NAMA : ANDIKA PRATAMA

NIM : 13020220009

KELAS : A1

KODE PROGRAM 1

public class Asign {

/\*\* \*

@param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* Kamus \*/

float f= 20.0f;

double fll; /\*

Algoritma \*/

fll=10.0f; S

ystem.out.println ("f : "+f+ "\nf11: "+fll);

}

}

Kode program di atas adalah sebuah program sederhana dalam bahasa pemrograman Java yang mencetak nilai dari dua variabel f dan fll. Berikut penjelasan lebih detailnya:

1. Deklarasi Variabel:

* float f= 20.0f;: Mendeklarasikan variabel f dengan tipe data float dan memberikan nilai awal 20.0.
* double fll;: Mendeklarasikan variabel fll dengan tipe data double tanpa memberikan nilai awal.

1. Algoritma:

* fll=10.0f;: Memberikan nilai 10.0 ke variabel fll.
* System.out.println ("f : "+f+ "\nf11: "+fll);: Mencetak nilai dari variabel f dan fll ke layar. Variabel f dicetak dengan format "f : "+f, sedangkan variabel fll dicetak dengan format "f11: "+fll.

1. Output:

Output yang dihasilkan dari program ini akan mencetak nilai variabel f dan fll ke layar dengan format yang sesuai dengan instruksi cetak pada baris terakhir dari program.

Jadi, keseluruhan program ini hanya melakukan inisialisasi variabel dan mencetak nilainya ke layar.

Kode Program 2

public class Asign

/ public static void main(String[] args) {

// TODO Auto-generated method stub

/\* Kamus \*/

int i;

/\* Program \*/

System.out.print ("hello\n"); i = 5; System.out.println ("Ini nilai i : " + i);

}

}

Kode program di atas adalah program sederhana dalam bahasa pemrograman Java yang mencetak pesan "hello" ke layar dan kemudian mencetak nilai dari sebuah variabel i. Berikut adalah penjelasan lebih detailnya:

1. Deklarasi Variabel:

* int i;: Mendeklarasikan variabel i dengan tipe data integer tanpa memberikan nilai awal.

1. Algoritma:

* System.out.print ("hello\n");: Menggunakan System.out.print untuk mencetak pesan "hello" ke layar tanpa ganti baris.
* i = 5;: Memberikan nilai 5 ke variabel i.
* System.out.println ("Ini nilai i : " + i);: Menggunakan System.out.println untuk mencetak pesan "Ini nilai i : " beserta nilai dari variabel i ke layar dengan ganti baris.

Kode Program 3

/\* Deskripsi : \*/

/\* Program ini berisi contoh sederhana untuk mendefinisikan \*/

/\* variabel-variabel bilangan bulat (short int, int, long int),

\*/ /\* karakter, bilangan riil, \*/

public class ASIGNi {

public static void main(String[] args) { //

TODO Auto-generated method stub

/\* KAMUS \*/

short ks = 1;

int ki = 1;

long kl = 10000;

char c = 65; /\* inisialisasi karakter dengan integer \*/

char c1 = 'Z'; /\* inisialisasi karakter dengan karakter \*/

double x = 50.2f;

float y = 50.2f;

/\* Algoritma \*/

/\* penulisan karakter sebagai karakter \*/

System.out.println("Karakter = "+ c);

System.out.println("Karakter = "+ c1);

/\* penulisan karakter sebagai integer \*/

System.out.println("Karakter = "+ c);

System.out.println("Karakter = "+ c1);

System.out.println ("Bilangan integer (short) = "+ ks);

System.out.println("\t(int) = "+ ki);

System.out.println("\t(long)= "+ kl);

System.out.println("Bilangan Real x = "+ x);

System.out.println("Bilangan Real y = "+ y);

}

PENJELASAN KODE :

1. Deklarasi Variabel:

* short ks = 1;, int ki = 1;, long kl = 10000;: Mendeklarasikan variabel-variabel dengan tipe data bilangan bulat (short, int, dan long) dan memberikan nilai awal masing-masing variabel.
* char c = 65;: Mendeklarasikan variabel c dengan tipe data karakter (char) dan memberikan nilai awal berupa integer 65. Dalam sistem ASCII, 65 mewakili karakter 'A'.
* char c1 = 'Z';: Mendeklarasikan variabel c1 dengan tipe data karakter (char) dan memberikan nilai awal berupa karakter 'Z'.
* double x = 50.2f;, float y = 50.2f;: Mendeklarasikan variabel-variabel dengan tipe data bilangan riil (double dan float) dan memberikan nilai awal masing-masing variabel.

1. Algoritma:

Program mencetak nilai variabel-variabel ke layar menggunakan perintah System.out.println().

