

Datu egiturak eta algoritmoak

0. GAIA: Java lengoaia



Aurkibidea

1. Sarrera

- Java aplikazio motak
- Edizioa, konpilazioa eta exekuzioa
- Aplikazio baten egitura

2. Java: objektuei zuzendutako lengoaia

- Objektuak eta klaseak
- Kapsulaketa
- Elementu pribatuak eta publikoak

3. Javaren ezaugarriak

- Datu motak
- Klaseak: atributuak eta metodoak



Aurkibidea

1. Sarrera

- Java aplikazio motak
- Edizioa, konpilazioa eta exekuzioa
- Aplikazio baten egitura

2. Java: objektuei zuzendutako lengoaia

- Objektuak eta klaseak
- Kapsulaketa
- Elementu pribatuak eta publikoak

3. Javaren ezaugarriak

- Datu motak
- Klaseak: atributuak eta metodoak



Zer da JAVA?

Programazio-lengoaia bat da

Objektuetan oinarritua

Sarean jarduteko egokitua

Bezera-aplikazioak egiteko baliagarria

Zerbitzari-aplikazioak egiteko baliagarria



JAVA aplikazio motak

4 aplikazio mota

Bezeroan:

Aplikazio arruntak

Appletak

Zerbitzarian

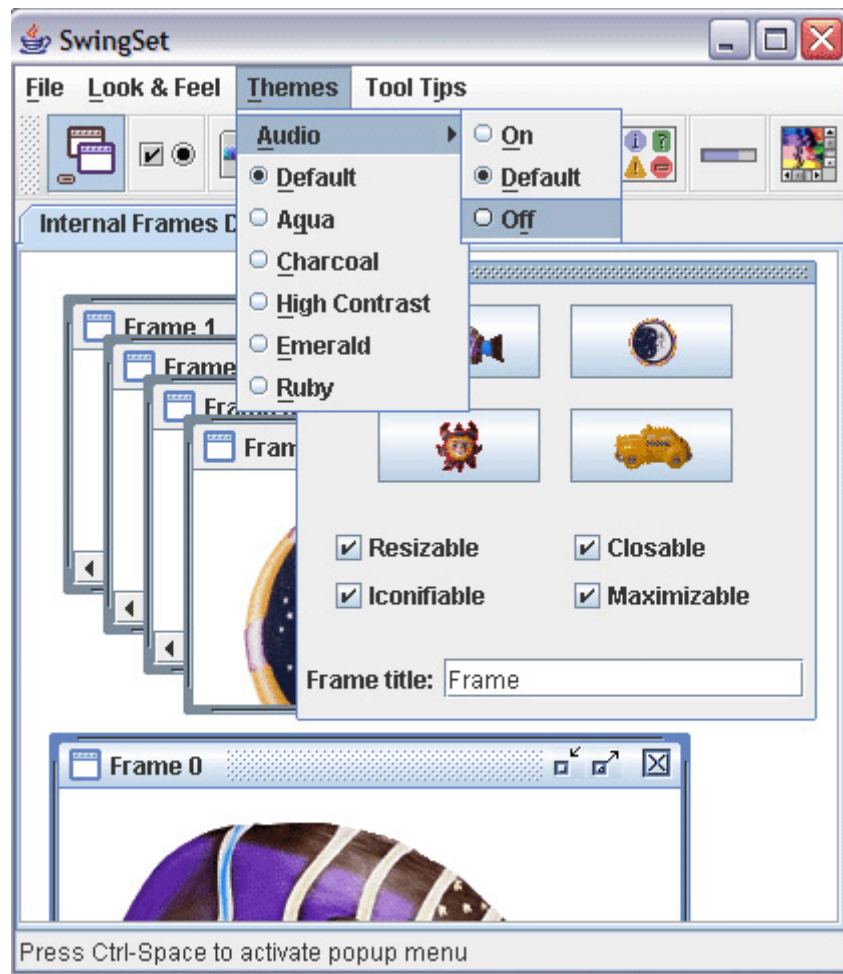
Servletak

JSP-ak



JAVA aplikazio motak

Bezeroan: Aplikazio arruntak

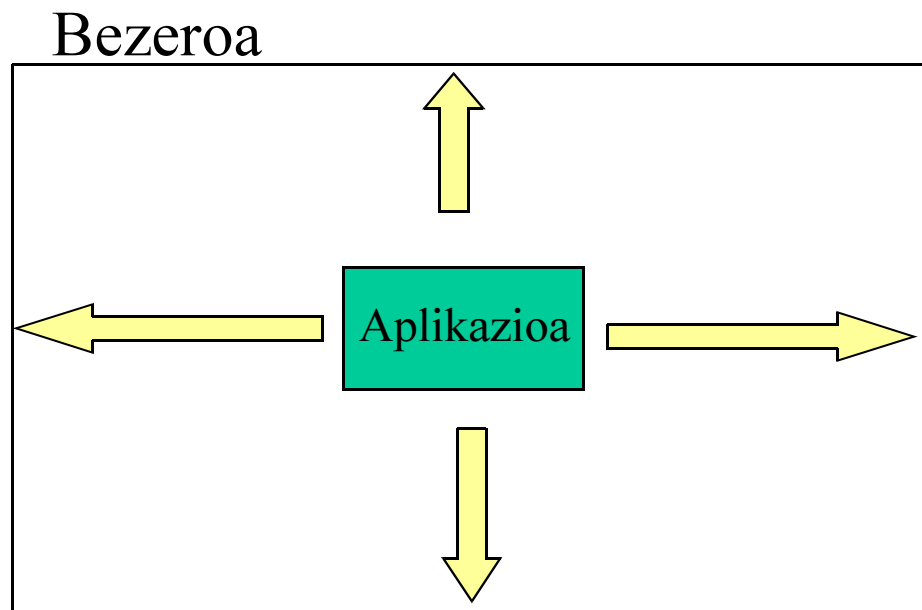




JAVA aplikazio motak

Bezeroan: Aplikazio arruntak

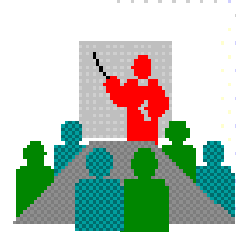
Murrizketarik gabe exekutatzeko diren Java **aplikazio lokalak dira**.





