**JAVA: ALDAGAIAK**

###### PROGRAMA BATEN EGITURA OROKORRA

public class KaixoMundua {

/\* KaixoMundua programa \*/

public static void main(String[] args) {

/\*programa honek gauza bakarra egiten du: “KaixoMundua” katea pantailan atera\*/

System.out.println("Kaixo, mundua!");

}

}

###### IDENTIFIKADOREAK ETA ALDAGAIAK

Programa batek datuak erabiltzen ditu hainbat lan egiteko: kalkuluak egiteko, datuok inprimagailu edo pantaila bidez aurkeztu edo erabiltzaileari eskatzeko, diskan gordetzeko, etab.

Datu horiek, erabili ahal izateko moduan egon daitezen, programak aldagaietan gordetzen ditu. **Aldagaia**: Ordenagailuaren memoriaren zati batean gordetzen den datua, behar denean programan erabili ahal izateko.

Aldagai bat 3 propietatek definitzen dute:

1. Izenak
2. Datu motak
3. Onartzen dituen balioen heinak

Aldagaiari ematen zaion izenari **identifikadore** esaten zaio.

***IDENTIFIKADOREAK***

Javan, **identifikadorea** programatzaileak programa baten aldagai, konstante, klase, metodo edo paketeei ematen dien izena da. Identifikadoreak honako ezaugarri hauek bete behar ditu:

* Lehen karakterea letra bat, azpimarra karakterea, \_, edo dolar karakterea, $, izan behar da.
* Hurrengo karaktereak letrak edo zenbakiak izan behar dira.
* Ezin da programazio-hizkuntzaren hitz erreserbaturik erabili.

Javak maiuskulen eta minuskulen artean bereizten du, eta hori kontuan hartu behar da identifikadoreak sortu eta erabiltzean.

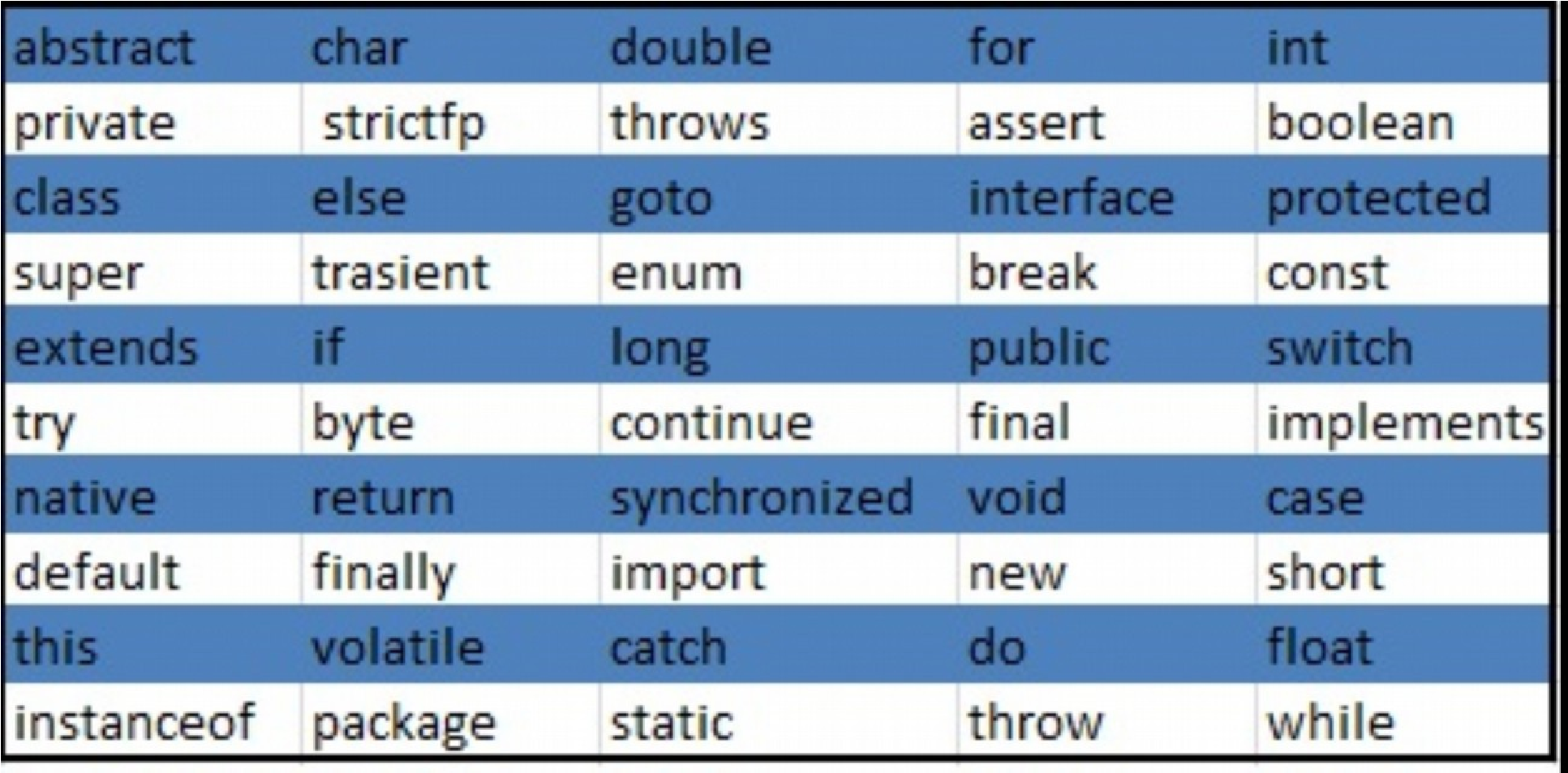
Nahiz eta araurik ez egon, emango zaien erabileraren arabera, egon badago mundu osoan onartzen den hitzarmen bat identifikadoreak izendatzeko:

| **Identifikadorea** | **Hitzarmena** | **Adibidea** |
| --- | --- | --- |
| Aldagaiaren izena | Minuskulaz hasten dira. Hitz batez baino gehiagoz osatuta badago, elkarren ondoan eta maiuskulaz hasita doaz gainerako hitzak. | batuketa, ikasleKopurua |
| Konstantearen izena | Dena maiuskulatan. Hitzak behe-marraz banatzen dira | PI, TAMAINA\_MAX |
| Klasearen izena | Maiuskulaz hasten dira | Pertsona, Karratua |
| Metodoaren izena | Minuskulaz hasten dira | aldatu\_Balioa, abiadura\_Eskuratu |

***HITZ ERRESERBATUAK***

Programazio-hizkuntzan erabiltzeko erreserbatzen dira eta, beraz, ezin dira identifikadore modura erabili.

Javan, hitz hauek daude erreserbatuta:



***ALDAGAI MOTAK.***

Programa batean, hainbat aldagai mota aurki ditzakegu: gordetzen duten datu motaren araberakoak, beren balioa programan zehar aldatu edo berdin mantentzearen araberakoak… Javan, aldagai mota hauek bereiz daitezke:

1. Oinarrizko datu motak gordetzen dituzten aldagaiak eta erreferentzia-aldagaiak: Aldagai bi hauek gordetzen dituzten datuen moten arabera sailkatzen dira.

1. Aldagarriak eta konstanteak:

Aldagarria. Datuak gordetzeko erabiltzen da. Datuaren balioa hainbat alditan alda daiteke programa exekutatzen den bitartean.

Konstanteak edo aldagai finalak. Izenak berak dioenez, hauen balioa ez da aldatzen, jakina, programa exekutatzen den bitartean.

1. Aldagai klasekideak eta aldagai lokalak, programan sortzen diren tokiaren arabera.

Aldagai klasekideak: Klase jakin baten barruan sortzen diren aldagaiak dira, baina metodoetatik kanpo.

Aldagai lokalak: Metodo edo beste kode-bloke baten barruan sortu eta erabiltzen direnak. Metodo edo kode-blokea amaitzean, aldagaia desagertu egiten da.

**ADIBIDEA**

public class AldagaiAdibidea{

final double PI = 3.1415927; // PI konstantea da

int x; // x AldagaiAdibidea klasearen

//aldagai klasekidea da

int eskuratuX (int x){ // x parametro bat da

int balioZaharra = this.x; // balioZaharra aldagai

//lokala da

return balioZaharra;

}

}

Aldagai hauek dituen klase bat sortu dugu:

**PI: Aldagai finala edo konstantea**. Balio finkoa du; beraz, ez da haren balioa aldatuko programa osoan zehar.

**x: Aldagai klasekidea.** AldagaiAdibidea izeneko klasearen barruan dago eta gordetzen dituen datuak **int** motakoak dira. Hainbat balio har ditzake programan zehar, normalean klase berean sortutako beste metodo baten bidez.

**balioZaharra: Aldagai lokala.** Aldagai hau lokala da, eskuratuX metodoaren barruan sortuta dagoelako. Metodo horretatik kanpo, aldagaia ez da existitzen.