Ada dua bagian dalam mencetak karakter ke layar:

* Pertama, karakter dicetak sebagai karakter dengan menggunakan format "Karakter = " + c.
* Kedua, karakter juga dicetak sebagai integer dengan menggunakan format "Karakter = " + (int) c. Ini menunjukkan bahwa dalam pemrograman Java, sebuah karakter dapat dianggap sebagai integer sesuai dengan nilai ASCII-nya.
* Setelah itu, nilai variabel bilangan bulat dan bilangan riil juga dicetak ke layar.
* Pada saat mencetak karakter sebagai integer, dilakukan penulisan karakter c dan c1 yang sama dengan sebelumnya. Ini menunjukkan bahwa nilai ASCII dari karakter 'A' dan 'Z' dicetak ke layar.

Kode Program 4

import java.util.Scanner;

/\* contoh membaca integer menggunakan Class Scanner\*/

public class BacaData {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* Kamus \*/

int a;

Scanner masukan;

/\* Program \*/

System.out.print ("Contoh membaca dan menulis, ketik nilai integer: \n");

masukan = new Scanner(System.in);

a = masukan.nextInt(); /\* coba ketik :

masukan.nextInt();

Apa akibatnya ?\*/

System.out.print ("Nilai yang dibaca : "+ a);

}

}

PENJELASAN KODE :

1. Impor Kelas Scanner:

* Kode program dimulai dengan pernyataan import java.util.Scanner;, yang mengimpor kelas Scanner dari paket java.util, sehingga kelas Scanner dapat digunakan dalam program.

1. Deklarasi Kelas dan Method main:

* public class BacaData: Mendefinisikan kelas BacaData.
* public static void main(String[] args): Mendefinisikan method main, yang merupakan titik masuk utama dari program.

1. Deklarasi Variabel:

* int a;: Mendeklarasikan variabel a dengan tipe data integer untuk menyimpan nilai yang akan dimasukkan oleh pengguna.
* Scanner masukan;: Mendeklarasikan variabel masukan dengan tipe data Scanner yang akan digunakan untuk membaca input dari pengguna.

1. Membaca Input dari Pengguna:

* System.out.print("Contoh membaca dan menulis, ketik nilai integer: \n");: Mencetak pesan ke layar untuk memberitahu pengguna bahwa program sedang meminta input integer.
* masukan = new Scanner(System.in);: Membuat objek Scanner baru dengan menggunakan konstruktor Scanner(System.in), yang menghubungkan Scanner dengan input dari konsol.
* a = masukan.nextInt();: Memanggil method nextInt() dari objek masukan untuk membaca input integer dari pengguna dan menyimpannya ke dalam variabel a.

1. Mencetak Nilai yang Dibaca:

* System.out.print("Nilai yang dibaca: " + a);: Mencetak nilai yang telah dibaca dari pengguna ke layar.

1. Komentar:

* Kode program juga berisi komentar yang memberikan beberapa instruksi dan menjelaskan tindakan-tindakan tertentu kepada pembaca.

1. Eksekusi:

* Ketika program dieksekusi, pesan instruksi akan dicetak ke layar. Pengguna diminta untuk memasukkan nilai integer melalui konsol.
* Setelah pengguna memasukkan nilai, nilai tersebut akan disimpan dalam variabel a.
* Nilai yang telah dibaca akan dicetak kembali ke layar.

Kode Program 5

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

//import javax.swing.\*;

public class Bacakar {

/public static void main(String[] args) throws IOException

{ // TODO Auto-generated method stub

/\* Kamus \*/

char cc; int bil;

InputStreamReader isr = new

InputStreamReader(System.in);

BufferedReader dataIn = new

BufferedReader(isr); // atau

BufferedReader datAIn = new BufferedReader(new InputStreamReader(System.in));

/\* Algoritma \*/

System.out.print ("hello\n");

System.out.print("baca 1 karakter : ");

//perintah baca karakter cc

cc =dataIn.readLine().charAt(0);

System.out.print("baca 1 bilangan : ");

//perintah baca bil

bil =Integer.parseInt(datAIn.readLine());

/\*String kar = JOptionPane.showInputDialog("Karakter 1 : "); System.out.println(kar);\*/

//JOptionPane.showMessageDialog(null,"hello");

System.out.print (cc +"\n" +bil+"\n");

System.out.print ("bye \n");

/\* Tambahkan program membaca/input data menggunakan Class Scanner, Class Console dan Class JOptionPane \*/

PENJELASAN KODE :

1. Impor Kelas BufferedReader dan InputStreamReader:

Kode program dimulai dengan pernyataan import java.io.BufferedReader; dan import java.io.InputStreamReader;, yang mengimpor kelas-kelas yang dibutuhkan untuk membaca input dari pengguna melalui konsol.

1. Deklarasi Kelas dan Method main:

public class Bacakar: Mendefinisikan kelas Bacakar.

public static void main(String[] args) throws IOException: Mendefinisikan method main yang melemparkan IOException.

1. Deklarasi Variabel:

char cc; int bil;: Mendeklarasikan variabel cc dengan tipe data char dan variabel bil dengan tipe data int untuk menyimpan input dari pengguna.

1. Membaca Input dari Pengguna:

InputStreamReader isr = new InputStreamReader(System.in);: Membuat objek InputStreamReader yang terhubung ke input stream standar (System.in).

