

IIP (E.T.S. d'Enginyeria Informàtica)  
Curs 2015-2016  
*Pràctica 2. Objectes, classes i programes.*  
*L'entorn BlueJ*

Professors d'IIP  
Departament de Sistemes Informàtics i Computació  
Universitat Politècnica de València



## Índex

<b>1</b>	<b>Context i treball previ a la sessió de pràctiques</b>	<b>2</b>
<b>2</b>	<b>Desenvolupament d'un projecte <i>BlueJ</i></b>	<b>2</b>
2.1	Invocació a <i>BlueJ</i> i creació d'un projecte . . . . .	2
2.2	Operacions d'edició i compilació . . . . .	3
2.3	Execució d'una classe: crides a <code>main</code> . . . . .	5
2.4	L'opció <i>Eines</i> . Generar documentació . . . . .	5
2.5	L'opció <i>Ajuda</i> . . . . .	6
<b>3</b>	<b>Ús del banc d'objectes (<i>Object Bench</i>)</b>	<b>6</b>
3.1	Operacions d'una classe . . . . .	6
3.2	Creació d'un objecte . . . . .	7
3.3	Execució de mètodes sobre l'objecte . . . . .	7
3.4	Observació de l'estat de l'objecte . . . . .	7
<b>4</b>	<b>Ús de l'avaluador d'expressions (<i>Code Pad</i>)</b>	<b>8</b>
<b>5</b>	<b>Ús del depurador (<i>Debugger</i>)</b>	<b>10</b>

# 1 Context i treball previ a la sessió de pràctiques

En el marc acadèmic, aquesta pràctica és la segona del curs i té com a objectiu principal la definició i ús de classes utilitzant l'entorn de programació *BlueJ*. Amb aquesta pràctica es pretén que l'alumne es familiaritze amb els aspectes bàsics de *BlueJ*, i siga capaç d'utilitzar-lo d'ara endavant com a entorn de treball per a les pràctiques de l'assignatura.

L'alumne serà capaç d'utilitzar l'entorn per:

- crear un projecte *BlueJ* amb les classes que es proporcionen a l'inici de la pràctica,
- editar alguna de les classes existents,
- crear objectes i executar mètodes sobre ells utilitzant el banc d'objectes de *BlueJ* (*Object Bench*),
- avaluar expressions utilitzant l'interpret d'instruccions de *BlueJ* (*Code Pad*),
- generar de forma automàtica la documentació del projecte,
- validar el funcionament d'una classe o de qualsevol dels seus mètodes, utilitzant el depurador de *BlueJ* (*Debugger*).

Per tal d'aprofitar al màxim la sessió de pràctiques, s'aconsella a l'alumne que, abans, faci una lectura comprensiva d'aquest butlletí.

## 2 Desenvolupament d'un projecte *BlueJ*

Com ja s'ha indicat en la secció 7 del butlletí de la pràctica 1 (*Introducció: Linux, Java i BlueJ*), un projecte *BlueJ* consta d'un conjunt de classes relacionades entre sí i l'entorn *BlueJ* no permet només desenvolupar el projecte esmentat (crear, compilar, executar, depurar, documentar, etc. el projecte i les seues classes) sino que també *permet interactuar amb qualsevol mètode (atribut u objecte) de qualsevol de les classes incloses en el projecte*. L'objectiu d'aquesta pràctica és que l'alumne es familiaritze amb l'ús de l'entorn *BlueJ*.

Es pot canviar l'idioma de l'entorn *BlueJ* fent servir l'opció **Eines - Preferències - Interface**.

