```java
/*********************************************************************
 *    PROGRAMA    : Ejes cartesianos
 *    AUTOR       : Gabriel Hernandez Grajeda
 *    FECHA       : 2022-03-19
 *    DESCRIPCION : Dibuja los ejes cartesianos (opcional) con sus marcas de separacion
  (opcional)
 *********************************************************************/

import java.awt.*;

import java.awt.event.*;

import javax.swing.*;
import javax.swing.event.ChangeEvent;
import javax.swing.event.ChangeListener;



public class Ejes_cartesianos {

    public static void main(String[] args) {
        //ACOMODANDO EL MARCO

        MarcoCartesianos mimarco = new MarcoCartesianos();

        mimarco.setTitle("Ejes cartesianos");

        mimarco.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        mimarco.setVisible(true);


    }



}


class MarcoCartesianos extends JFrame{

    PanelCartesianos milamina3 = new PanelCartesianos();

    public MarcoCartesianos() {


        this.setBounds(200, 200, 500, 500);

        this.setLayout(new BorderLayout());

        JPanel panelNorte = new JPanel();

        panelNorte.setLayout(new GridLayout(3,1));

        panelNorte.setBorder(BorderFactory.createEtchedBorder());
        //ETIQUETAS

        JPanel mipanel1 = new JPanel();

        mipanel1.setLayout(new FlowLayout(FlowLayout.LEFT));
```

```
62
63
64          JLabel xmmin_label = new JLabel("XMmin");
65
66          JLabel ymmin_label = new JLabel("YMmin");
67
68          JLabel xmmax_label = new JLabel("XMmax");
69
70          JLabel ymmax_label = new JLabel("YMmax");
71
72          //Agregar etiquetas
73
74          mipanel1.add(xmmin_label);
75
76          mipanel1.add(Box.createHorizontalStrut(8));
77
78          mipanel1.add(ymmin_label);
79
80          mipanel1.add(Box.createHorizontalStrut(8));
81
82          mipanel1.add(xmmax_label);
83
84          mipanel1.add(Box.createHorizontalStrut(8));
85
86          mipanel1.add(ymmax_label);
87
88
89          panelNorte.add(mipanel1);
90
91
92
93          //CONTROLES
94
95          JPanel mipanel2 = new JPanel();
96
97          //xmmin
98
99          mipanel2.setLayout(new FlowLayout(FlowLayout.LEFT));
100