BufferedReader dataIn = new BufferedReader(isr);: Membuat objek BufferedReader yang digunakan untuk membaca input dari InputStreamReader.

cc = dataIn.readLine().charAt(0);: Membaca satu baris input dari pengguna dan mengambil karakter pertama dari baris tersebut.

bil = Integer.parseInt(dataIn.readLine());: Membaca satu baris input dari pengguna dan mengonversi input tersebut menjadi bilangan integer menggunakan method parseInt() dari kelas Integer.

1. Mencetak Input:

System.out.print(cc +"\n" + bil + "\n");: Mencetak karakter dan bilangan yang telah dibaca dari pengguna ke layar.

1. Output:

Output yang dihasilkan adalah karakter dan bilangan yang dimasukkan oleh pengguna, kemudian diikuti dengan pesan "bye".

1. Komentar:

Kode program juga berisi komentar yang memberikan beberapa instruksi dan menjelaskan tindakan-tindakan tertentu kepada pembaca.

1. IOException Handling:

Method main melemparkan IOException, yang menangkap kesalahan jika terjadi masalah saat membaca input dari pengguna.

1. Penambahan Program:

Ada komentar di akhir program yang menginstruksikan untuk menambahkan program yang membaca/input data menggunakan kelas Scanner, Console, dan JOptionPane.

Kode Program 6

/\*Casting menggunakan tipe data primitif\*

/ public class Casting1 {

public static void main(String[] args) {

// TODO Auto-generated method stub

int a=5, b=6;

float d=2.f,e=3.2f;

char g='5';

double k=3.14;

System.out.println((float)a); // int <-- float

System.out.println((double)b); // int <-- double

System.out.println((int)d); // float <-- int

System.out.println((double)e); // float <-- double

System.out.println((int)g); // char <-- int (ASCII)

System.out.println((float)g); // char <-- float (ASCII)

System.out.println((double)g); // char <-- double (ASCII)

System.out.println((int)k); // double <-- int

System.out.println((float)k); // double <-- float

}

}

PENJELASAN KODE :

1. Kode program di atas adalah contoh penggunaan operator casting untuk mengkonversi nilai antara tipe data primitif dalam bahasa pemrograman Java. Berikut adalah penjelasan mengenai setiap baris kode:
2. int a=5, b=6;: Mendefinisikan dua variabel a dan b dengan tipe data int dan memberikan nilai awal masing-masing 5 dan 6.
3. float d=2.f, e=3.2f;: Mendefinisikan dua variabel d dan e dengan tipe data float dan memberikan nilai awal masing-masing 2.0 dan 3.2. Penambahan huruf 'f' pada nilai angka mengindikasikan bahwa nilai tersebut adalah float.
4. char g='5';: Mendefinisikan variabel g dengan tipe data char dan memberikan nilai awal karakter '5'.
5. double k=3.14;: Mendefinisikan variabel k dengan tipe data double dan memberikan nilai awal 3.14.
6. System.out.println((float)a);: Mengkonversi nilai variabel a (int) menjadi tipe data float dan mencetaknya ke layar.
7. System.out.println((double)b);: Mengkonversi nilai variabel b (int) menjadi tipe data double dan mencetaknya ke layar.
8. System.out.println((int)d);: Mengkonversi nilai variabel d (float) menjadi tipe data int dan mencetaknya ke layar.
9. System.out.println((double)e);: Mengkonversi nilai variabel e (float) menjadi tipe data double dan mencetaknya ke layar.
10. System.out.println((int)g);: Mengkonversi nilai variabel g (char) menjadi tipe data int dengan menggunakan nilai ASCII dari karakter '5' dan mencetaknya ke layar.
11. System.out.println((float)g);: Mengkonversi nilai variabel g (char) menjadi tipe data float dengan menggunakan nilai ASCII dari karakter '5' dan mencetaknya ke layar.
12. System.out.println((double)g);: Mengkonversi nilai variabel g (char) menjadi tipe data double dengan menggunakan nilai ASCII dari karakter '5' dan mencetaknya ke layar.
13. System.out.println((int)k);: Mengkonversi nilai variabel k (double) menjadi tipe data int dan mencetaknya ke layar.
14. System.out.println((float)k);: Mengkonversi nilai variabel k (double) menjadi tipe data float dan mencetaknya ke layar.

Kode Program 7

/\*Casting menggunakan tipe data Class\*/

public class Casting2 {

public static void main(String[] args) {

// TODO Auto-generated method stub

int a=8,b=9;

float d=2.f,e=3.2f;

char g='5';

double k=3.14;

String n="67",m="45", l="100";

a = Integer.parseInt(n); /\*Konversi String ke Integer\*/

k = Double.parseDouble(m); /\*Konversi String ke Double\*/

d = Float.parseFloat(l); /\*Konversi String ke Float\*/

System.out.println("a : "+a+"\nk : "+k+"\nd : "+d);

n = String.valueOf(b); /\*Konversi Integer ke String\*/

m = String.valueOf(g); /\*Konversi Karakter ke String\*/

l = String.valueOf(e); /\*Konversi Float ke String\*/

System.out.println("n : "+n+"\nm : "+m+"\nl : "+l);

k = Double.valueOf(a).intValue();