### 2.1 Invocació a *BlueJ* i creació d'un projecte

Com ja es va indicar en la pràctica anterior, *BlueJ* es pot invocar des del menú desplegable de l'entorn gràfic del sistema **Aplicaciones - Programación - BlueJ 3.1.5** o bé des de la línia d'ordres com segueix:

```
bluej &    ó    bluej nomProjecte &
```

Noteu que la segona opció és equivalent a invocar a *BlueJ* sense arguments i després, utilitzant l'opció **Projecte - Obre Projecte...** del menú, seleccionar el projecte **nomProjecte**. Si es vol obrir amb *BlueJ* una aplicació ja existent però que **no** ha estat desenvolupada amb *BlueJ*, s'ha d'invocar a *BlueJ* sense arguments i aleshores obrir l'aplicació existent amb l'opció **Projecte - Obre No BlueJ...** del menú.

## Activitat #1

1. Crear el subdirectori **pract2** en el directori de treball **iip**.
2. Descarregar els fitxers **.java** i **.class** disponibles a la carpeta **Recursos/Laboratorio/Práctica 2/Código de PoliformaT** al directori **pract2**.
3. Invocar a *BlueJ* sense arguments.
4. Obrir la carpeta **pract2** amb l'opció **Projecte - Obre No BlueJ...** del menú.

En la finestra principal de *BlueJ* apareixen les icones de cadascuna de les classes del projecte (Figura 1). Noteu que les icones associades als fitxers **.class** apareixen marcades amb el text **(no source)** i les associades als fitxers **.java** apareixen ratllades si encara no han estat compilats. Les fletxes indiquen les relacions d'ús que existeixen entre les classes.

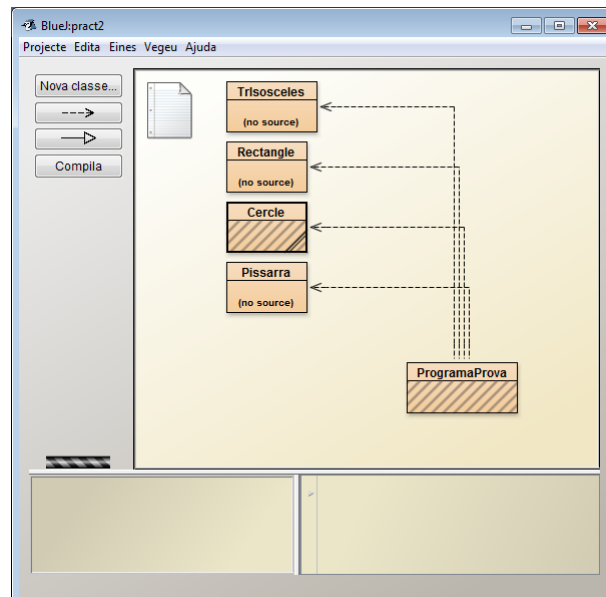


Figura 1: Projecte **pract2**.

## 2.2 Operacions d'edició i compilació

Per editar una classe, en el nostre cas **Cercle**, es pot usar l'opció **Obre Editor** del menú de la classe, o també fer doble clic sobre la icona de la classe.

Una classe es pot compilar des del propi editor o, *si es desitja compilar tot el projecte*, des de la barra de ferramentes de la finestra principal de *BlueJ*. Si el compilador detecta algun error, la línia que el conté apareix ombrejada; així mateix, apareixerà un missatge d'error a la *zona d'informació* de l'editor, a la part inferior de la pantalla (Figura 2). És més, si es clica el botó amb el símbol "?" que apareix a l'extrem dret de la zona d'informació de l'editor, es pot obtenir més informació sobre l'error detectat.

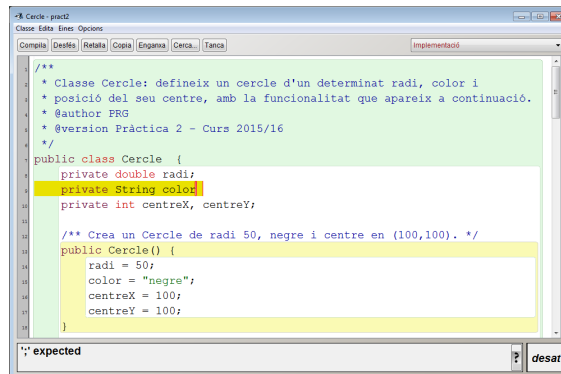


Figura 2: Compilació de la classe Cercle.