101         JSpinner control_xmmin = new JSpinner(new SpinnerNumberModel(-1,-1000,-1,1));
102
103         control_xmmin.setPreferredSize(new Dimension(50,15));
104
105         control_xmmin.addChangeListener(new ChangeListener() {
106
107             @Override
108             public void stateChanged(ChangeEvent e) {
109
110                 milamina3.set_xmmin((int)control_xmmin.getValue());
111
112                 milamina3.repaint();
113             }
114
115
116         });
117
118         mipanel2.add(control_xmmin);
119
120
121         //ymmin
122         JSpinner control_ymmin = new JSpinner(new SpinnerNumberModel(-1,-1000,-1,1));
123
```

```
124         control_ymmin.setPreferredSize(new Dimension(50,15));
125 
126         control_ymmin.addChangeListener(new ChangeListener() {
127 
128             @Override
129             public void stateChanged(ChangeEvent e) {
130 
131                 milamina3.set_ymmin((int)control_ymmin.getValue());
132 
133                 milamina3.repaint();
134             }
135 
136 
137         });
138 
139         mipanel2.add(control_ymmin);
140 
141 
142         //xmmax
143         JSpinner control_xmmax = new JSpinner(new SpinnerNumberModel(1,1,1000,1));
144 
145         control_xmmax.setPreferredSize(new Dimension(50,15));
146 
147         control_xmmax.addChangeListener(new ChangeListener() {
148 
149             @Override
150             public void stateChanged(ChangeEvent e) {
151 
152                 milamina3.set_xmmax((int)control_xmmax.getValue());
153 
154                 milamina3.repaint();
155             }
156 
157 
158         });
159 
160         mipanel2.add(control_xmmax);
161 
162 
163         //ymmax
164         JSpinner control_ymmax = new JSpinner(new SpinnerNumberModel(1,1,1000,1));
165 
166         control_ymmax.setPreferredSize(new Dimension(50,15));
167 
168         control_ymmax.addChangeListener(new ChangeListener() {
169 
170             @Override
171             public void stateChanged(ChangeEvent e) {
172 
173                 milamina3.set_ymmax((int)control_ymmax.getValue());
174 
175                 milamina3.repaint();
176             }
177 
178 
179         });
180 