/\*Konversi Integer ke Double\*/

double c = Integer.valueOf(b).doubleValue();

System.out.println("k : "+k+"\nc : "+c+"\nl : "+l);

PENJELASAN KODE :

1. int a=8,b=9;: Mendeklarasikan dua variabel a dan b dengan tipe data int dan memberikan nilai awal masing-masing 8 dan 9.
2. float d=2.f,e=3.2f;: Mendeklarasikan dua variabel d dan e dengan tipe data float dan memberikan nilai awal masing-masing 2.0 dan 3.2. Penambahan huruf 'f' pada nilai angka mengindikasikan bahwa nilai tersebut adalah float.
3. char g='5';: Mendeklarasikan variabel g dengan tipe data char dan memberikan nilai awal karakter '5'.
4. double k=3.14;: Mendeklarasikan variabel k dengan tipe data double dan memberikan nilai awal 3.14.
5. String n="67", m="45", l="100";: Mendeklarasikan tiga variabel string n, m, dan l dan memberikan nilai awal masing-masing "67", "45", dan "100".
6. a = Integer.parseInt(n);: Mengkonversi nilai string variabel n menjadi tipe data int menggunakan method parseInt() dari kelas Integer.
7. k = Double.parseDouble(m);: Mengkonversi nilai string variabel m menjadi tipe data double menggunakan method parseDouble() dari kelas Double.
8. d = Float.parseFloat(l);: Mengkonversi nilai string variabel l menjadi tipe data float menggunakan method parseFloat() dari kelas Float.
9. n = String.valueOf(b);: Mengkonversi nilai variabel b (int) menjadi tipe data String menggunakan method valueOf() dari kelas String.
10. m = String.valueOf(g);: Mengkonversi nilai variabel g (char) menjadi tipe data String menggunakan method valueOf() dari kelas String.
11. l = String.valueOf(e);: Mengkonversi nilai variabel e (float) menjadi tipe data String menggunakan method valueOf() dari kelas String.
12. k = Double.valueOf(a).intValue();: Mengkonversi nilai variabel a (int) menjadi tipe data Double, kemudian dikonversi kembali menjadi int menggunakan method intValue() dari kelas Double.
13. double c = Integer.valueOf(b).doubleValue();: Mengkonversi nilai variabel b (int) menjadi tipe data Integer, kemudian dikonversi kembali menjadi double menggunakan method doubleValue() dari kelas Integer.
14. System.out.println("a : "+a+"\nk : "+k+"\nd : "+d);: Mencetak nilai variabel a, k, dan d ke layar.
15. System.out.println("n : "+n+"\nm : "+m+"\nl : "+l);: Mencetak nilai variabel n, m, dan l ke layar.
16. System.out.println("k : "+k+"\nc : "+c+"\nl : "+l);: Mencetak nilai variabel k, c, dan l ke layar.
17. Kode Program

/\* pemakaian operator kondisional \*/

public class Ekspresi {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* KAMUS \*/

int x = 1;

int y = 2;

/\* ALGORITMA \*/

System.out.print("x = "+ x + "\n");

System.out.print("y = "+ y + "\n");

System.out.print("hasil ekspresi = (x<y)?x:y = " +

((x < y) ? x : y));

/\*Gunakan dalam kurung "(statemen dan kondisi)" untuk menyatakan satu kesatuan pernyataan\*/

Kode program di atas merupakan contoh penggunaan operator kondisional (ternary) dalam bahasa pemrograman Java. Berikut adalah penjelasan mengenai kode program tersebut:

int x = 1;: Mendefinisikan variabel x dengan tipe data integer dan memberikan nilai awal 1.

int y = 2;: Mendefinisikan variabel y dengan tipe data integer dan memberikan nilai awal 2.

ALGORITMA:

System.out.print("x = "+ x + "\n");: Mencetak nilai variabel x ke layar.

System.out.print("y = "+ y + "\n");: Mencetak nilai variabel y ke layar.

System.out.print("hasil ekspresi = (x<y)?x:y = " + ((x < y) ? x : y));: Ini adalah ekspresi menggunakan operator kondisional. Ekspresi (x<y)?x:y berarti jika nilai x kurang dari nilai y, maka hasilnya adalah nilai x, jika tidak, maka hasilnya adalah nilai y. Hasil dari ekspresi ini kemudian dicetak ke layar.

Komentar di dalam kode menjelaskan penggunaan tanda kurung dalam (x<y)?x:y untuk menyatakan satu kesatuan pernyataan.

Jadi, keseluruhan kode program ini menggunakan operator kondisional untuk mengevaluasi perbandingan antara dua variabel x dan y, dan mencetak hasilnya ke layar.