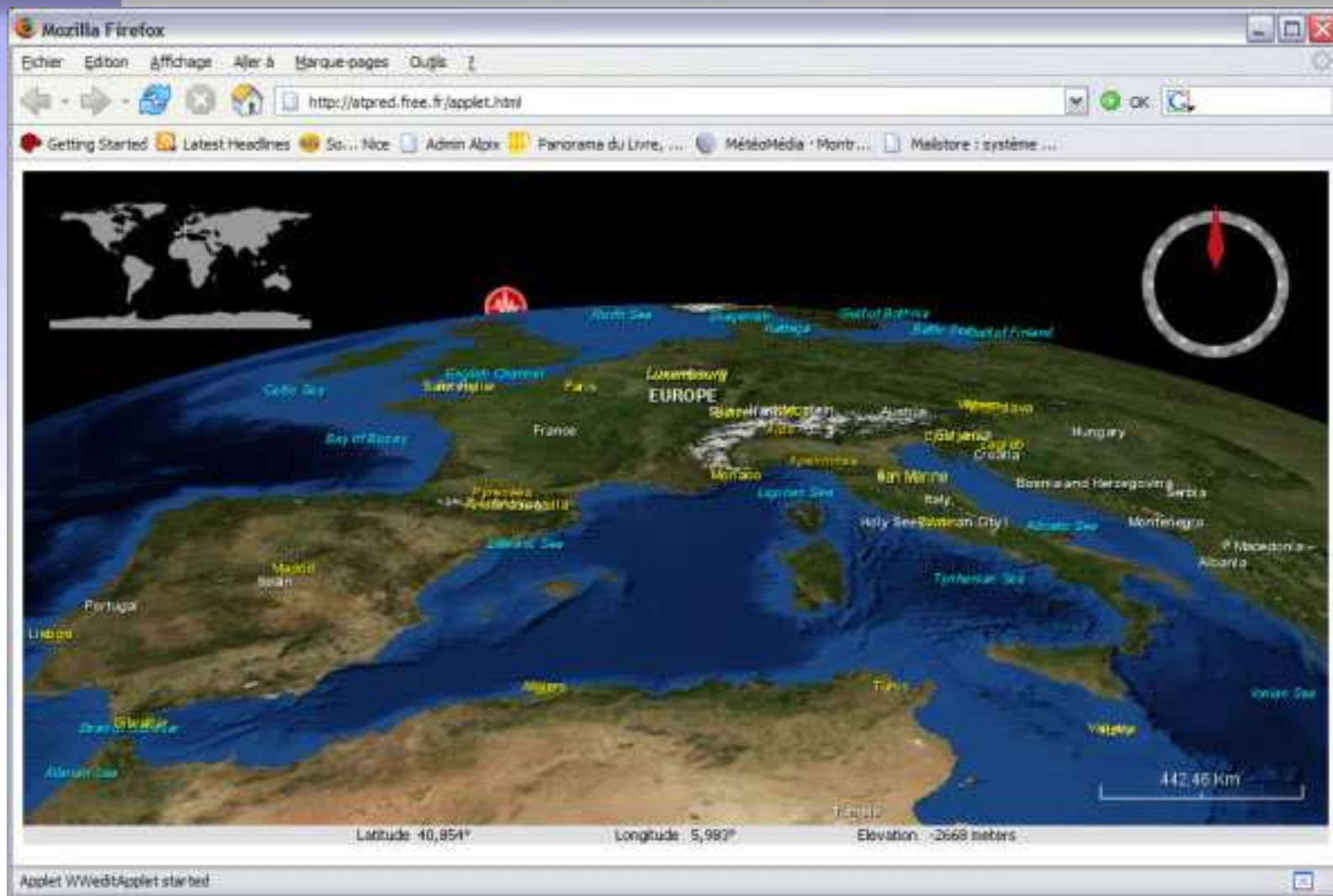
Aplikazio lokalaren kodea

```
public class AplikazioLokala {  
    public static void main(String args[]) {  
        System.out.println("Hellooo World");  
    }  
}
```