## Activitat #2

1. Compilar totes les classes, corregir els possibles errors de compilació i observar què passa a la part central de la finestra principal de *BlueJ*.

Una de les característiques que ofereix l'editor de *BlueJ* és l'autocompletat de codi que s'activa en polsar les tecles **Ctrl-Space**.

## Activitat #3

1. Editar la classe *ProgramaProva* i completar la instrucció que mostra per pantalla el perímetre de l'objecte *Cercle* *c*. Escriure *c.* i polsar **Ctrl-Space**. Comprovar que, com en la figura 3, apareix un llistat de tots els mètodes disponibles en la classe junt amb una xicoteta descripció dels mateixos. Seleccionar el que interesse i després de polsar **Enter**, *BlueJ* ho autocompletarà.

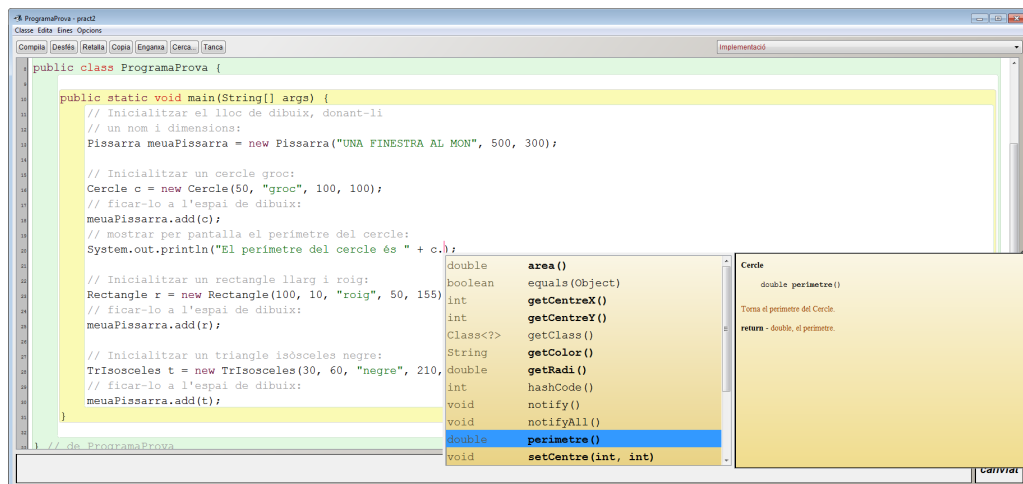


Figura 3: Autocompletat de codi en l'editor de *BlueJ*.

## 2.3 Execució d'una classe: crides a main

Com ja s'ha dit anteriorment, al marcar la icona d'una classe i fer clic amb el botó dret del ratolí, s'obté el menú de la classe; doncs bé, de la llista d'operacions que aquest conté destacarem ara l'operació de crida al mètode `main` de la classe; *es pot executar una classe des del seu menú*. No és necessari passar cap argument al `main` de `ProgramaProva`, només apareixen les claus.

A la figura 4(a) es mostra el menú emergent de la classe `ProgramaProva` i a la figura 4(b) la finestra de la crida al mètode `main` de la classe `ProgramaProva`.