181         mipanel2.add(control_ymmax);
182 
183 
184         panelNorte.add(mipanel2);
185 
```

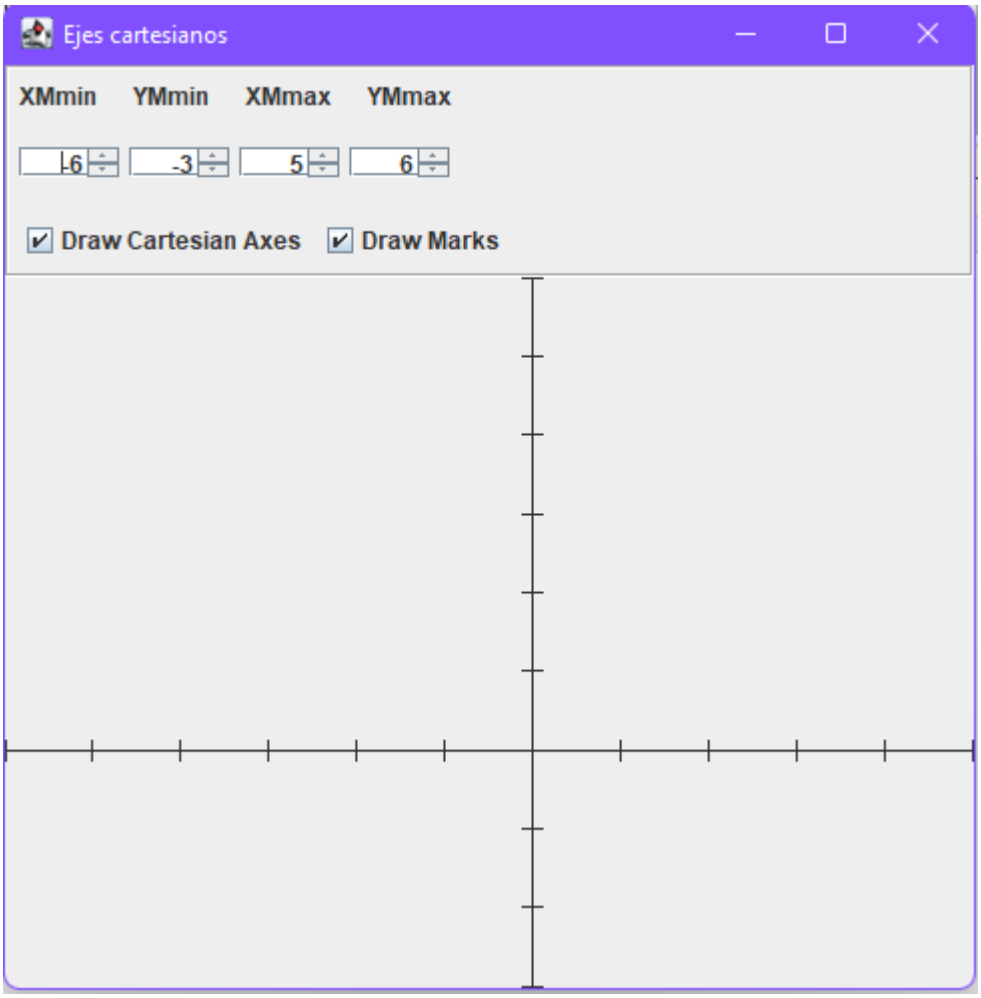
```
186         //CHECKBOX
187 
188         JPanel mipanel3 = new JPanel();
189 
190         mipanel3.setLayout(new FlowLayout(FlowLayout.LEFT));
191 
192 
193 
194 
195         JCheckBox check1 = new JCheckBox("Draw Cartesian Axes");
196 
197         JCheckBox check2 = new JCheckBox("Draw Marks");
198 
199 
200         check1.addActionListener(new ActionListener() {
201 
202             @Override
203             public void actionPerformed(ActionEvent e) {
204                 // TODO Auto-generated method stub
205                 milamina3.setdrawCartesianAxes(!milamina3.getdrawCartesianAxes());
206 
207                 milamina3.repaint();
208             }
209 
210         });
211 
212 
213         check2.addActionListener(new ActionListener() {
214 
215             @Override
216             public void actionPerformed(ActionEvent e) {
217                 // TODO Auto-generated method stub
218                 milamina3.setdrawMarks(!milamina3.getdrawMarks());
219 
220                 milamina3.repaint();
221             }
222 
223         });
224 
225         mipanel3.add(check1);
226 
227         mipanel3.add(check2);
228 
229 
230         panelNorte.add(mipanel3);
231 
232 
233 
234         //Agregar panel de etiquetas, spinners y checkbox
235         this.add(panelNorte,BorderLayout.NORTH);
236 
237         //PANEL QUE DIBUJA EL PLANO CARTESIANO
238 
239         this.add(milamina3);
240 
241 
242     }
243 
244 }
245 
246 
247 
```

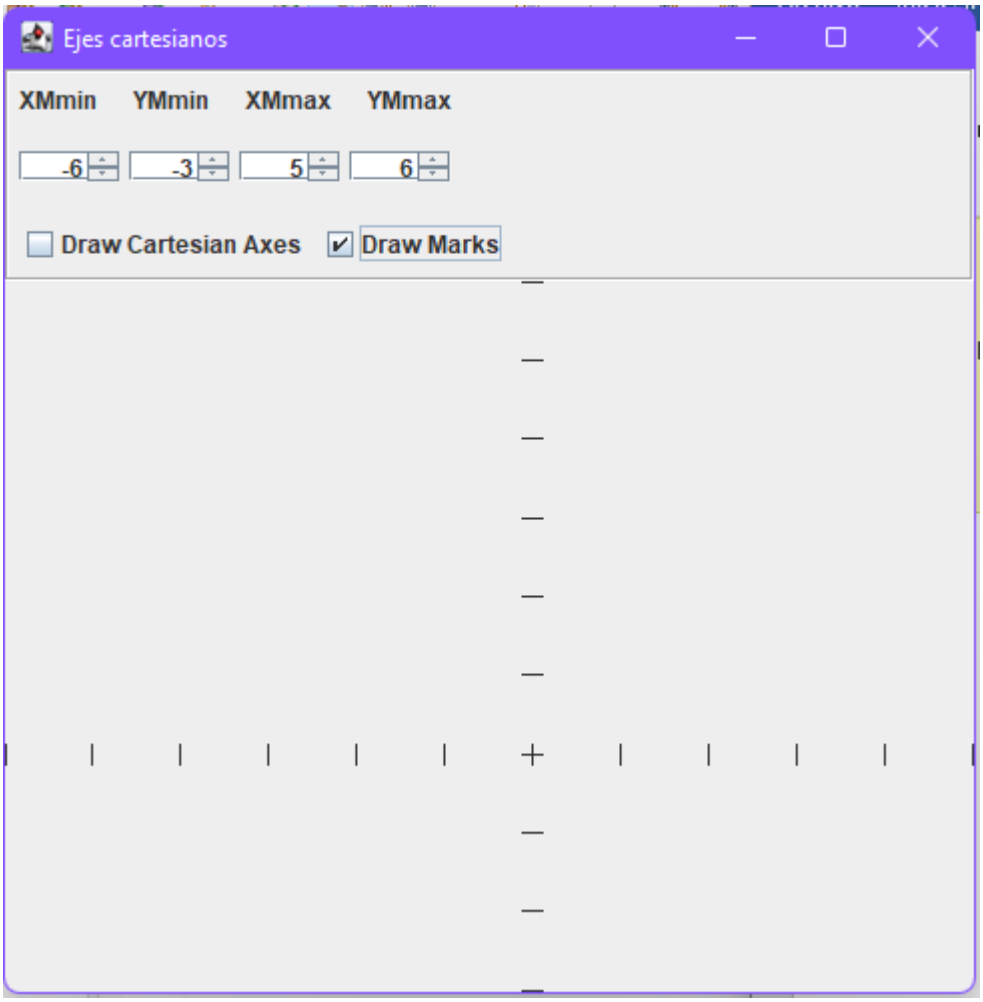
```
248
249 class PanelCartesianos extends JPanel{
250
251     private static Graphics g;
252     private Coordenadas cor = new Coordenadas();  // la mitad del ancho de las lineas
    separadoras es 5 pixeles
253     private int xmmin=-1,ymmin=-1,xmmax=1,ymmax=1,mitadAncho=5;
254     private boolean drawMarks = false ,drawCartesianAxes = false;
255
256     public void set_xmmin(int valor) {
257
258         this.xmmin = valor;
259
260     }
261
262     public void set_ymmin(int valor) {
263
264         this.ymmin=valor;
265
266     }
267
268     public void set_xmmax(int valor) {