1. Kode Program

/\* pembagian integer, casting \*/

public class Ekspresi1 {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* KAMUS \*/

int x = 1; int y = 2; float fx; float fy;

/\* ALGORITMA \*/

System.out.print ("x/y (format integer) = "+ x/y);

System.out.print ("\nx/y (format float) = "+ x/y);

/\* supaya hasilnya tidak nol \*/

fx=x;

fy=y;

System.out.print ("\nx/y (format integer) = "+ fx/fy);

System.out.print ("\nx/y (format float) = "+ fx/fy);

/\* casting \*/

System.out.print("\nfloat(x)/float(y) (format integer)=

"+ (float)x/(float)y);

System.out.print("\nfloat(x)/float(y) (format float) =

"+ (float)x/(float)y);

x = 10; y = 3;

System.out.print ("\nx/y (format integer) = "+ x/y); System.out.print ("\nx/y (format float = "+ x/y);

Kode program di atas adalah contoh penggunaan operator pembagian dalam bahasa pemrograman Java, serta penggunaan operator casting untuk mengkonversi tipe data. Berikut adalah penjelasan mengenai kode program tersebut:

Pendefinisian Variabel:

int x = 1; int y = 2;: Mendefinisikan dua variabel x dan y dengan tipe data integer dan memberikan nilai awal masing-masing 1 dan 2.

float fx; float fy;: Mendefinisikan dua variabel fx dan fy dengan tipe data float tanpa memberikan nilai awal.

Operasi Pembagian dan Cetak Hasil:

System.out.print ("x/y (format integer) = "+ x/y);: Mencetak hasil pembagian x dan y dalam format integer ke layar. Perhatikan bahwa ini adalah pembagian integer, sehingga hasilnya akan menghasilkan angka bulat.

System.out.print ("\nx/y (format float) = "+ x/y);: Mencetak hasil pembagian x dan y dalam format float ke layar. Karena x dan y adalah integer, hasil pembagian juga akan menjadi integer.

Menggunakan Variabel Float dan Cetak Hasil:

fx=x; fy=y;: Mengassign nilai dari x ke fx dan nilai dari y ke fy. Ini bertujuan untuk menggunakan tipe data float dalam pembagian selanjutnya.

System.out.print ("\nx/y (format integer) = "+ fx/fy);: Mencetak hasil pembagian fx dan fy dalam format integer ke layar. Karena fx dan fy adalah float, hasil pembagian akan menghasilkan float.

System.out.print ("\nx/y (format float) = "+ fx/fy);: Mencetak hasil pembagian fx dan fy dalam format float ke layar.

Penggunaan Operator Casting:

(float)x/(float)y: Menggunakan operator casting untuk mengkonversi nilai x dan y ke tipe data float sebelum dilakukan pembagian. Ini dilakukan untuk memastikan pembagian dilakukan dalam format float.

System.out.print("\nfloat(x)/float(y) (format integer)= "+ (float)x/(float)y);: Mencetak hasil pembagian x dan y dalam format integer ke layar setelah dilakukan casting ke tipe data float.

Operasi Pembagian Kembali Setelah Mengubah Nilai Variabel:

Setelah mengubah nilai x menjadi 10 dan y menjadi 3, dilakukan lagi operasi pembagian dan hasilnya dicetak ke layar. Namun, penjelasan detail mengenai operasi ini tidak diberikan dalam komentar.

Jadi, keseluruhan kode program ini mendemonstrasikan pembagian integer dan float, serta penggunaan operator casting untuk memastikan pembagian dilakukan dalam format yang diinginkan.

1. Kode Program

public class PrintHello {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* menuliskan hello ke layar \*/

System.out.print("Hello");

/\* menuliskan hello dan ganti baris\*/

System.out.print("\nHello ");

/\* menuliskan hello dan ganti baris\*/

System.out.println("World"); S

ystem.out.println("Welcome");

Kode program di atas adalah sebuah program sederhana yang mencetak teks ke layar menggunakan metode System.out.print() dan System.out.println() dalam bahasa pemrograman Java. Berikut adalah penjelasan mengenai kode program tersebut:

Mencetak "Hello" tanpa Ganti Baris:

System.out.print("Hello");: Mencetak teks "Hello" ke layar tanpa melakukan perubahan baris.

Mencetak "Hello" dengan Ganti Baris:

System.out.print("\nHello ");: Mencetak teks "Hello" ke layar dengan melakukan perubahan baris sebelumnya menggunakan karakter escape sequence \n. Setelah itu, teks "Hello " dicetak tanpa perubahan baris.

Mencetak "World" dengan Ganti Baris Menggunakan println():

System.out.println("World");: Mencetak teks "World" ke layar dan secara otomatis melakukan perubahan baris setelah mencetak teks tersebut karena menggunakan println().

Mencetak "Welcome" dengan Ganti Baris Menggunakan println():

System.out.println("Welcome");: Mencetak teks "Welcome" ke layar dan secara otomatis melakukan perubahan baris setelah mencetak teks tersebut karena menggunakan println().

Dengan demikian, program ini mencetak beberapa baris teks ke layar dengan menggunakan metode print() dan println(), serta memanfaatkan karakter escape sequence \n untuk melakukan perubahan baris secara manual.