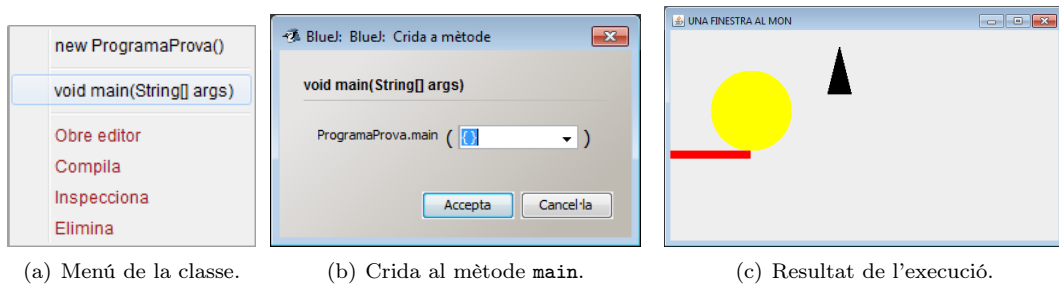


Figura 4: Execució de la classe `ProgramaProva`.

En cas de que el programa sol·licite dades des del teclat o mostre per pantalla algun resultat (Figura 5), apareix de forma automàtica un terminal de text. Si no apareix, s'ha de seleccionar l'opció **Mostrar Terminal** del menú **Vegeu**.

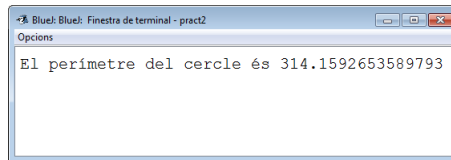


Figura 5: Finestra de terminal de *BlueJ*.

### Activitat #4

1. Executar el mètode `main` de la classe `ProgramaProva`. El resultat ha de ser com el que apareix a la figura 4(c) i en el terminal s'ha de mostrar el missatge de la figura 5.

## 2.4 L'opció *Eines*. Generar documentació

De les diferents utilitats de l'opció **Eines** cal destacar les següents:

- **Documentació del projecte**, que genera el subdirectori `doc` amb la documentació sobre les classes en format `html`. Noteu que la documentació individual d'una classe es pot generar a l'editar la classe i seleccionar, en lloc d'**Implementació**, l'opció **Interfície**.
- **Preferències...**, que en la seua opció **Miscel·lània** permet definir la ubicació de la documentació de les classes del llenguatge.

## Activitat #5

1. Generar la documentació del projecte i consultar-la. El resultat serà com el que apareix a la figura 6.



Figura 6: Documentació del projecte pract2.

## 2.5 L'opció Ajuda

La utilitat **Llibreries de classes Java** permet accedir a la documentació de Java. En la instal·lació per defecte aquesta ajuda es troba a <http://docs.oracle.com/javase/8/docs/api/>. En la instal·lació del laboratori es troba a un directori local, la qual cosa s'haurà de comprovar des de l'opció **Eines/Preferències/Miscel·lània** del menú.

Altres utilitats d'Ajuda permeten consultar el manual de *BlueJ* així com accedir a la seua web [www.bluej.org](http://www.bluej.org).

## 3 Ús del banc d'objectes (*Object Bench*)

Una de les característiques més interessants de l'entorn *BlueJ* és que permet interactuar amb objectes aïllats de qualsevol classe i executar els mètodes que sobre ells s'hagen definit; d'aquesta manera es pot comprovar la funcionalitat de la classe abans d'escriure qualsevol aplicació que la utilitze.

### 3.1 Operacions d'una classe

Per accedir a les operacions aplicables a una determinada classe hi ha que marcar la icona de la classe i fer clic amb el botó dret del ratolí. Apareix una llista amb les operacions constructores

de la classe i altres operacions permeses per l'entorn com, per exemple, esborrar la classe o compilar-la (Figura 7).

