

Assignment 8: Save & Load

```
JMenuItem save = new JMenuItem("Save");  
save.setAccelerator(KeyStroke.getKeyStroke("control S"));  
fileMenu.add(save);  
save.addActionListener(e -> doSave());  
  
JMenuItem open = new JMenuItem("Open");  
fileMenu.add(open);  
open.setAccelerator(KeyStroke.getKeyStroke("control O"));  
open.addActionListener(e -> doOpen());
```

doSave (1/2)

```
private void doSave() {  
    JFileChooser chooser = new JFileChooser(currentDirectory);  
    chooser.setDialogTitle("Save Graphic");  
    chooser.setDialogType(JFileChooser.SAVE_DIALOG);  
    FileFilter filter = new FileFilter() {  
        @Override  
        public String getDescription() {  
            return "JDraw Graphic (*.draw)";  
        }  
        @Override  
        public boolean accept(File f) {  
            return f.isDirectory() || f.getName().endsWith(".draw");  
        }  
    };  
    chooser.setFileFilter(filter);  
  
    int res = chooser.showSaveDialog(this);  
    ...  
}
```

doSave (2/2)

```
if (res == JFileChooser.APPROVE_OPTION) {  
    try {  
        File file = chooser.getSelectedFile();  
        if(chooser.getFileFilter() == filter && !filter.accept(file)) {  
            file = new File(chooser.getCurrentDirectory(),  
                            file.getName() + ".draw");  
        }  
        try(ObjectOutputStream oos = new ObjectOutputStream(  
            new FileOutputStream(file))) {  
            for (Figure f : getModel().getFigures()) {  
                oos.writeObject(f.clone());  
            }  
            oos.writeObject(null);  
        }  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

Writes copies (which do not contain listeners)

Marks end of stream

doOpen (1/2)

```
private void doOpen() {  
    JFileChooser chooser = new JFileChooser(currentDirectory);  
    chooser.setDialogTitle("Open Graphic");  
    chooser.setDialogType(JFileChooser.OPEN_DIALOG);  
    chooser.setFileFilter(new javax.swing.filechooser.FileFilter() {  
        @Override  
        public String getDescription() {  
            return "JDraw Graphic (*.draw)";  
        }  
        @Override  
        public boolean accept(File f) {  
            return f.isDirectory() || f.getName().endsWith(".draw");  
        }  
    });  
  
    int res = chooser.showOpenDialog(this);  
    ...  
}
```

doOpen (2/2)

```
if (res == JFileChooser.APPROVE_OPTION) {  
    try (ObjectInputStream ois = new ObjectInputStream(  
        new FileInputStream(chooser.getSelectedFile()))) {  
        DrawModel m = getModel();  
        m.removeAllFigures();  
        while (true) {  
            try {  
                Object x = ois.readObject();  
                if(x == null) { break; }  
                m.addFigure((Figure) x);  
            } catch (ClassNotFoundException e) {  
                System.out.println("Figure not found, "+e.getMessage());  
            }  
        }  
    } catch (Exception e) { System.out.println(e); }  
}
```

End of stream reached

Ignore unknown figures