###### OINARRIZKO DATU MOTAK I

Zenbakiak, karaktereak edo balio logikoak (boolearrak) gordetzeko erabiltzen dira. **Javan, datu mota hauek ez dira objektuak.**

| **Datu mota** | **Adierazitako informazioa** | **Heina** | **Azalpena** |
| --- | --- | --- | --- |
| byte | Zenbaki osoak | - 128 ------ +127 | 8 bit erabiltzen dira datua gordetzeko |
| short | Zenbaki osoak | - 32768 ------ +32767 | 16 bit erabiltzen dira datua gordetzeko |
| int | Zenbaki osoak | - 2147483648 ------  +2147483647 | 32 bit erabiltzen dira datua gordetzeko |
| long | Zenbaki osoak | - 9223372036854775808  ------ +  9223372036854775807 | 64 bit erabiltzen dira datua gordetzeko |
| char | UNICODE karaktereak | 0 ------ 65535 | Zenbakiak UNICODE bidez adierazteko erabiltzen da. |
| float | Koma mugikorreko datuak  (32 bit) | 7 digituko zehaztasuna gutxi gorabehera. | Bit 1 zeinurako, 11 bit berretzailerako eta 24 mantisarako. |
| double | Koma mugikorreko datuak  (64 bit) | 16 digituko zehaztasuna gutxi gorabehera. | Bit 1 zeinurako, 11 bit berretzailerako eta 52 mantisarako. |
| boolean | Balio logikoak | true/false | Adierazpen boolear bat egia (true) edo faltsua (false) den adierazteko erabiltzen da. |

**ADIBIDEA**

public class Adibidea { public static void main(String[] args) { //aldagaiak deklaratu byte hil = 12;

int kontagailua = 0;

double pi = 3.1415926535897932384626433832795;

float interesa = 4.25e2F; char letra = 'Z';

boolean aurkituta = true;

//balioak inprimatu

System.out.println(hil); //12 inprimatuko du

System.out.println(kontagailua); //0 inprimatuko du

System.out.println(pi); //3.141592653589793 inprimatuko du

System.out.println(interesa); //425.0 inprimatuko du

System.out.println(letra); //Z inprimatuko du

System.out.println(aurkituta); //true inprimatuko du

}

}

Kontuan izan behar dugu Java konpilatzaileak **char** datu mota zenbakizkotzat hartzen duela. Horren azalpena erraza da: ez du gordetzen karaktere bat, zenbaki bat baizik, karaktere bakoitzari UNICODE kodean dagokion zenbakia hain zuzen.

Bestalde, oso garrantzitsua da egoki aukeratzea zein datu mota erabiliko dugun. Horretarako, kontuan hartu beharko dugu zer informazio (zenbakiak, testua...) gorde nahi dugun, eta hartu ditzakeen balioen heina ere bai. Arrazoi horrengatik, batzuetan, agian, zenbaki errealak gordetzeko erabiltzen den datu mota erabili beharko dugu, nahiz eta zenbaki osoekin lan egin. Esate baterako, **int** mota erabilita ezin da lurraren populazioa adierazi, har dezakeen baliorik altuena 2.147.483.647 baita (32 bit birako osagarrian). Lurraren populazioa 7.000 milioi biztanlekoa da, beraz, **long** motako datua erabili beharko dugu, gutxienez.

###### OINARRIZKO DATU MOTAK II

Datu mota errealak zenbaki errealak adierazteko erabiltzen dira. Javan, 2 daude: **float** (32 bit) eta **double** (64 bit). Bit gehiago erabilita, bi helburu hauek lortzen dira:

* Zenbaki handiagoak adieraztea
* Zati hamartarra zehatzagoa izatea

*Java-programatzaile gehienek* ***double*** *datu mota erabiltzen dute. Horrela eginda, gutxiagotu egiten dira zenbaki hamartarrak biribiltzean sortzen diren erroreak.*

###### ALDAGAI ETA KONSTANTEAK DEKLARATZEA ETA HASIERATZEA

Dagoeneko, badakigu zer diren aldagaiak eta konstanteak. Baina hori ez da nahikoa, eta, pare bat galdera etortzen zaizkigu burura: Nola sortzen dira aldagaiak programa batean? Zer egin behar da programa batean aldagaiak erabili ahal izateko?

Programa batean aldagaiak erabili ahal izateko, aldagaiak berak sortu behar dira lehendabizi: izen bat eman eta zer informazio mota gordeko duten adierazi; hau da, **aldagaia deklaratu**.