JAVA aplikazio motak

Bezeroan: Applet-ak



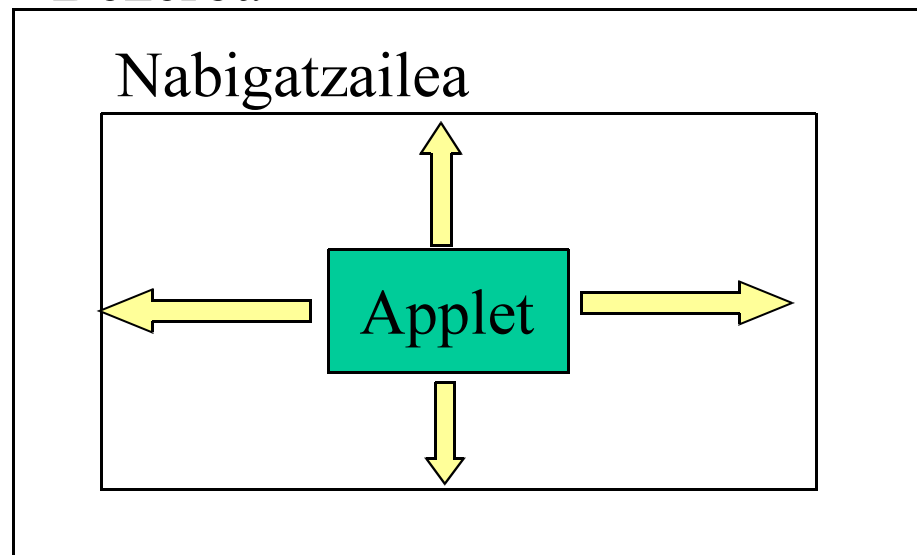


JAVA aplikazio motak

Bezeroan: Applet-ak

Nabigatzailearen testuinguruan exekutatzeko diren Java aplikazioak d

Bezeroa





Applet baten kodea

```
import java.applet.*;
import java.awt.*;

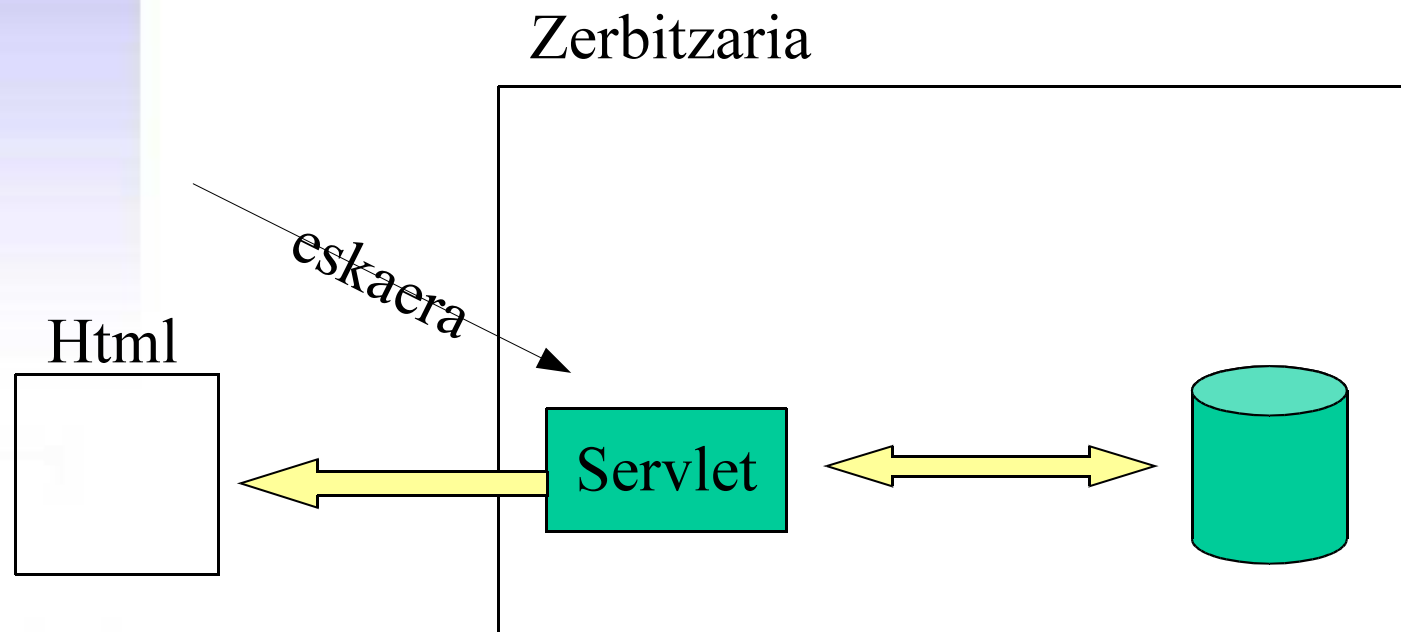
public class HelloWorld extends Applet
{
    public void paint(Graphics g)
    {
        //method to draw text on screen
        // String first, then x and y coordinate.
        g.drawString("Hey hey hey",20,20);
        g.drawString("Hellooo World",20,40);
    }
}
```



JAVA aplikazio motak

Zerbitzarian: Servlet-ak

HTML kodea itzultzen duten Java aplikazioak dira
Orri dinamikoak sortzeko erabiltzen dira





Servlet adibidea

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class HelloWorld extends HttpServlet {

    public void doGet(HttpServletRequest request,
                      HttpServletResponse response) {
        PrintWriter out = response.getWriter();
        out.println("Hello World"); }
}
```

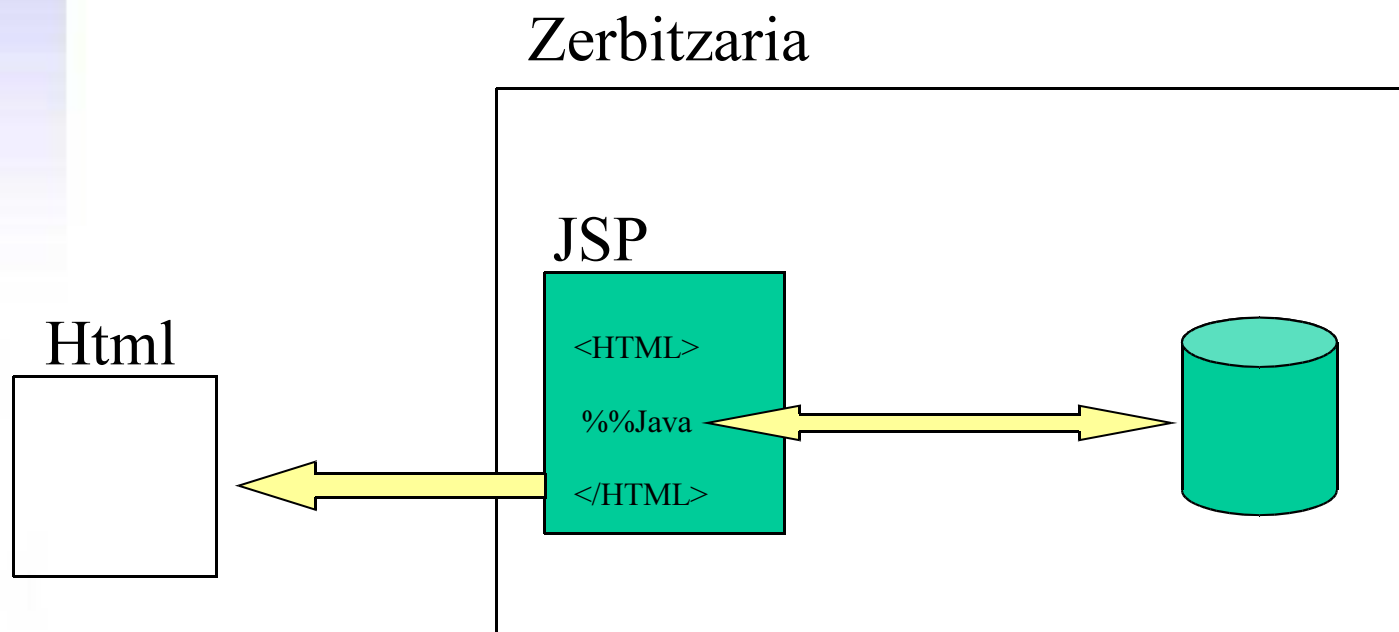


JAVA aplikazio motak

Zerbitzarian: JSP

Java kodea txertatuta duten HTML orriak (*embedded*).

Lehendabiziko deian, sistemak JSP kodea Servlet batean bihurtzen du.





JSP Adibidea

