I.Atala: Java Sarrera

Java lengoaia

Helburuak:

Nola adierazten dira Klaseak eta Objektuak Java-n

Java
- Identifikatzaileak
- Hitz erreserbatuak
- Datu motak

OO
- Klase baten erazagupena
- Aldagai baten erazagupena
- Metodo baten erazagupena
- Objektu baten eraikuntza

Identifikatzaileak

- Aldagaiak, metodoak, klaseak eta objektuak izendatzeko balio dute.
- Hizki bat, azpimarra edo $ ikurraz hasten dira
- **Letra larri eta xeheen artean desberdintzen da**
- Ez dago luzera maximorik

identifikatzaileak ezin dira hitz erreserbatuak izan

Identifikatzaileak

- **Hitzarmena**:

– Aldagai, metodo eta objektuen izenak letra xehez hasten dira
– Klaseen izenak letra larriekin hasten dira
– Hitz bat baino gehiago baldin badu, 'eraHonetara' idatziko dugu, hitz bakoitzaren lehenengo letra larriaz idatziz eta azpimarra deuseztatuz (CamelCase)

Hitz erreserbatuak

Erreserbatuak

abstract	double	int	static
boolean	else	interface	super
break	extends	long	switch
byte	final	native	synchronized
case	finally	new	this
catch	float	null	throw
char	for	package	throws
class	goto	private	transient *
const *	if	protected	try
continue	implements	public	void
default	import	return	volatile
do	instanceOf	short	while

Erreserbatuak baina erabiltzen ez direnak

cast	future	generic	inner
operator	outer	rest	var

Datu motak

- Javan dauden aldagai/atributu guztiak datu mota batenak dira

- Atributu/aldagaien datu motak zehazten du:
  - Har dezaketen **balioak**
  - Egin daitezkeen **eragiketak**
- Ikusiko ditugu
  - ***Oinarrizko datu motak***
  - ***Erreferentziazko datu motak***

Oinarrizko datu-motak

Primitive type	Size	Minimum	Maximum	Wrapper type
boolean	1-bit	–	–	**Boolean**
char	16-bit	Unicode 0	Unicode 2^{16} - 1	**Character**
byte	8-bit	-128	+127	**Byte**[1]
short	16-bit	-2^{15}	$+2^{15} - 1$	**Short**[1]
int	32-bit	-2^{31}	$+2^{31} - 1$	**Integer**
long	64-bit	-2^{63}	$+2^{63} - 1$	**Long**
float	32-bit	IEEE754	IEEE754	**Float**
double	64-bit	IEEE754	IEEE754	**Double**
void		–	–	**Void**

OHARRA: Tamaina aurrez zehaztuta eta plataformarekiko independentea

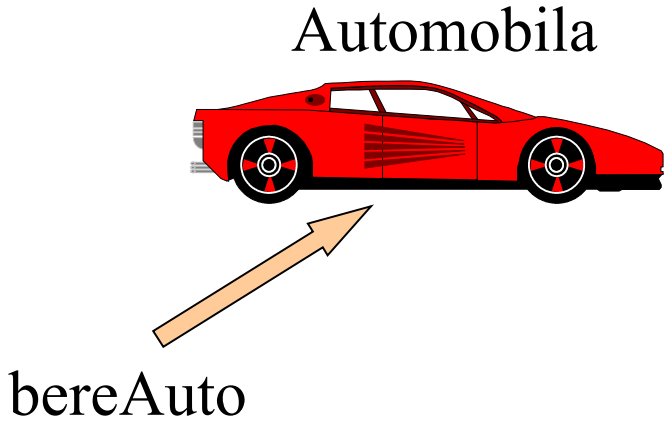
java.lang

Errefentziazko datu motak

Aldagaiaren balioa objektu baten (erakusle) erreferentzia bat da

Adibide:

```
public class Pertsona{
 Automobila bereAuto;
 Pertsona bereLaguna;
}
```



Helburuak:

Nola adierazten dira Klaseak eta Objektuak Java-n

Java
- Identifikatzaileak
- Hitz erreserbatuak
- Datu motak

OO
- Klase baten erazagupena
- Aldagai baten erazagupena
- Metodo baten erazagupena
- Objektu baten sorketa

Klase baten erazagupena

```
public class Auto{
  // Atributuen erazagupena
  // (kolore, matrikula, abiadura, etab.)
  // Metodoen erazagupena
  // (martxan jarri, abiatu, gelditu, etab.)
}
```

Sintaxia

(deskribatzaileak) class KlaseIzena{
//klase
}

Estiloaren aldetik...