Hona hemen adibide bat:

**int ikasleKop = 15;**

**double radio = 3.14;**

**guztira = 102.95;**

Aurreko adibidean, 3 aldagai deklaratu ditugu:

*ikasleKop* **int** motakoa, eta *radio* eta *guztira* **double** motakoak. Ikusten denez, bi aldagai datu mota berekoak direnean, aski da behin bakarrik adierazita eta bi aldagaiak koma baten bidez banatuta.

Gainera, nahiz eta derrigorrezkoa ez izan, aurreko aldagaiak deklarazioan bertan hasieratu ditugu: 15, 3.14 eta 102.95 hasierako balioak emanda, hurrenez hurren.

Aldagaia, aldaketarik jasango ez badu programa osoan zehar, **konstante** bezala deklaratuko dugu. Konstanteak ***final*** hitz erreserbatua erabilita deklaratzen dira:

**final double PI = 3.1415926536;**

Zer gertatzen da aldagai bat hasieratzen ez badugu? Aldagai motaren arabera, bi gauza gerta litezke:

• Aldagai-klasekideak: automatikoki hasieratzen dira baliorik ematen ez badiegu:

* + zenbakizkoak 0 batekin.
  + *char* motakoak **null** moduan.
  + boolearrak **false** moduan.
  + Erreferentziak **null** moduan.

•Aldagai lokalak ez dira automatikoki hasieratzen. Hurrengo adibidean, **p** izeneko aldagaia deklaratzen da, baina hasieratu gabe. Gero, **q** izeneko aldagaia deklaratzen da eta **p** aldagaiaren balioarekin hasieratzen da. Zer gertatzen da orduan? p **aldagaiak baliorik ez daukanez, programa konpilatzean, konpilatzaileak errore bat erakutsiko digu.**

**int p; int q = p; // error**

**DATU-MOTEN ARTEKO BIHURKETA (CAST)**

Pentsatu bi zenbakiren arteko zatiketa egin nahi dugula. Izango ote du zenbaki hamartarrik zatiketa horren emaitzak? Baietz pentsatuko dugu zatikizuna zatitzailearen multiploa baldin ez bada. Eta mundu errealean hala da; baina informatikoan ez dago zertan horrela izan. Javan, zatikizuna eta zatitzailea zenbaki osoak badira, zatidura ere zenbaki osoa izango da; hau da, zenbaki hamartarrik gabea, besterik esan ezean.

Emaitzak zenbaki hamartarrak izan ditzan, datu motaren bihurketa bat egin beharra dago; hau da, emaitza guk nahi dugun motakoa izan dadin, datu moten bihurketa egiten da. Bi bihurketa mota daude:

* 1. **Berehalako bihurketak:** Berehalako bihurketa deritzo datu mota bateko aldagai bati bit gutxiagoko beste datu mota bateko balio bat esleitzea. Adibidez, *double* motako aldagai bati *float* motako balio bat esleitzen zaionean, balio hori *double* datu motara bihurtzen da programatzaileak ezer egin gabe.

* 1. **Bihurketa esplizituak:** Programatzaileak espresuki egiten du bihurketa **Cast eragilea** erabilita. Cast eragilea batarra da. Bihurtu nahi den datuaren aurrean kokatzen da, datu mota parentesi artean jarrita. Cast eragilearen lehentasuna eragile batarrena da, eta ezkerretik eskuinera elkartzen da.

Adibideak: Okerra

**int a; byte b;**

**a = 12; b = a; // errorea, ez dago baimenduta b-ko datu mota a-ko datu mota baino txikiagoa delako**:

Zuzena

**int a; byte b = (byte) a;**

**IRUZKINAK**

Iruzkinak oso garrantzitsuak dira programa batek zer egiten duen deskribatzeko, unitatean zehar erabili izan ditugun adibideetan idatzi dugun kodea hobeto ulertzeko. Normalean, programak iruzkin-lerro batzuekin hasten dira. Lerro horietan, programak zer egiten duen, nork idatzi duen eta azken aldaketaren data idatzi ohi dira.

Javan, honako iruzkin mota hauek aurki ditzakegu:

Lerro bateko iruzkinak: // mugatzailea erabiltzen da

// lerro bateko iruzkina

Hainbat lerrotako iruzkinak: /\* paragrafoaren hasieran eta \*/ paragrafoaren amaieran.

/\* hau hainbat lerrotako iruzkin bat da \*/

Javadoc iruzkinak: /\*\* eta \*/ mugatzaileak erabiltzen ditu. Dokumentazio automatikoa sortzeko erabiltzen dira, javadoc programaren bidez.