1. Kode Program

/\* Effek dari operator ++ \*/

public class Incr {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* Kamus \*/

int i, j;

/\* Program \*/

i = 3;

j = i++;

System.out.println ("Nilai i : " + (++i) + "\nNilai j : " + j); } }

Kode program di atas adalah sebuah contoh untuk memahami efek dari operator penambahan ++ (increment) dalam bahasa pemrograman Java. Berikut adalah penjelasan mengenai kode program tersebut:

Pendefinisian Variabel:

int i, j;: Mendefinisikan dua variabel i dan j dengan tipe data integer.

Pemberian Nilai Awal:

i = 3;: Memberikan nilai awal 3 kepada variabel i.

Penggunaan Operator ++:

j = i++;: Memasukkan nilai dari variabel i ke dalam variabel j, kemudian nilai variabel i akan ditambah 1. Ini disebut dengan post-increment, artinya nilai akan diinkremen setelah nilai variabel diberikan ke variabel lain.

System.out.println ("Nilai i : " + (++i) + "\nNilai j : " + j);: Mencetak nilai variabel i yang telah ditambahkan 1 menggunakan pre-increment (nilai akan diinkremen sebelum digunakan) dan nilai variabel j yang telah diberikan nilai sebelumnya (nilai sebelum diinkremen).

Hasil Output:

Output program ini akan mencetak:

Nilai i : 5: Ini adalah nilai variabel i setelah operasi pre-increment (++i), yang telah ditambahkan 1 dari nilai sebelumnya (3 + 1 = 4), kemudian menjadi 5.

Nilai j : 3: Ini adalah nilai variabel j yang telah diberikan nilai sebelumnya, yaitu nilai i sebelum diinkremen. Oleh karena itu, nilai j adalah 3.

Dengan demikian, kode program tersebut menjelaskan bagaimana operator penambahan ++ bekerja dalam konteks post-increment dan pre-increment, serta bagaimana penggunaannya mempengaruhi nilai variabel yang terlibat.

1. Kode Program

/\* pemakaian beberapa operator terhadap bit \*/

public class Oper1 {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* KAMUS \*/

int n = 10; /\* 1010 \*/

int x = 1; /\* 1 \*/

int y = 2; /\* 10 \*/

/\* ALGORITMA \*/

System.out.println ("n = "+ n);

System.out.println ("x = "+ x);

System.out.println ("y = "+ y);

System.out.println("n & 8 = "+ (n & 8)); /\* 1010 AND 1000 \*/

System.out.println ("x & ~ 8 = "+ (x & ~8)); /\* 1 AND 0111 \*/

System.out.println ("y << 2 = "+ (y << 2)); /\*10==> 1000 = 8 \*/

System.out.println ("y >> 3 = "+ (y >>3)); /\*10 ==>0000 = 0 \*/

Kode program di atas adalah contoh pemakaian beberapa operator bit dalam bahasa pemrograman Java. Berikut penjelasan mengenai kode program tersebut:

Pendefinisian Variabel:

int n = 10;: Mendefinisikan variabel n dengan nilai 10 yang dalam representasi biner adalah 1010.

int x = 1;: Mendefinisikan variabel x dengan nilai 1 yang dalam representasi biner adalah 0001.

int y = 2;: Mendefinisikan variabel y dengan nilai 2 yang dalam representasi biner adalah 0010.

Operasi-operasi Bit:

System.out.println("n & 8 = "+ (n & 8));: Melakukan operasi bitwise AND antara nilai n (1010) dengan nilai 8 (1000). Hasilnya adalah 1000.

System.out.println ("x & ~ 8 = "+ (x & ~8));: Melakukan operasi bitwise AND antara nilai x (0001) dengan kebalikan (NOT) dari nilai 8 (1000). Keadaan bit yang di-NOT dari 8 adalah 0111, sehingga hasil operasi adalah 0001 & 0111 = 0001.

System.out.println ("y << 2 = "+ (y << 2));: Melakukan operasi bitwise shift left pada nilai y (0010) sebanyak 2 kali. Hal ini akan menggeser bit ke kiri sebanyak 2 langkah, sehingga menghasilkan 1000 yang setara dengan nilai desimal 8.

System.out.println ("y >> 3 = "+ (y >>3));: Melakukan operasi bitwise shift right pada nilai y (0010) sebanyak 3 kali. Hal ini akan menggeser bit ke kanan sebanyak 3 langkah, sehingga menghasilkan 0000 yang setara dengan nilai desimal 0.

Output:

Program akan mencetak hasil operasi-operasi bit yang dilakukan pada layar, yaitu nilai-nilai hasil operasi.

Dengan demikian, kode program tersebut adalah contoh penggunaan operator-operasi bit dalam bahasa pemrograman Java untuk melakukan manipulasi data pada tingkat bit.