- Fitxategi izena = klase izena
- Lehenengoa, letra larriaz
- Hitzak lehenengo letra larriaz batuko dira NireLehenengoKlasea
- Koska (indent)

Atributuen erazagupena

```
public class Auto{
  // Atributuen erazagupena
  private String kolore;
  private int abiadura;
  // Metodoen deklarazioa
  // (martxan jarri, abiatu, gelditu, etab.)
}
```

Sintaxia

(deskr) mota izena;
(deskr) mota izena1, izena2;
(deskr) mota izena = balioa;

Aldagaiaren hasieraketa balio batekin

Estiloa

- Izen intuitiboak
- Lehenengo letra xeheaz
- Zuriune eta gidoirik gabe
- Hitzak lehenengo hizki larriaz batuko dira (nireAldagaia)

Atributuen erazagupena
(besterik ezeko balioak)

Oinarrizko datu moten **besterik ezeko** balioak.

bool	**false**
char	**\u0000**
byte	**0**
short	**0**
int	**0**
long	**0**
float	**0**
double	**0**

String	**null**

Atributuen erazagupena
Nork atzitu dezake atributua?

private String kolore;
private int abiadura

	Atzipen baimena
public	Klase guztientzat eskuragarri
private	Bere klaseen metodoengatik soilik eskuragarri
(friendly)	Pakete berberan dauden klaseengatik eskuragarri
protected	Pakete berberan dauden klaseengatik eskuragarri (eta azpiklaseengatik, dauden paketean daudela)

Atributuen erazagupena

Aldatzen al da atributuen balioa exekuzio garaian ?

- ***Konstanteak*** (Aldaezinak diren atributuak):
  - ***static final*** hitz erreserbatuak erabili
  - **Derrigorrezkoa da deklarazioan hasieratzea**

```
public class Auto{
   private static final  int    GURPILZENB = 4;
   private String kolore;
}
```

- **Hitzarmena**:

Konstanteen identifikatzaileak letra LARRIAZ idatziko dira (eta _ karakterea erabili daiteke hitzak banatzeko)

Metodoen erazagupena

```
public class Auto{
  //Atributuen deklarazioa
  private String kolore;
  private int abiadura;

  //Metodoen deklarazioa
  public void martxanJarri(){
     //martxanJarri metodoaren kodea
  }
  public void aurreratu(int abiadura){
     // aurreratu metodoaren kodea
  }
  public String getKolore(){
      //kotxearen kolorea lortzeko kodea
     return kolore;
  }
```

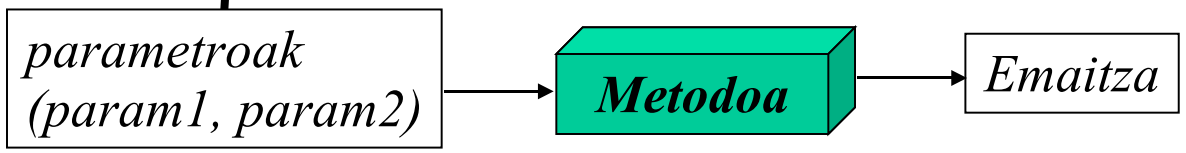
Estiloa

- Izen intuitiboak
- Lehenengo letra xehea
- Zuriune eta gidoirik gabe
- Hitzak lehenengo letra larriaz batuko dira nireMetodoa

Metodoen erazagupena

```
public class Auto{
//...
public void aurreratu (int abiadura) {
        //aurreratu metodoaren kodea
}
//...
}
```

Auto.java

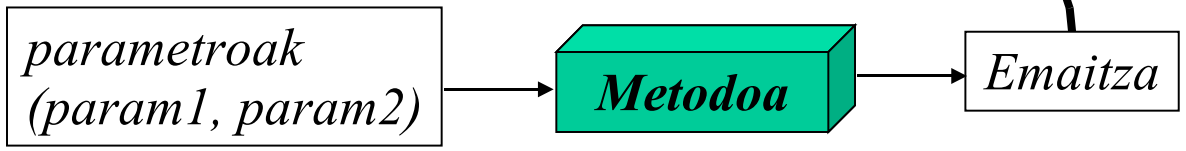


(deskribatzaileak) EmaitzaMota MetodoIzena(mota1 param1, mota2 param2){
//metodoaren kodea
return espresioa; //Emaitza void denean ez da jartzen
}