269
270         this.xmmax=valor;
271     }
272
273     public void set_ymmax(int valor) {
274
275         this.ymmax=valor;
276
277     }
278
279
280     public void setdrawCartesianAxes(boolean estado) {
281
282         this.drawCartesianAxes = estado;
283
284     }
285
286
287     public void setdrawMarks(boolean estado) {
288
289         this.drawMarks=estado;
290
291     }
292
293
294     public boolean getdrawCartesianAxes() {
295
296         return this.drawCartesianAxes;
297     }
298
299     public boolean getdrawMarks() {
300
301         return this.drawMarks;
302     }
303
304
305
306
307     public void paintComponent(Graphics g) {
308
```

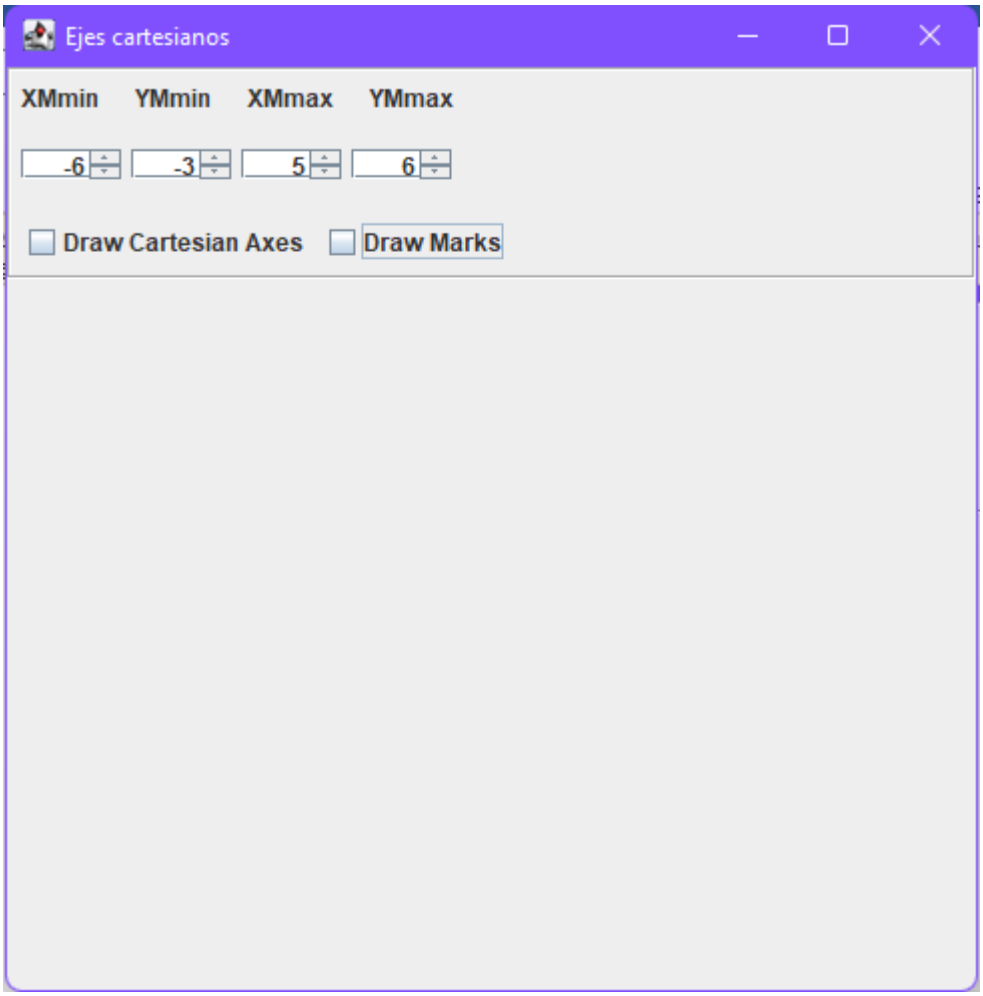
```
309         super.paintComponent(g);
310 
311         this.g=g;
312 
313         this.coordenadas();
314 
315 
316     }
317 
318     public void coordenadas() {
319 
320 
321 
322         //ANCHO Y ALTO DEL PANEL
323         int maxX = g.getClipBounds().width-1;
324         int maxY = g.getClipBounds().height-1;
325 
326         cor.setRangoDispositivo(0, 0, maxX, maxY); //Establecemos las medidas del
    dispositivo
327 
328         cor.setRangoMundiales(xmmin, ymmin, xmmax, ymmax); //Establecemos las medidas del
    plano cartesiano
329 
330         if(this.getdrawCartesianAxes()==true) {
331 
332             //Linea vertical plano cartesiano
333             cor.setXM(0);
334             cor.setYM(ymmax);
335 
336             int xdv1 = cor.getXD();
337             int ydv1 = cor.getYD();
338 
339             cor.setXM(0);
340             cor.setYM(ymmin);
341 
342             int xdv2 = cor.getXD();
343             int ydv2 = cor.getYD();
344 
345             g.drawLine(xdv1, ydv1, xdv2, ydv2);
346 
347             //Linea horizontal plano cartesiano
348 
349             cor.setXM(xmmin);
350             cor.setYM(0);
351 
352             int xdh1 = cor.getXD();
353             int ydh1 = cor.getYD();
354 
355             cor.setXM(xmmax);
356             cor.setYM(0);
357 
358             int xdh2 = cor.getXD();
359             int ydh2 = cor.getYD();
360 
361             g.drawLine(xdh1, ydh1, xdh2, ydh2);
362 
363         }
364 
365 
366 