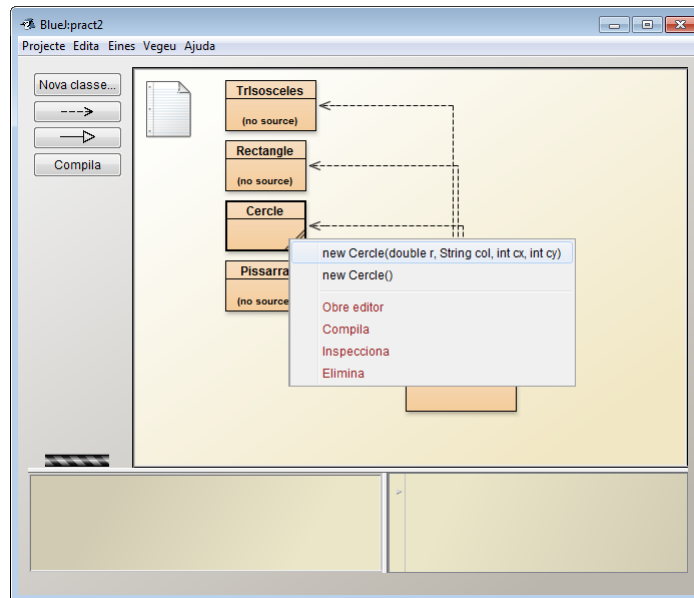


Figura 7: Menú de la classe **Cercle**.

### 3.2 Creació d'un objecte

Si es vol crear un objecte s'ha de seleccionar d'aquest menú una de les operacions constructors i seguir el quadre de diàleg que s'obri (Figura 8(a)). En concret, es demana un nom per a l'objecte. Quan es crea aquest objecte, apareix a la part inferior esquerra de la pantalla principal de *BlueJ*, a la zona coneguda com *banc d'objectes* (*Object Bench*) (Figura 8(b)).

### 3.3 Execució de mètodes sobre l'objecte

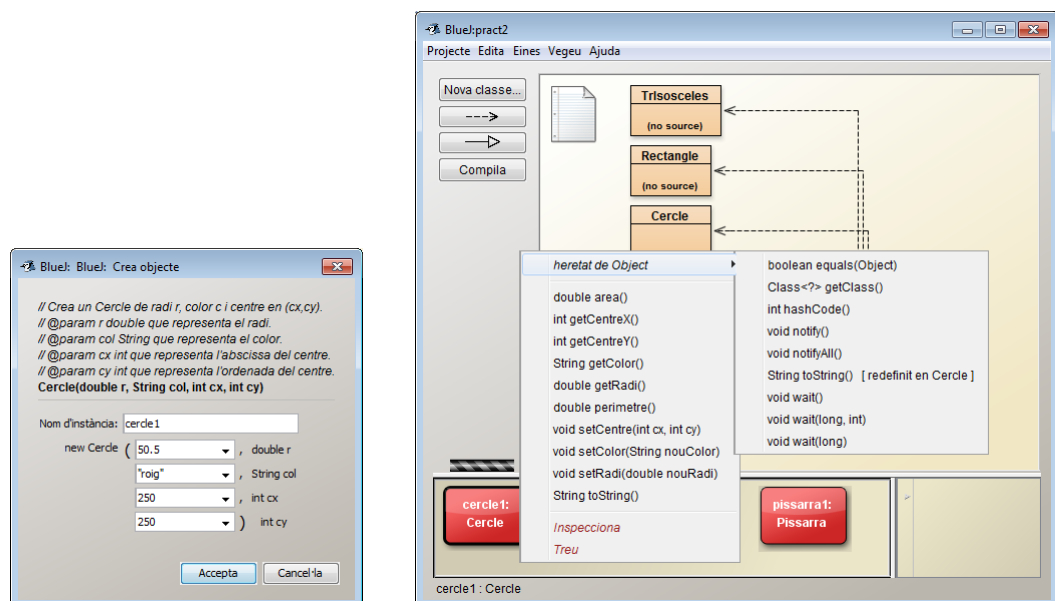
Si es clica amb el botó dret del ratolí sobre l'objecte creat s'accedeix als mètodes que es poden executar sobre el mateix (Figura 8(b)). Per executar un d'ells només cal seleccionar-lo. Si l'objecte hereta mètodes d'altres classes també apareixen a través de submenús.