```
<HTML>
  <HEAD>
  </HEAD>
  <BODY>
    Data: <%= new java.util.Date() %>
    <%= if (Math.random() < 0.5) { %>
      Buenas <B>tardes</B> !
    <%= } else { %>
      <B>Arratsalde on</B>!
    <%= } %>
  </BODY>
</HTML>
```

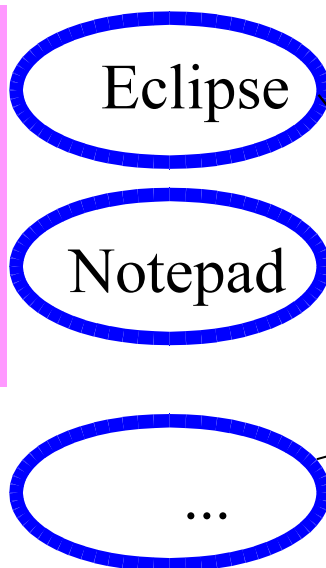


Garapen-arkitektura

Garapen Tresnak

- Eclipse
- NetBeans
- JDeveloper
- JBuilder

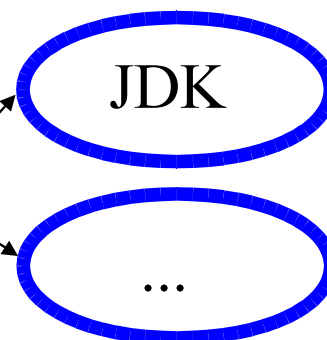
Editoreak



Java
Kodea

fitxategia.java

Konpiladoreak



Bytecode

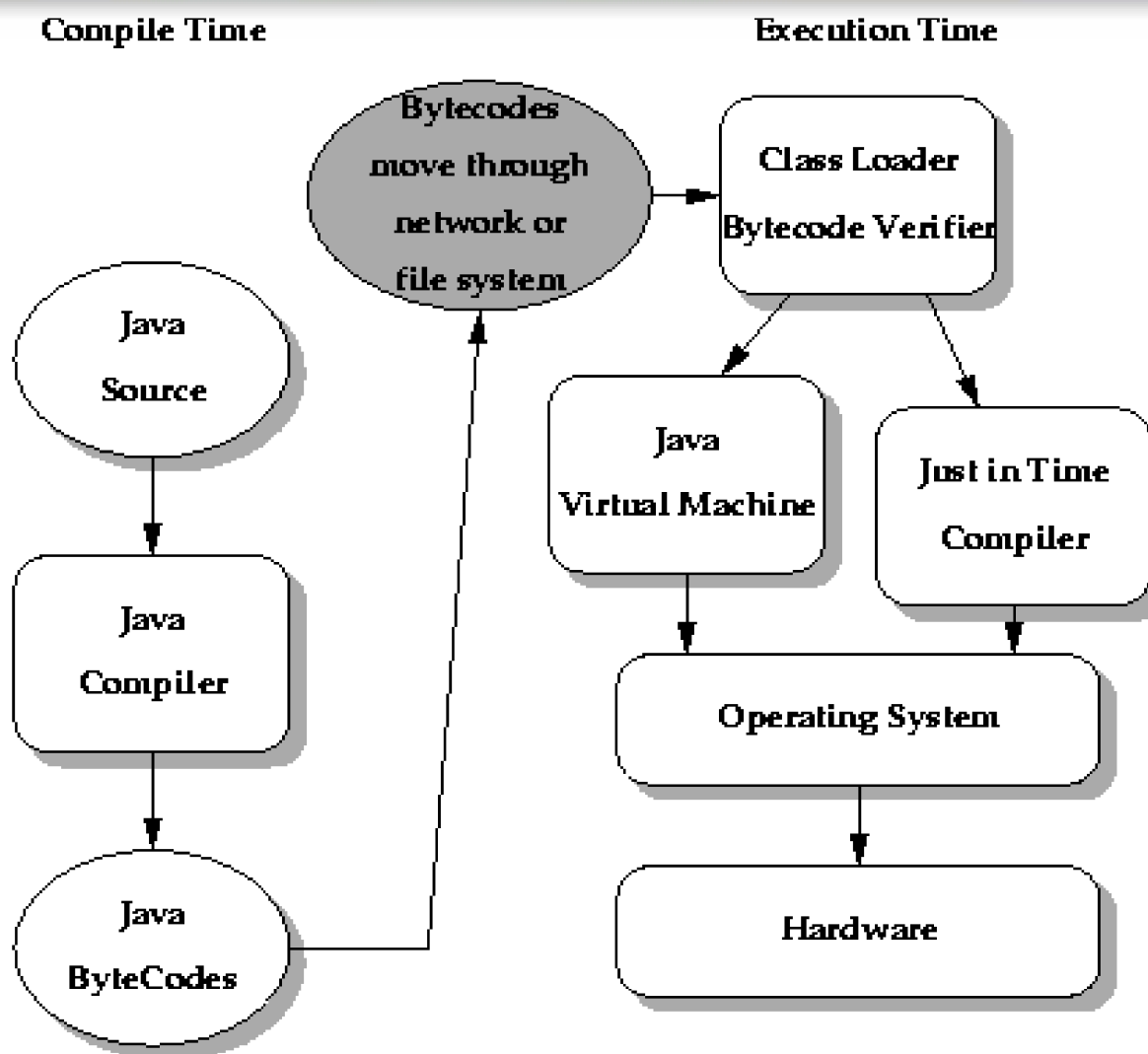
fitxategia.class

```
javac fitxategia.java
```

IDE: garapenerako ingurune integratua (GII)



Exekuzio-arkitektura





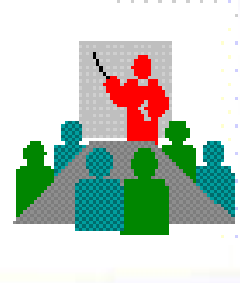
Exekuzio-arkitektura

Klase kargatzailea

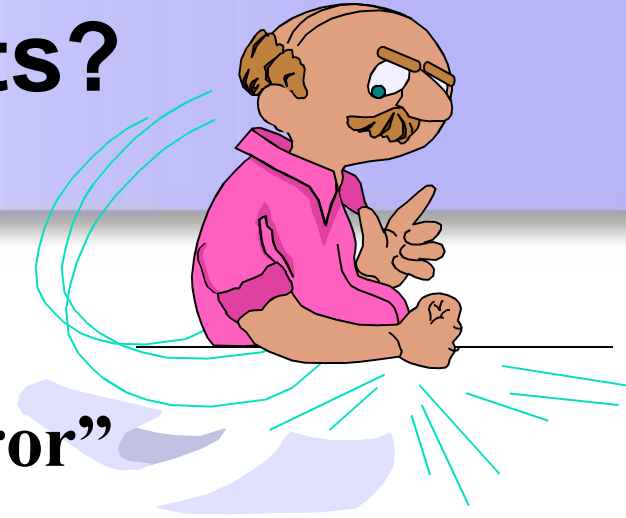
- Aplikazioaren *bytekodeak* egiaztatzen ditu
- *Bytekodeak* behar dituen klaseak kargatzen ditu

JVM (Java Virtual Machine)

- *Bytekodea* (sasikodea) exekutatzen du
- Web-nabigatzaile edo sistema eragilearen mende
- Bi aukera:
 - 1- Kodea interpretatzea (Java Interpreter)
 - 2- Era dinamikoan konpilatzea (JIT)



Zerk egin dezake huts?



Konpilatu ➔ “Syntax Error”

Kargatu ➔ “Class not found Exception”

Exekutatu ➔ “Null Pointer Exception”