Metodoen deklarazioa

```
public class Kotxe{
//...
 public String zerKolore(){
   //kolorea lortzeko kodea
   return kolore;
 }
//...
}
```

Auto.java



(deskribatzaileak) *EmaitzaMota MetodoIzena(mota1 param1, mota2 param2){*
//metodoaren kodea
return espresioa; //Emaitza mota void denean ez da jartzen
}

Metodoen erazagupena

- Objektuen portaera deskribatzen duten kode-lerroak dira.

Metodoek:

– 0, 1, 2 ... n **argumentu** dauzkate (parametroak)
– Deklarazioan **emaitzaren datu mota** definitzen da (eraikitzaileak salbu)
– Aldagai lokalak egon daitezke. Aldagai hauek ez dira besterik ezeko balioekin hasieratzen

Metodoak

- Metodoak emaitza bat itzultzen badu, **return** agindua erabiliz egingo du

- Metodo bereziak
  - Eraikitzailea eta ***main*** metodoa (metodo nagusia)

Metodo eraikitzaileak

- Objektu bat sortzen denean, bere atributuak **hasieratzen** dira eraikitzaile metodoaren bitartez.

- Metodo Eraikitzaileek:
  - Klasearen **izen berdina** daukate
  - Ez dute ezer itzultzen (ez void ez eta return ere ez)

Metodo eraikitzaileak

- Komenigarria da gutxienez eraikitzaile bat izatea.

Horren ezean, sistemak berak besterik ezeko eraikitzailea sortuko du, existitzen diren atributuak besterik ezeko balioekin hasieratuz

- Klaseak eraikitzaile bat badauka, besterik ezekoa deuseztatu egingo da

Metodo eraikitzaileak

Instantzia sorketa

```
public class Zirkunferentzia {
  private  float erradioa;
  public Zirkunferentzia(float e) { // eraikitzailea
            erradioa=e;
   }
}
```

Zirkunferentzia klaseko objektu bat sortzen

```
Zirkunferentzia z;
z=new Zirkunferentzia(3.4);
```

Metodo nagusia (main)

```
public class AdibideKlasea {
    public static void main(String args[ ])
```

• Interpretatzaileak aplikazioa exekutatzeko bilatzen duen **lehen** metodoa da.

• main funtzioaren parametroak ***(String args[ ]),*** array baten bidez jasotzen dira, eta bere balioak komando-lerrotik idazten diren balioak dira