### 3.4 Observació de l'estat de l'objecte

Per depurar els mètodes dissenyats es pot utilitzar l'opció **Inspecciona**. Aquesta operació permet conèixer els valors dels camps (atributs) dels objectes (Figura 8(c)).

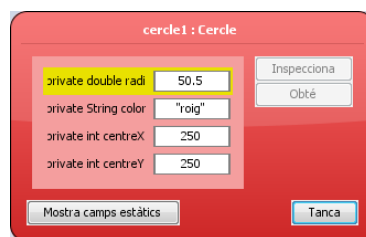
## Activitat #6

1. Crear un objecte de la classe **Cercle** de radi 50.5, color roig i amb centre en (250,250).
2. Consultar els valors dels atributs de l'objecte creat.
3. Executar el mètode `toString()` definit a la classe **Cercle** sobre l'objecte creat.



(a) Creació d'un objecte **Cercle**.

(b) Banc d'objectes i menú de l'objecte **Cercle**.



(c) Inspecció de l'objecte **Cercle**.

Figura 8: Creació d'un objecte en el banc d'objectes de *BlueJ*, menú i observació de l'estat d'aquest objecte.

4. Modificar el radi del **Cercle** perquè valga 30.0.
5. Executar novament el mètode `toString()`.
6. Crear un objecte de la classe **Pissarra** amb títol "Dibuix" i dimensió 500 x 500. Per tal que pugui veure's l'efecte de l'ítem 8, l'objecte creat no ha de tancar-se.
7. Consultar els valors dels atributs de l'objecte creat.
8. Afegir l'objecte **Cercle** a la **Pissarra**.

## 4 Ús de l'avaluador d'expressions (*Code Pad*)

La zona de codi (*Code Pad*) de *BlueJ* està situada al cantó inferior dret junt al banc d'objectes (Figura 9). Si no es mostra, s'ha de seleccionar l'opció **Mostrar el quadern de notes** del menú **Vegeu**.



Aquesta zona es pot utilitzar per introduir expressions o instruccions en Java; polsant *Enter*, cada línia serà avaluada i es mostrarà el valor resultant, seguit pel seu tipus (entre parèntesi), o un missatge d'error si l'expressió és incorrecta. Es poden també utilitzar objectes del banc d'objectes.

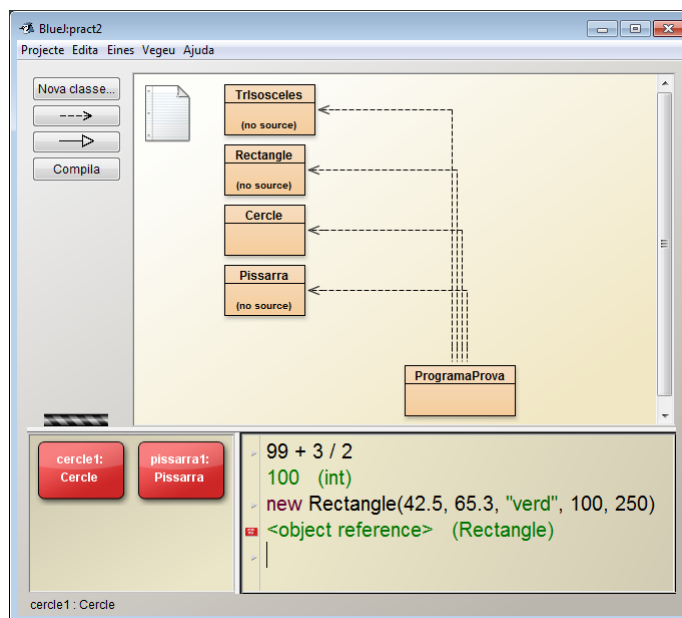


Figura 9: Zona de codi de *BlueJ*.