```
Autoa a; // a aldagaia erazagutu  
a.martxanJarri(); // a hasieratu baino lehen...
```

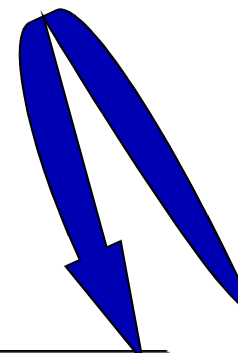


Lehenengo aplikazioa

HelloWorld.java

```
// Adibide aplikazioa  
// "Hello world"  
//  
public class HelloWorld{  
    public static void main(String args[]){  
        System.out.println("Hello world");  
    }  
}
```

HelloWorld.class



1. Konpilazioa:

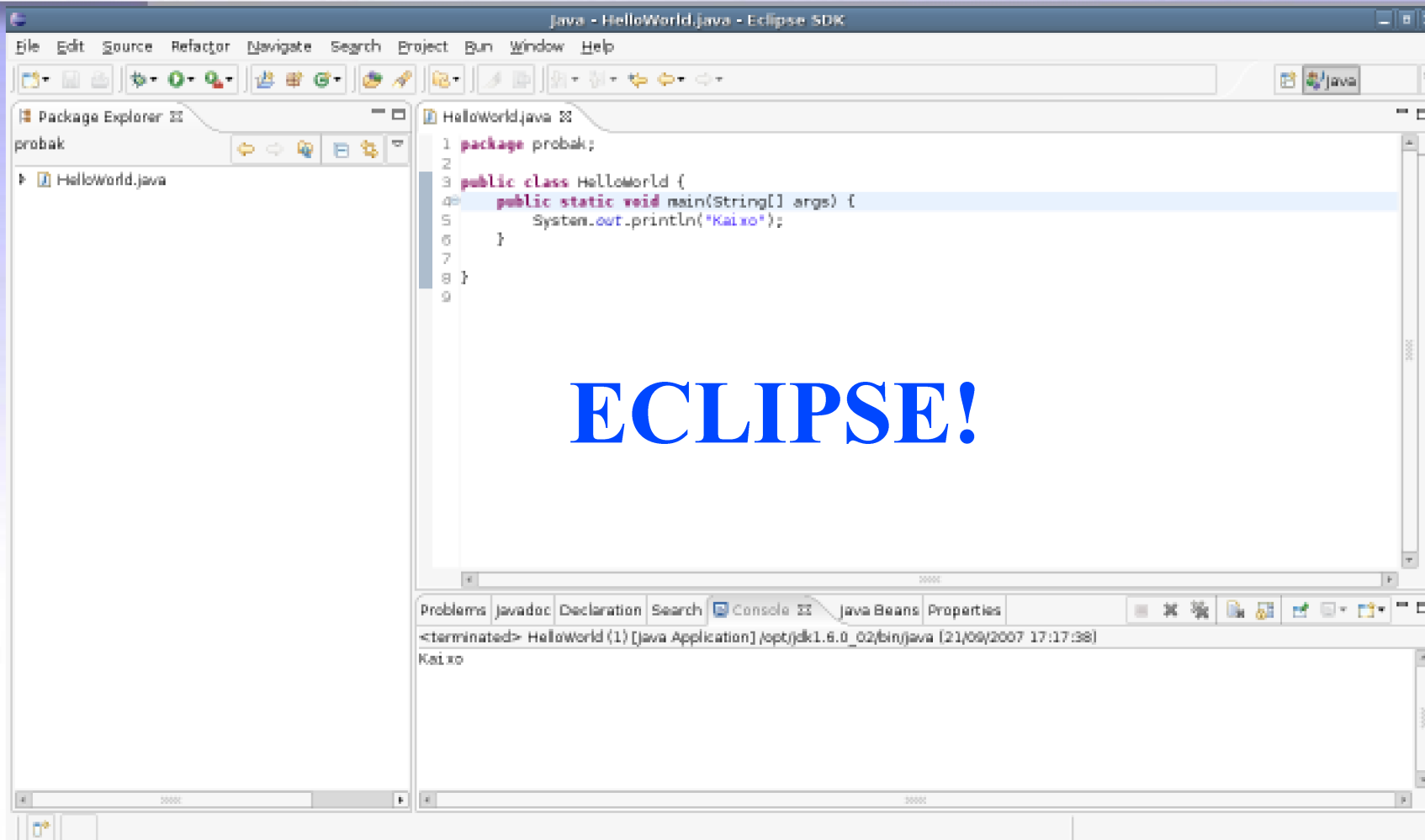
javac HelloWorld.java

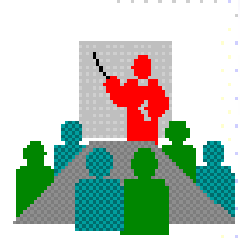
2. Exekuzioa:

java HelloWorld



Lehenengo aplikazioa





Aurkibidea

1. Sarrera

- Java aplikazio motak
- Edizioa, konpilazioa eta exekuzioa
- Aplikazio baten egitura

2. Java: objektuei zuzendutako lengoaia

- Objektuak eta klaseak
- Kapsulaketa
- Elementu pribatuak eta publikoak

3. Javaren ezaugarriak

- Datu motak
- Klaseak: atributuak eta metodoak



Helburuak

- Objektuei zuzendutako programazioaren oinarrizko kontzeptuak finkatu:
 - **klaseak eta objektuak**
 - osagaiak (atributuak, metodoak)