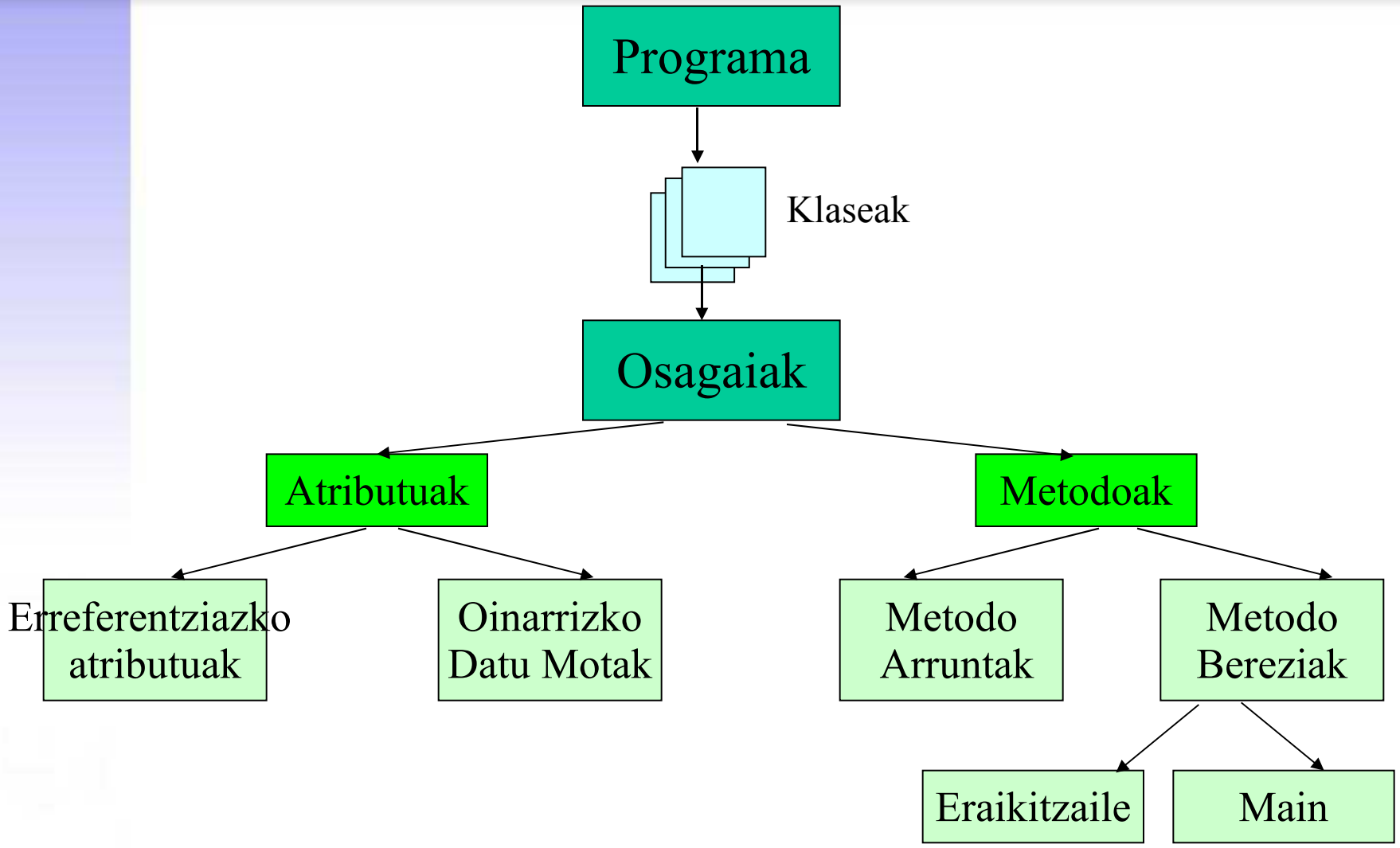
java AdibideKlasea arg1 arg2 ...

Metodo nagusia (main)

```
public static void main (String args[]) { }
```

- ***public*** edozein lekutik deitu daiteke
- ***static*** klasearen metodo bat dela adierazten du, hau da, ez dago main metodo bat instantzia bakoitzeko
- ***void*** ez du baliorik itzultzen

Laburpena


Programa
Klaseak
Osagaiak
Atributuak
Metodoak
Erreferentziazko atributuak
Oinarrizko Datu Motak
Metodo Arruntak
Metodo Bereziak
Eraikitzaile
Main

Exekuzio kontrola

baldintzazkoak

if(*Boolean-expression***)**
statement

if(*Boolean-expression***)**
statement
else
statement

Iterazioak/Begiztak

while(*Boolean-expression***)**
statement

do
statement
while(*Boolean-expression***);**

for(*initialization***;** *Boolean-expression***;** *step***)**
statement

Adibidea

```
public class Ikaslea
{
  private String nan
  private String izena;
  private int kurtsoa;

  public Ikaslea (String pNan)
  {
    nan=pNan;
  }
}
```

```
public void setIzena(String pIzena)
{
   izena=pIzena;
}

public void setKurtsoa(int pKurtsoa)
{
   kurtsoa=pKurtsoa;
 }

 public void print() {
   System.out.println(nan);
   System.out.println(izena);
   System.out.println(kurtsoa);
  }
```

Adibidea

```
class Nagusia {
   public static void main(String[] args) {
    Ikaslea i1=new Ikaslea(“44.123.456V”);
    Ikaslea i2=new Ikaslea(“00.000.000A”)
    Ikaslea i3=i2;
    i3.setIzena(“Mikel Elexpuru”);
    i2.print();
  }
}
```

Ariketa

- Hurrengo metodoen sinadurak dituen Osoko izeneko klasea definitu :
  - eraikitzailea
  - isLehena(); // ez ahaztu sarrera
  - isPositibo() // eta itzulera
  - inprimatu(); // parametroak
  - balioaAldatu(int berria); // ...