**JAVA: ERREPIKAPENAK**

## ERREPIKAPEN-EGITURAK

Orain arte ikusi dugunez, gure programek baldintzak har ditzakete kontuan exekutatzen direnean, baina oraindik ez dakigu nola egin instrukzio multzo bat errepikatzea. Hori egiteko, errepikapenegiturak daude. Egitura hauek instrukzio multzo bat errepikatzen dute baldintza bat kontuan hartuta.

Egitura hauei errepikapen-egitura, begizta edo iterazio-egitura esaten zaie. Javan, 4 begizta mota daude:

***for*** begizta ***for/in*** begizta (aurrerago ikusiko da). ***while*** begizta

***do/while*** begizta

*for* eta *for/in* kontagailu bidez kontrolatutako begiztak dira. *while* eta *do/while*, ordea, gertaera bidez kontrolatutakoak.

Zein begizta erabili erabakitzeko, galdera hauei erantzungo diegu:

* Ba al dakigu, aldez aurretik, instrukzio multzoa zenbat aldiz errepikatu behar dugun?
* Errepikapen-egitura hasieran exekutatzea, berez egiten da, ala baldintzatuta dago?
* Noiz arte errepikatu behar dugu instrukzio multzoa?
* Baldintza bat betetzen bada, egin behar ote dugu errepikapena?

Begizta bakoitza banan-banan aztertuko dugu:

#### FOR EGITURA

*for* kontagailu bidez kontrolatutako begizta da. Begizta honek ezaugarri hauek dauzka:

* Badakigu zenbat bider errepikatuko den.
* Kontagailu-aldagai bat erabiltzen du zenbat iterazio egiten diren kontrolatzeko.

Begizta mota honetan, 3 eragiketa egin ohi dira:

* Kontagailu-aldagaia hasieratzen da.
* Kontagailu-aldagaia ebaluatzen da. Beraren balioa eta ezarritako iterazio kopurua konparatzen dira.
* Kontagailu-aldagaiaren balioa eguneratzen da, eta handitu edo txikitu egiten da iterazio bakoitzean.



***for*** errepikapen egitura erabilita. Erreparatu, kontu handiz, kodean sartuta dauden komentarioei.

public class Errepikapenak\_for {

/\* for egitura erabiliko dugu pantailan zazpiaren taula erakusteko

\*/

public static void main(String[] args) { // Aldagaiak deklaratu eta hasieratu

int zenbakia = 7; int kontagailua; int emaitza = 0;

//Informazioa ateratzea

System.out.println (zenbakia+" zenbakiaren biderkatze-taula");

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

//for begizta erabiliko dugu

for (kontagailua = 1; kontagailua< = 10;kontagailua++){

/\* Begiztaren buruan, aldagai-kontagailua hasieratzen da. Baldintza da errepikapena 1etik 10era egitea, eta aldagaia banan-banan handitzen da iterazio bakoitzean (kontagailua + +).

\*/

emaitza = kontagailua \* zenbakia;

System.out.println(zenbakia + " x " + kontagailua + " = " + emaitza); /\* + eragilea erabiliz testuak eta aldagaien balioak kateatzen ditugu.

\*/

}

}

}

#### WHILE EGITURA

*while* egitura gertaeren bidez kontrolatutako begizta da, ez kontagailuaren bidez. *while* erabiliko dugu ez dakigunean kode-bloke bat zenbat aldiz errepikatu behar den, baina jakin dakigunean errepikatu egin behar dugula baldintza bat betetzen den bitartean.

Lehenik eta behin*, while* begizta kontrolatzen duen baldintza ebaluatzen da, eta egia bada, begiztaren gorputza exekutatuko egingo da eta baldintza faltsu bihurtu arte jarraituko du exekutatzen.

Baldintza faltsua bada ebaluatzen den lehen aldian, begiztaren gorputza ez da exekutatuko. Derrigorrezkoa da baldintza aldatuko duen sententziaren bat egotea *while* begiztaren gorputzaren barruan. Hala ez balitz, begizta infinitu bat izango genuke.



***while*** errepikapenegitura erabilita. Erreparatu, kontu handiz, kodean sartuta dauden komentarioei.

public class Errepikapenak\_while {

/\* while egitura erabiliko dugu pantailan zazpiaren taula erakusteko

\*/

public static void main(String[] args) {

// Aldagaiak deklaratu eta hasieratu

int zenbakia = 7;

int kontagailua;

int emaitza = 0;

//Informazioa ateratzea

System.out.println (zenbakia+" zenbakiaren biderkatze-taula");

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

//while begizta erabiliko dugu kontagailua = 0;

while (kontagailua<10){

/\* Begiztaren baldintza ezarri dugu (kontagailua < 10)

\*/

emaitza = kontagailua \* zenbakia;

System.out.println(zenbakia + " x " + kontagailua + " = " + emaitza); kontagailua ++;

/\*Kontagailuaren balioa aldatzen dugu, begizta infiniturik ez sortzeko.\*/

}

}

}

## DO/WHILE EGITURA

*while* egitura bezala, hau ere gertaeren bidez kontrolatutako egitura da. Kasu honetan, begiztaren gorputzean dagoen kodea gutxienez behin exekutatuko da, eta errepikatzen jarraituko du baldintza bat betetzen den bitartean. *while* eta *do/while* egituren arteko desberdintasunik handiena hau da : *while* egituran, hasierako baldintza betetzen ez bada, begiztaren gorputza ez da sekula exekutatuko, eta *do/while* egituran, aldiz, begiztaren gorputzeko kodea gutxienez behin exekutatuko da.

Derrigorrezkoa da baldintza aldatuko duen sententziaren bat egotea *do/while* begiztaren gorputzaren barruan. Hala ez balitz, begizta infinitu bat izango genuke.



***do/while*** errepikapen egitura erabilita. Erreparatu, kontu handiz, kodean sartuta dauden komentarioei.

public class Errepikapenak\_do\_while {

/\* do/while egitura erabiliko dugu pantailan zazpiaren taula erakusteko

\*/

public static void main(String[] args) { // Aldagaiak deklaratu eta hasieratu

int zenbakia = 7; int kontagailua; int emaitza = 0;

//Informazioa ateratzea

System.out.println (zenbakia+" zenbakiaren biderkatze-taula");

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

//do/while begizta erabiliko dugu kontagailua = 0;

do{

emaitza = kontagailua \* zenbakia;

System.out.println(zenbakia + " x " + kontagailua + " = " + emaitza); kontagailua ++;

/\*Kontagailuaren balioa aldatzen dugu, begizta infiniturik ez sortzeko.\*/

}while (kontagailua<10);

/\* Begiztaren baldintza ezarri dugu (kontagailua < 10) \*/

}

}