- Klase eta objektuen arteko erlazioa aztertu



Helburuak

instantziak



- Objektu soil bat sortu eta modelatu:
 - bere **egoera** (aldagaien edo atributuen bidez)
 - bere **portaera** (metodoen bidez)



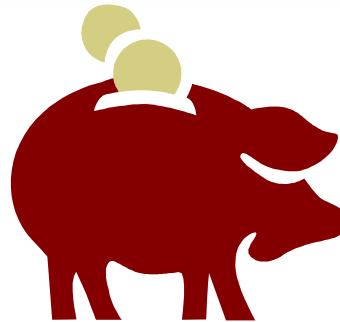
Zer da objektu bat ?



- Objektu bat edozein gauzaren (oinarrizkoa/konplexua) (erreala/irudizkoa) errepresentazioa da
- Gauza guztiak ez dira objektu bezala kontuan hartu behar, beren **ezaugarri** gisa baizik: kolorea, izena, abiadura...



Zer da objektu bat ?



Bankuko kontu bat

- *Datu-abstrakzioa*

+Kontuaren ezaugarriak:

- jabea
- kopurua
- dibisa
- mugimenduak

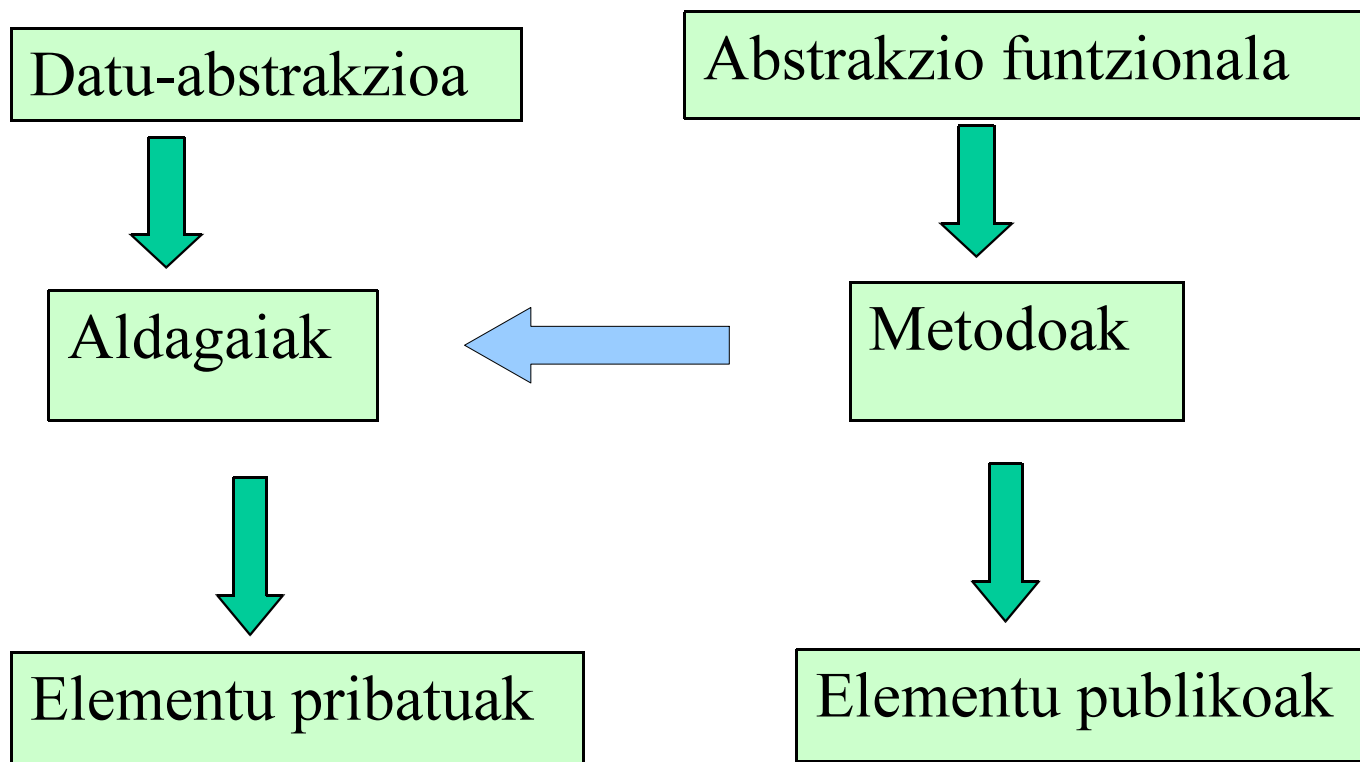
- *Abstrakzio funtzionala*

+Kontuaren portaera:

- atera/sartu
- galdetu
- ireki/itxi

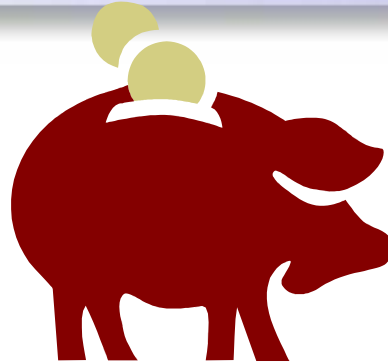


Zer da objektu bat ?





Objektu adibide bat



Datu-abstrakzioa
(ezaugarriak = atributuak)

```
String jabea="Patxi Irigoien"  
float kopurua=0  
String dibisa="euro"  
...
```

Abstrakzio funtzionala
(portaera = metodoak)