Alguns resultats d'expressions són objectes en lloc de valors simples. En aquest cas l'objecte es mostra com una referència a objecte **<object reference>**, seguida pel tipus de l'objecte i es mostra una icona menuda representativa de l'objecte al costat de la línia de resultat. Aquesta icona es pot utilitzar ara per continuar treballant amb l'objecte resultant. Es pot arrossegar la icona al banc d'objectes. Això situarà l'objecte al banc, on estarà disponible per a futures crides als seus mètodes, bé via el seu menú emergent o bé via la zona de codi.

## Activitat #7

1. Quin resultat s'obté en avaluar cadascuna de les expressions següents a la zona de codi de *BlueJ*?

1	$99 + 3$	6	$9 / 2$
2	$8 \% 3$	7	$9.0 / 2.0$
3	$-8 \% 3$	8	$9 / 2.0$
4	<code>Math.sqrt(121)</code>	9	$9 / (\text{double}) 2$
5	<code>Math.sqrt(-5)</code>	10	$9 / 0$

2. Declarar i inicialitzar tres variables *h*, *m* i *s* que representen, respectivament, hores, minuts i segons (en format 24 hores) i escriure una expressió per tal de comprovar si es tracta d'una hora correcta.

3. Crear un objecte de la classe `Cercle` al banc d'objectes (utilitzant el menú emergent de la classe com s'ha descrit anteriorment). Anomenar-lo `cercleA`.
4. Escriure a la zona de codi una expressió que torne el radi de `cercleA`.
5. Crear un nou objecte `Cercle` a la zona de codi i arrossegant la seua icona al banc d'objectes. Anomenar-lo `cercleB`.
6. Escriure a la zona de codi una expressió que torne el color de `cercleB`. L'objecte `String` resultat es pot també arrastrar al banc d'objectes.

## 5 Ús del depurador (*Debugger*)

El *depurador* de *BlueJ* és una eina senzilla, però de gran utilitat a l'hora de validar el funcionament d'una classe (és a dir, el del seu `main`) o qualsevol mètode d'aquesta. El seu ús permet, bàsicament,

- observar l'execució de qualsevol mètode, és a dir, *fer la seua traça per uns valors donats*, bé pas a pas, bé de només alguna(es) de les seues línies;
- inspeccionar la seqüència de crides associada a la invocació del mètode en execució;
- comprovar el pas de paràmetres i els valors que prenen les variables locals al mètode en execució.

Per aconseguir aquests resultats, el depurador disposa de les següents funcions:

1. **Establir punts de ruptura.** Només quan es deté l'execució d'un mètode en un cert punt del seu codi és possible observar l'estat de la seua execució en aquest punt. El depurador proporciona una funció que deté l'execució d'un mètode en un cert punt del codi, o equivalentment, *estableix punts de ruptura*.

A *BlueJ* els punts de ruptura s'estableixen en l'anomenada *àrea de punts de ruptura* de l'editor, situada a l'esquerra del text; només cal fer-hi clic, a l'altura de la línia de codi on es vol detindre l'execució d'un mètode, i apareixerà un xicotet signe d'stop com a marca de punt de ruptura. Quan durant l'execució d'un mètode s'arriba a la línia així marcada, l'execució s'interromp. A més, apareixen, una darrere l'altra,

(a) *la finestra de l'editor*, on figura ressaltada la línia següent a la que conté el punt de ruptura, ja que és *la següent línia a executar*;

(b) *la finestra del depurador*; els diferents tipus d'informació i botons que conté aquesta finestra es presenten a continuació.

2. **Execució pas a pas.** Una vegada detinguda l'execució, aquesta es pot reprendre pas a pas, instrucció a instrucció, el que permet seguir el codi observant com progressa l'execució, és a dir, *fer una traça* del codi.

Per realitzar una execució pas a pas en *BlueJ* n'hi ha prou amb fer clic repetidament sobre el botó **Pas** de la finestra del depurador. Cada clic suposa l'execució d'una única línia de codi, després d'això l'execució es torna a detindre.