1. Kode Program

/\* pemakaian beberapa operator terhadap RELATIONAL DAN bit \*/

public class Oper2 {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* KAMUS \*/

char i, j;

/\* ALGORITMA \*/

i = 3; /\* 00000011 dalam biner \*/

j = 4; /\* 00000100 dalam biner \*/

System.out.println("i = "+ (int) i);

System.out.println("j = "+ j);

System.out.println("i & j = "+ (i & j));

/\* 0: 00000000 dalam biner \*/

System.out.println("i | j = "+ (i | j));

/\* 7: 00000111 biner \*/

System.out.println("i ^ j = "+ (i ^ j));

/\* 7: 00000111 biner Ingat!!! operator "^" pada bahasa

java bukan sebagai pangkat\*/

System.out.println(Math.pow(i, j));

/\* Class Math memiliki method pow(a,b) untuk

pemangkatan\*/

System.out.println("~i = "+ ~i);

/\* -4: 11111100 biner \*/

Kode program di atas merupakan contoh pemakaian beberapa operator terhadap operator relasional dan operator bitwise dalam bahasa pemrograman Java. Berikut adalah penjelasan mengenai kode program tersebut:

Pendefinisian Variabel:

char i, j;: Mendefinisikan dua variabel i dan j dengan tipe data karakter.

Pemberian Nilai:

i = 3;: Menginisialisasi variabel i dengan nilai 3. Dalam representasi biner, nilai ini adalah 00000011.

j = 4;: Menginisialisasi variabel j dengan nilai 4. Dalam representasi biner, nilai ini adalah 00000100.

Operasi-operasi Bit:

System.out.println("i & j = "+ (i & j));: Melakukan operasi bitwise AND antara nilai i dan j. Hasilnya adalah 00000000 dalam biner atau 0 dalam desimal.

System.out.println("i | j = "+ (i | j));: Melakukan operasi bitwise OR antara nilai i dan j. Hasilnya adalah 00000111 dalam biner atau 7 dalam desimal.

System.out.println("i ^ j = "+ (i ^ j));: Melakukan operasi bitwise XOR antara nilai i dan j. Hasilnya adalah 00000111 dalam biner atau 7 dalam desimal.

System.out.println("~i = "+ ~i);: Melakukan operasi bitwise NOT pada nilai i. Hasilnya adalah -4 dalam desimal atau 11111100 dalam biner.

Pemangkatan:

System.out.println(Math.pow(i, j));: Melakukan pemangkatan nilai i pangkat j menggunakan method Math.pow(). Namun, perlu dicatat bahwa hasil ini tidak dicetak karena operasi pemangkatan ini akan menyebabkan error, karena parameter yang diterima adalah bilangan floating-point, sementara i dan j bertipe char.

Output:

Program akan mencetak hasil dari operasi-operasi bitwise dan relasional yang dilakukan pada layar.

Dengan demikian, kode program tersebut adalah contoh penggunaan operator bitwise dan relasional dalam bahasa pemrograman Java untuk melakukan operasi pada level bit.

Kode Program 14

public class Oper3 {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* Algoritma \*/

if (true && true){ System.out.println(true && true);} /\*

true = true and true \*/ if (true & true) { System.out.println(true & false); }

/\* true & true \*/ if (true){ System.out.println(true); }/\* true \*/

if (true || true){ System.out.println(true);} /\* true = true or true \*/

if (true|false) { System.out.println(true|false); } /\* true|false \*/

PENJELASAN KODE :

if (true && true) { System.out.println(true && true); }: Pada baris ini, terdapat pernyataan if dengan kondisi true && true. Kondisi tersebut menggunakan operator logika AND (&&) yang akan bernilai true jika kedua kondisi di dalamnya bernilai true. Oleh karena itu, kondisi ini akan terpenuhi dan program akan mencetak nilai true.

if (true & true) { System.out.println(true & false); }: Pada baris ini, juga terdapat pernyataan if dengan kondisi true & true. Kondisi ini menggunakan operator bitwise AND (&), yang akan menghasilkan true jika kedua operandnya (bit) berada di posisi yang sama-sama menyala (1). Oleh karena itu, hasilnya akan menjadi true, dan program akan mencetak nilai true.

if (true) { System.out.println(true); }: Pernyataan ini memiliki kondisi true, yang berarti selalu terpenuhi. Oleh karena itu, program akan selalu mencetak nilai true.

if (true || true) { System.out.println(true); }: Pernyataan ini memiliki kondisi true || true, yang menggunakan operator logika OR (||). Kondisi ini akan terpenuhi jika salah satu atau kedua kondisi di dalamnya bernilai true. Oleh karena itu, program akan mencetak nilai true.

if (true|false) { System.out.println(true|false); }: Pada baris ini, juga terdapat pernyataan if dengan kondisi true|false. Kondisi ini menggunakan operator bitwise OR (|), yang akan menghasilkan true jika salah satu atau kedua operandnya (bit) berada di posisi yang salah satu atau keduanya menyala (1). Oleh karena itu, hasilnya akan menjadi true, dan program akan mencetak nilai true.