```
sartu(float kop)  
atera(float kop)  
galdetuMuga()
```

Metodoak BETI objektuetara bideratuta daude!



Objektuen definizioa

Elementu pribatuak

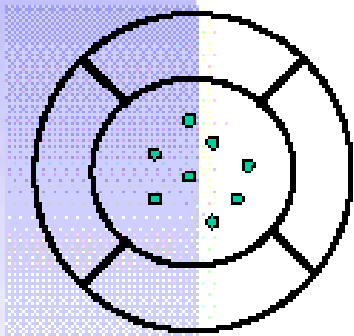
- Objektuen egoera (status) deskribatzen dute
- Objektuek egiten dutena nola egiten duten ere deskribatzen da alde pribatuan.

Elementu publikoak

- Objektuek zer egin dezaketen deskribatzen dute, hau da, objektuen portaera zein den.



Objektuen kapsulazioa



OBJEKTU BATEN OSAGIAK

Objektuek atributuak kapsulatzen dituzte, beren atzipena metodoen bidez soilik baimenduz

Atributuak: Balio-kokalekua

Metodoak: Funtzio-kokalekua

getters, setters!



Objektuen kapsulazioa

Objektu batek dauka:

Egoera, beren aldagai edo atributuen balioen bidez adierazita

Portaera, beren metodoen bidez definitua



Eskuarki:

Metodoak publikoak dira

Aldagaiak pribatuak

Metodo pribatuak egon daitezke

Aldagai publikoak edukitzea arriskutsua da



Adibidea

Kubo objektua



Atributuak: *goikoAldea:int*

Metodoak:

jaurti()

zenbakiaLortu()



Objektuen sailkapena

Autoa



Klasea: atributu eta portaera berdineko objektu multzoa.

“Auto” klaseari buruz hitz egin daiteke.

“Nire autoa” objektu bat da, hau da, “Auto” klasearen **instantzia** bat.



Objektuak vs. Klaseak

Klase bat entitate abstraktu bat da

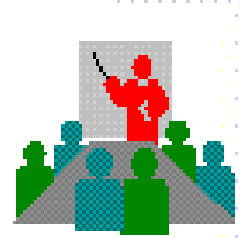
- Datu sailkapen mota bat da
- Egitura eta portaera antzekoa duten objektuen metodoak eta atributuak klaseetan definitzen dira

Kontu klasea

Atributuak: jabea, kantitatea, dibisa

Metodoak: atera, sartu, itxi, ireki, ...

- Klasearen izena
- Atributuak(datuak)
- Metodoak (eragiketak)

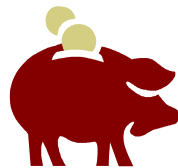


Objektuak vs. Klaseak

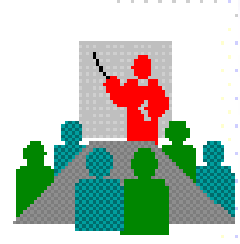
Objektu bat klase baten instantzia edo aldagaia da

- Objektu bat bere klaseko beste objektuengandik desberdintzen da objektu bakoitzak atributuetan bere balio propioak (besteengandik desberdinak izan daitezkeenak) dituelako.

**Patxiren
kontua**
(Kontu
klaseko
objektua)



Metodoak: *sartu()*, *atera()*, *galdetuMug()*
Atributuak: *jabea*="Patxi Irigoien", *kopurua*=0
dibisa=euro



Aurkibidea

1. Sarrera

- Java aplikazio motak
- Edizioa, konpilazioa eta exekuzioa
- Aplikazio baten egitura

2. Java: objektuei zuzendutako lengoaia

- Objektuak eta klaseak
- Kapsulaketa
- Elementu pribatuak eta publikoak

3. Javaren ezaugarriak

- Datu motak
- Klaseak: atributuak eta metodoak



Helburuak:

Nola adierazten dira klaseak eta objektuak Javan

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

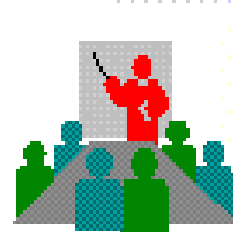
identifikatzaileak ezin dira hitz erreserbatuak izan



Identifikatzaileak

- **Hitzarmena:**

- Aldagai, metodo eta objektuen izenak letra xehez hasten dira
- Klaseen izenak letra larriz hasten dira
- Hitz bat baino gehiago baldin badu, 'eraHonetara' idatziko dugu, hitz bakoitzaren lehenengo letra larriz eta azpimarrarik gabe (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 aldagai/atributu guztiei datu mota zehatz bat esleitu behar zaie
- Atributu/aldagaien datu motak zehazten du:
 - Har ditzaketen **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 ¹

OHARRA: Tamaina aurrez zehaztuta eta
plataformarekiko independentea

java.lang



Erreferentziako datu motak

Aldagaiaren balioa objektu baten (erakusle) erreferentzia bat da

Adibidez:

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



Helburuak:

Nola adierazten dira klaseak eta objektuak Javan

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

(modifikatzaileak) class

```
KlaseIzena{
```

```
//klase
```

```
}
```

Estiloaren aldetik...

- Fitxategi izena = klase izena
- Lehenengoa, letra larriz
- Hitzak letra larria erabiliz batuko dira. Adib.: NireLehenengoKlasea
- Tabulazioa (indent) erabili!



Atributuen erazagupena

```
public class Auto{  
    // Atributuen erazagupena  
    private String kolore;  
    private int abiadura=100;  
    // Metodoen deklarazioa  
    // (martxan jarri, abiatu, gelditu, etab.)  
}
```

Sintaxia

```
(descr) mota izena;  
(descr) mota izena1, izena2;  
(descr) 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.

boolean	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;
int abiadura;

	<i>Atzipen baimena</i>
public	Klase guztiek dute eskuragarri
private	Bere klaseen metodoek soilik eskuragarri
(friendly edo <i>package</i>)	Pakete bereko klaseengandik



Atributuen erazagupena

Aldatzen al da atributuen balioa exekuzio garaian ?

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

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

- **Hitzarmena:**

Konstanteen identifikatzaileak letra LARRIZ 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

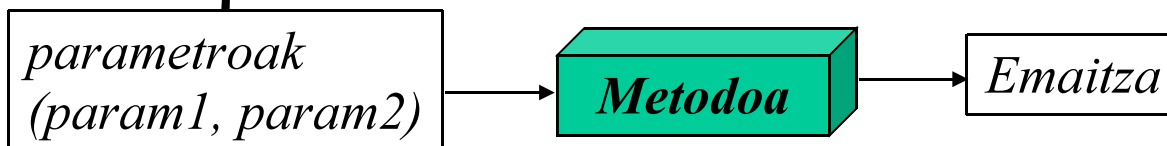
- Izen intuitiboak
- Lehenengo letra xehea
- Zuriune eta gidoirik gabe
- Hitzak lehenengo letra larriz
batuko dira: nireMetodoa



Metodoen erazagupena

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

Auto.java



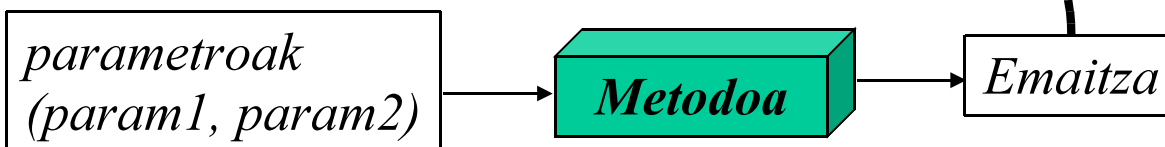
```
(modifikatzaileak) EraitzaMota MetodoIzena(mota1 param1, mota2 param2)  
{  
    //metodoaren kodea  
    return espresioa; //Eraitza void denean ez da jartzen  
}
```



