

Licence Info 3^{ème} année - module Programmation Objet 2

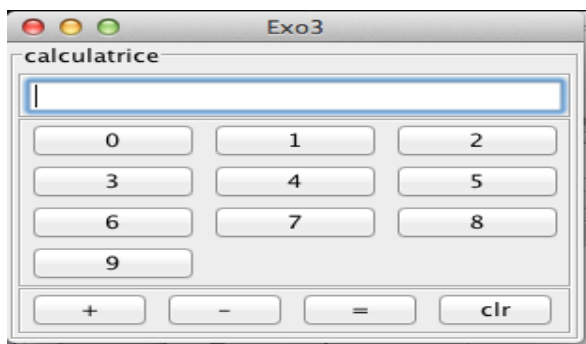
Examen – janvier 2017 – session 1 – durée : 2h

Documents autorisés : les polycopiés du cours (pas de TD/TP)

Partie A : Programmation graphique (environ 10 points)

Nous vous donnons un programme qui a réalisé une interface graphique d'un calculateur qui contient (c'est l'exo3 que vous avez fait en TD/TP) :

- 10 boutons correspondant aux chiffres : 0, 1, ..., 9
- 4 boutons correspondant aux opérations : + - = clr
- 1 textfield pour afficher le résultat



```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class Exo3 extends JFrame {
    private JPanel panelAffichage;
    private JPanel panelChiffre;
    private JPanel panelOperation;
    private JPanel panelTout;

    private JButton bouton;

    private JTextField tfDisplay;
    private int result = 0; // le resultat
    private String numberInStr = ""; // le nombre saisi en String
    private int previousNb = 0; // le premier operand
    private int currentNb = 0; // le seconde operand
    private String previousOpr = ""; // le premier operateur
    private String currentOpr = ""; // le seconde operateur

    // constructeur
    public Exo3() {
        super("Exo3");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        panelAffichage = new JPanel(new BorderLayout());
        panelAffichage.setBorder(BorderFactory.createEtchedBorder());
        tfDisplay = new JTextField("", 6);
        panelAffichage.add(tfDisplay, BorderLayout.CENTER);

        panelChiffre = new JPanel(new GridLayout(4, 3));
        panelChiffre.setBorder(BorderFactory.createEtchedBorder());

        NumberBtnListener numBtnListener = new NumberBtnListener();
```

```

for (int i=0; i<10; i++) {
    bouton = new JButton(""+i);
    panelChiffre.add(bouton);
    bouton.addActionListener(numBtnListener);
}

OprBtnListener oprBtnListener = new OprBtnListener();

panelOperation = new JPanel(new GridLayout(1,4));
panelOperation.setBorder(BorderFactory.createEtchedBorder());
bouton = new JButton("+");
bouton.addActionListener(oprBtnListener);
panelOperation.add(bouton);
bouton = new JButton("-");
bouton.addActionListener(oprBtnListener);
panelOperation.add(bouton);
bouton = new JButton("=");
bouton.addActionListener(oprBtnListener);
panelOperation.add(bouton);

bouton = new JButton("clr");
bouton.addActionListener(new ActionListener() {

    public void actionPerformed(ActionEvent e) {
        tfDisplay.setText("");
        currentOpr = "";
        currentNb = 0;
        numberInStr = "";
    }
});

panelOperation.add(bouton);

panelTout = new JPanel(new BorderLayout());
panelTout.setBorder(BorderFactory.createTitledBorder("calculatrice"));
panelTout.add(panelAffichage, BorderLayout.NORTH);
panelTout.add(panelChiffre, BorderLayout.CENTER);
panelTout.add(panelOperation, BorderLayout.SOUTH);

this.setContentPane(panelTout);
this.pack();
this.setVisible(true);
}

// Ecouteur pour des boutons numeriques (inner class)
class NumberBtnListener implements ActionListener {
    public void actionPerformed(ActionEvent evt) {
        numberInStr += evt.getActionCommand();
        tfDisplay.setText(numberInStr);
    }
}

// Ecouteur pour des boutons operateurs (inner class). Pour simplifier le travail, nous ne
réalisons pour l'instant que l'addition.
class OprBtnListener implements ActionListener {
    public void actionPerformed(ActionEvent evt) {
        tfDisplay.setText("");
        previousOpr = currentOpr; // save
        currentOpr = evt.getActionCommand();

        if (currentOpr.equals("="))
            if (previousOpr.equals("+")) {
                currentNb = Integer.parseInt(numberInStr);
                result = previousNb+currentNb;
                tfDisplay.setText(""+result);
                currentOpr = "";
                currentNb = 0;
                numberInStr = "";
            }
        else

```

```

        System.out.println("Invalid input");
    else {
        currentNb = Integer.parseInt(numberInStr);
        previousNb = currentNb;
        numberInStr = "";
    }
}
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new Exo3();
        }
    });
}
}

```

Question A.1 : Nous vous demandons de structurer ce programme en utilisant le design pattern MVC, et le programmer à nouveau en respectant MVC.

