

Digunakan dalam deklarasi metode untuk menentukan pengecualian tidak ditangani dalam metode melainkan diteruskan ke tingkat yang lebih tinggi berikutnya program. Semua pengecualian tidak tertangkap dalam sebuah metode yang tidak case RuntimeException harus dideklarasikan menggunakan throws kata kunci.

Contoh :

import java.io.IOException;

public class MyClass

{

public method readFile(String filename) throws IOException

{

if (error)

{

throw new IOException("error reading file");

}

}

}

**46.**  **transient**

Menyatakan bahwa medan Contoh bukan bagian dari default serial bentuk objek. Ketika suatu objek serial, hanya nilai-nilai non-fana bidang misalnya dimasukkan dalam representasi serial default. Ketika suatu objek deserialized, while bidang diinisialisasi hanya untuk nilai default mereka. Jika bentuk standar tidak digunakan, misalnya ketika sebuah tabel serialPersistentFields ini dideklarasikan pada hirarki kelas, semua kata kunci while diabaikan.

Contoh :

public class MyClass

{

private transient String password;

}

**47.**  **try**

Mendefinisikan suatu blok statemen yang memiliki penanganan eksepsi. Jika eksepsi dilemparkan di dalam blok try, sebuah blok catch opsional dapat menangani jenis pengecualian dideklarasikan. Juga, sebuah blok akhirnya opsional dapat dideklarasikan yang akan dijalankan ketika eksekusi keluar dari blok mencoba dan menangkap klausa, terlepas dari apakah eksepsi dilemparkan atau tidak. Sebuah try blok harus memiliki minimal satu klausa catch atau finally blok.

Contoh :

try

{

}

catch ( e)

{

}

try

{

}

catch (FooException e)

{

}

catch (BarException e)

{

}

try

{

}

catch ( e)

{ }

finally

{

}

**48.**  **void**

Keyword yang merupakan tipe jenis return value dimana method yang menggunakan keyword ini tidak mengembalikan nilai apapaun setelah dipanggil atau dieksekusi.

Contoh :

public class MyClass

{

public void doSomething();

{

return;

}

}

**49.**  **volatile**

Digunakan dalam deklarasi lapangan untuk menentukan bahwa variabel yang diubah asynchronously oleh bersamaan menjalankan thread. Metode, kelas dan interface sehingga tidak dapat dideklarasikan volatile.

Contoh :

public class MyClass

{

volatile int sharedValue;

}

**50.**  **while**

Kata kunci while digunakan untuk membuat pengulangan while, yang menguji suatu ekspresi boolean dan mengeksekusi blok pernyataan yang berhubungan dengan pengulangan jika ekspresi bernilai true; sampai ekspresi bernilai salah. Kata kunci ini juga dapat digunakan untuk membuat pengulangan do-while.

Contoh :

while (!found)

{

}