Metodoen deklarazioa

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

Auto.java



```
(modifikatzaileak) EraitzaMota MetodoIzena(mota1 param1, mota2 param2)  
{  
    //metodoaren kodea  
    return espresioa; //Eraitza void denean ez da jartzen  
}
```



Metodoen erazagupena

- Objektuen portaera deskribatzen dute

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 metodo eraikitzailearen bitartez.
- Metodo eraikitzaileek:
 - Klasearen **izen bera** daukate
 - Ez dute ezer itzultzen (ez void, ez ezer)



Metodo eraikitzaileak

Instantziaren sorkuntza

```
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 jasotzen dira:

```
java AdibideKlasea arg1 arg2 ...
```



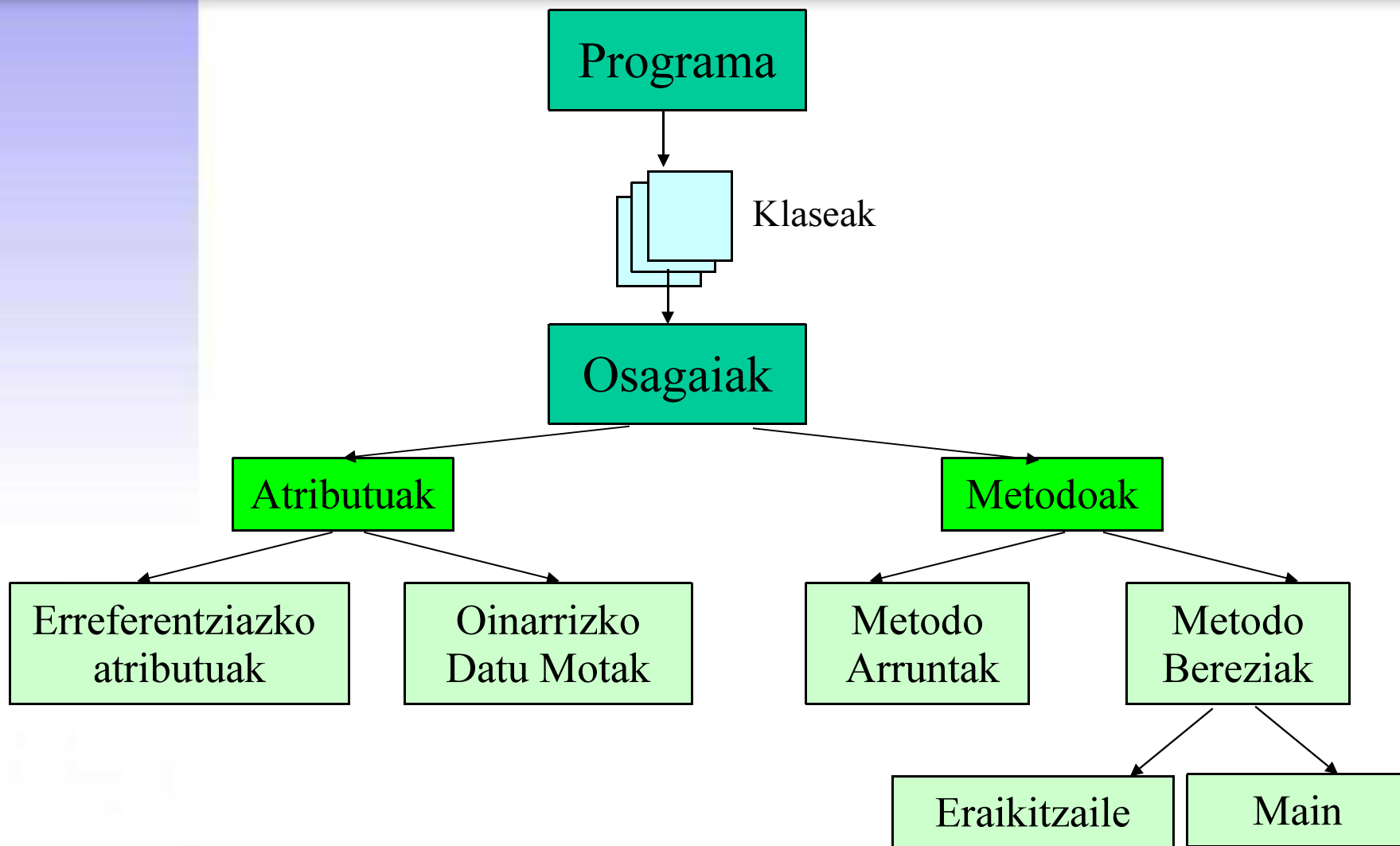
Metodo nagusia (main)

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

- *public*: edozein lekutik dei diezaiokegu
- *static*: klasearen metodo bat dela adierazten du, hau da, ez dago main metodo bat instantzia (objektu) bakoitzeko
- *void*: ez du baliorik itzultzen



Laburpena





Laburpena II: modifikatzaileak

	<i>Ikusgaitasuna</i>				• <i>Klaseen hedapena</i> • <i>Eremu konstanteak</i> • <i>Metodoen birdefinizioa</i>	<i>Kidearen partekotza</i> (heredatu ala ez)	<i>Klase eta metodoen abstraktutasuna</i>
	<i>package</i> (besteri ezean)	<i>public</i>	<i>protected</i>	<i>private</i>	<i>final</i>	<i>static</i>	<i>abstract</i>
Klaseak	X	X	X	X	X		X
Eremuak (aldagaiak)	X	X	X	X	X	X	
Metodoak	X	X	X	X	X	X	X



Exekuzioaren 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

Kontua klasea sortu:

Atributu batekin: kopurua

Hiru eragiketa:

atera(int x) => boolean

sartu(int x) => int (balio berria)

balioaEman() => int



Ariketa

Hurrengo metodoak dituen Osoko
izeneko klasea definitu :

eraikitzailea

isLehena(); // ez ahaztu sarrera

isPositibo(); // eta irteera

inprimatu(); // parametroak

balioaAldatu(int berria); // ...