367         if(this.getdrawMarks()==true) {
368 
```

```
369             //Separaciones eje X
370 
371             cor.setYM(0);
372             for(int i=xmmin;i<=xmmax;i++) {
373 
374                 cor.setXM(i);
375 
376                 g.drawLine(cor.getXD(), cor.getYD()-mitadAncho, cor.getXD(), cor.getYD
    ()+mitadAncho);
377             }
378 
379 
380             //Separaciones eje Y
381             cor.setXM(0);
382             for(int i=ymmin;i<=ymmax;i++) {
383 
384                 cor.setYM(i);
385 
386                 g.drawLine(cor.getXD()-mitadAncho, cor.getYD(), cor.getXD()+mitadAncho,
    cor.getYD());
387             }
388 
389 
390         }
391 
392 
393 
394 
395     }
396 
397 }
398 
399 class Coordenadas{
400 
401     double XMmin, YMmin, XMmax,YMmax,XM,YM;
402     int XDmin,YDmin,XDmax,YDmax;
403 
404     public void setRangoMundiales(double XMmin, double YMmin, double XMmax, double YMmax){
405 
406         this.XMmin=XMmin;
407         this.YMmin=YMmin;
408         this.XMmax=XMmax;
409         this.YMmax=YMmax;
410     }
411 
412     public void setRangoDispositivo(int XDmin,int YDmin,int XDmax, int YDmax) {
413 
414         this.XDmin=XDmin;
415         this.YDmin=YDmin;
416         this.XDmax=XDmax;
417         this.YDmax=YDmax;
418     }
419 
420     public void setXM(double XM) {
421         this.XM=XM;
422     }
423 
424     public void setYM(double YM) {
425         this.YM=YM;
426     }
427 
428     public double getXM() {
```

```
429         return XM;
430     }
431 
432     public double getYM() {
433         return YM;
434     }
435 
436     public int getXD() {
437 
438         double calculo = XDmin + (XM-XMmin)*(XDmax-XDmin)/(XMmax-XMmin);
439 
440         return (int)calculo;
441 
442     }
443 
444     public int getYD() {
445 
446         double calculo = YDmax - (YM-YMmin)*(YDmax-YDmin)/(YMmax-YMmin);
447 
448         return (int)calculo;
449 
450     }
451 }
452 
453 
```


Ejes cartesianos
XMmin
YMmin
XMmax
YMmax
-6
-3
5
6
Draw Cartesian Axes
Draw Marks


Ejes cartesianos
XMmin
YMmin
XMmax
YMmax
-6
-3
5
6
Draw Cartesian Axes
Draw Marks


Ejes cartesianos
XMmin
YMmin
XMmax
YMmax
-6
-3
5
6
Draw Cartesian Axes
Draw Marks