Si es vol sortir d'aquest procés, tornar a l'execució normal del mètode, només cal esborrar la marca de ruptura establerta, simplement fent clic sobre ella, i després pulsar el botó **Continua** de la finestra del depurador.

### 3. Inspecció de variables i del pas de paràmetres.

Només amb observar la finestra de depuració de *BlueJ* es pot veure la seqüència de crides associada a la invocació del mètode en execució, comprovar el pas de paràmetres i inspeccionar els valors que prenen les seues variables locals.

Per inspeccionar qualsevol variable o paràmetre d'aquest tipus només cal fer un doble clic sobre ell, en la finestra del depurador.

## Activitat #8

1. En el mètode `main` de la classe `ProgramaProva`, establir un punt de ruptura en les línies on es creen els objectes de tipus `Cercle`, `Rectangle` i `TrIsosceles`.
2. Executar el mètode `main`. Observeu què passa quan, una vegada arribat al punt de ruptura es fa clic sobre el botó `Pas` de la finestra del depurador.
3. Per inspeccionar les variables del `main`, fer doble clic sobre elles en la finestra del depurador.

A la figura 10 es mostra el depurador de *BlueJ* una vegada alcançat l'últim punt de ruptura en l'execució de la classe `ProgramaProva`.

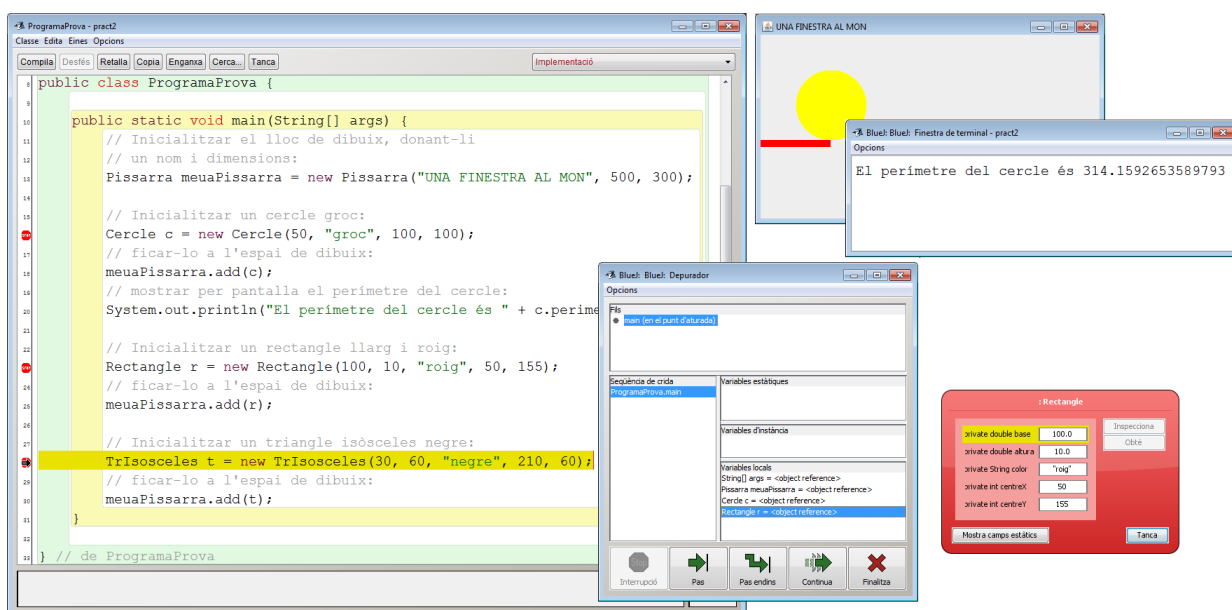


Figura 10: Depurador de *BlueJ*.