KODE PROGRAM 15

/\* Operator terner \*/

public class Oper4 {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* KAMUS \*/

int i = 0; /\* perhatikan int i,j=0 bukan seperti ini \*/

int j = 0;

char c = 8; char d = 10;

int e = (((int)c > (int)d) ? c: d);

int k = ((i>j) ? i: j);

/\* ALGORITMA \*/

System.out.print ("Nilai e = "+ e);

System.out.print ("\nNilai k = "+ k);

i = 2;

j = 3;

k = ((i++>j++) ? i: j) ;

System.out.print ("\nNilai k = "+ k);

PENJELASAN KODE :   
  
Variabel yang dideklarasikan:

i dan j diinisialisasi dengan nilai 0.

c dan d adalah variabel bertipe char yang diinisialisasi dengan nilai 8 dan 10.

e diinisialisasi dengan hasil operasi ternary (((int)c > (int)d) ? c: d). Ini berarti jika nilai ASCII dari c lebih besar dari nilai ASCII dari d, maka e akan diisi dengan nilai c, jika tidak, maka akan diisi dengan nilai d.

k diinisialisasi dengan hasil operasi ternary ((i>j) ? i: j). Jadi, k akan diisi dengan nilai i jika i lebih besar dari j, dan diisi dengan nilai j jika tidak.

Keluaran pertama:

Mencetak nilai e dan k.

Nilai e akan dicetak terlebih dahulu, yang akan bergantung pada perbandingan nilai ASCII antara c dan d.

Nilai k akan dicetak setelahnya, yang akan bergantung pada perbandingan nilai i dan j, di mana keduanya saat ini memiliki nilai 0.

Setelah itu, nilai i diubah menjadi 2 dan nilai j diubah menjadi 3.

Perubahan nilai i dan j akan mempengaruhi nilai k dalam operasi ternary selanjutnya.

k diinisialisasi kembali dengan hasil operasi ternary baru ((i++>j++) ? i: j). Operasi ini akan membandingkan nilai i dan j setelah operasi penambahan yang ditunjukkan oleh i++ dan j++. Jika i awalnya lebih besar dari j, maka nilai i akan diincrement, jika tidak, nilai j yang akan diincrement. Hasil operasi ternary ini kemudian akan diassign ke k.

Keluaran kedua:

Mencetak kembali nilai k setelah operasi ternary kedua. Nilai k sekarang akan berbeda dari sebelumnya karena perubahan nilai i dan j.

KODE PROGRAM 16

/\* Contoh pengoperasian variabel bertype dasar \*/

public class Oprator {

public static void main(String[] args) {

// TODO Auto-generated method stub

/\* Kamus \*

/ boolean Bool1, Bool2, TF ;

int i,j, hsl ;

float x,y,res;

/\* algoritma \*/

System.out.println("Silahkan baca teksnya dan tambahkan perintah program di bawah ini untuk menampilkan output program");

Bool1 = true; Bool2 = false;

TF = Bool1 && Bool2 ; /\* Boolean AND \*/

TF = Bool1 || Bool2 ; /\* Boolean OR \*/

TF = ! Bool1 ; /\* NOT \*/

TF = Bool1 ^Bool2; /\* XOR \*/

/\* operasi numerik \*/

i = 5; j = 2 ;

hsl = i+j;

hsl = i - j;

hsl = i / j;

hsl = i \* j;

hsl = i /j ; /\* pembagian bulat \*/

hsl = i%j ; /\* sisa modulo \*/

/\* operasi numerik \*/ x = 5 ; y = 5 ;

res = x + y;

res = x - y;

res = x / y;

res = x \* y;

/\* operasi relasional numerik \*/

TF = (i==j);

TF = (i!=j);

TF = (i < j);

TF = (i > j);

TF = (i <= j);

TF = (i >= j);

/\* operasi relasional numerik \*/

TF = (x != y);

TF = (x < y);

TF = (x > y);

TF = (x <= y);

TF = (x >= y);

PENJELASAN KODE :

Deklarasi Variabel:

Variabel Bool1, Bool2, dan TF dideklarasikan dengan tipe data boolean.

Variabel i, j, dan hsl dideklarasikan dengan tipe data int.

Variabel x, y, dan res dideklarasikan dengan tipe data float.

Operasi Boolean:

Dilakukan beberapa operasi logika boolean:

TF = Bool1 && Bool2; (Boolean AND)

TF = Bool1 || Bool2; (Boolean OR)

TF = !Bool1; (NOT)

TF = Bool1 ^ Bool2; (XOR)

Operasi Numerik:

Operasi-operasi aritmatika pada tipe data int dan float dilakukan:

Penambahan, pengurangan, perkalian, dan pembagian antara i dan j.

Operasi tersebut juga termasuk pembagian bulat (i / j) dan sisa modulo (i % j).

Penambahan, pengurangan, perkalian, dan pembagian antara x dan y.

Operasi Relasional Numerik:

Dilakukan beberapa operasi relasional antara i dan j, serta antara x dan y:

TF = (i == j); (sama dengan)

TF = (i != j); (tidak sama dengan)

TF = (i < j); (kurang dari)

TF = (i > j); (lebih dari)

TF = (i <= j); (kurang dari atau sama dengan)

TF = (i >= j); (lebih dari atau sama dengan)

Operasi yang sama juga dilakukan antara x dan y.