Partie B : File FIFO ... généralité, Exception, Thread, (environ 10 points)

```

package df.file;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.LinkedList;
import java.util.List;

public class MaFile {
    private List liste;

    public MaFile() {
        liste = new LinkedList();
    }

    public void enfiler(Object element) {
        liste.add(element);
    }

    public Object getPremier() {
        return liste.get(0);
    }

    public Object defiler() {
        return liste.remove(0);
    }

    public boolean estVide(){
        return (liste.size() == 0);
    }

    public String toString() {
        StringBuffer result = new StringBuffer("");
        int max = liste.size();
        for ( int i=0; i < max; i++)
            result.append(liste.get(i).toString()+"\n");
        return result.toString();
    }

    public void sauvegarder(String nomFichier) throws IOException {
        File fichierPersistant = new File(nomFichier);
        if ((fichierPersistant.exists() && fichierPersistant.isFile())

```

```

        && fichierPersistant.canWrite())
    || fichierPersistant.createNewFile()) {
    ObjectOutputStream flotEcriture = new ObjectOutputStream(new
        FileOutputStream(fichierPersistant));
    flotEcriture.writeObject(this);
    flotEcriture.close();
} else
    throw new IOException ("impossible de Sauvegarder");
}

public void restaurer(String nomFichier) throws IOException {
    File fichierPersistant = new File(nomFichier);
    try {
        if (fichierPersistant.exists() && fichierPersistant.isFile()
            && fichierPersistant.canRead()) {
            ObjectInputStream flotLecture = new ObjectInputStream(new
                FileInputStream(fichierPersistant));
            MaFile monExFile = (MaFile) flotLecture.readObject();
            this.liste = monExFile.liste;
            flotLecture.close();
        } else
            throw new IOException ("impossible de Restaurer");
    } catch (ClassNotFoundException cnfe) {
        throw new IOException ("impossible de Restaurer");
    }
}
}
}

```

Et l'interface graphique la classe ManipulationFile :

Ci-contre une exécution

```

package df.file;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import javax.swing.Box;
import javax.swing.BoxLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextArea;
import javax.swing.JTextField;

public class ManipulationFile extends JFrame {
    private final String SAUVEGARDE = ".maFile";
    private MaFile maFile;
    private JLabel labelPremier;
    private JTextField nouveauMot;
    private JTextArea resultat;
    private JLabel etatSauvegarde;

    public ManipulationFile() {
        super();
        maFile = new MaFile();
        Box boiteTout = new Box(BoxLayout.Y_AXIS);
        Box boitePremier = new Box(BoxLayout.X_AXIS);
        JButton boutonPremier = new JButton("Premier");
        boitePremier.add(boutonPremier);
        labelPremier = new JLabel("          ");
        boitePremier.add(labelPremier);
        boutonPremier.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                Object premier = maFile.getPremier();
                labelPremier.setText((String)premier);
            }
        });
    }
}

```

```

boiteTout.add(boitePremier);
JButton boutonDefiler = new JButton("Defiler");
boiteTout.add(boutonDefiler);
boutonDepiler.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Object premier = maFile.defiler();
        resultat.setText(maFile.toString());
    }
});
Box boiteEmfiler = new Box(BoxLayout.X_AXIS);
JButton boutonEmfiler = new JButton("Emfiler");
boiteEmfiler.add(boutonEmfiler);
nouveauMot = new JTextField("      ");
boiteEmfiler.add(nouveauMot);
boutonEmfiler.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        String motSansBlanc = nouveauMot.getText().trim();
        if (motSansBlanc.length() > 0) {
            maFile.enfiler(motSansBlanc);
            resultat.setText(maFile.toString());
            nouveauMot.setText("");
        } else
            nouveauMot.setText("erreur nouveau mot vide ");
    }
});
boiteTout.add(boiteEmfiler);
resultat = new JTextArea(10, 10);
resultat.setEditable(false);
boiteTout.add(resultat);
Box boiteSauverRestaurer = new Box(BoxLayout.X_AXIS);
JButton boutonSauver = new JButton("Sauvegarder");
boutonSauver.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Sauvegarder sauver = new Sauvegarder(SAUVEGARDE);
        sauver.go();
    }
});
boiteSauverRestaurer.add(boutonSauver);
JButton boutonRestaurer = new JButton("Restaurer");
boutonRestaurer.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        Restaurer restaurer = new Restaurer(SAUVEGARDE);
        restaurer.go();
    }
});
boiteSauverRestaurer.add(boutonRestaurer);
boiteTout.add(boiteSauverRestaurer);
etatSauvegarde = new JLabel("      ");
boiteTout.add(etatSauvegarde);
this.setContentPane(boiteTout);
this.setDefaultCloseOperation(getDefaultCloseOperation());
this.pack();
this.setVisible(true);
}

class Sauvegarder {
    private String nomFichier;
    private boolean etat ;
    public Sauvegarder(String nomFichier) {
        this.nomFichier = nomFichier;
    }
    public void go() {
        etat = true;
        try {
            maFile.sauvegarder(nomFichier);
        } catch (IOException ioe) {
            etat = false;
        }
        if (etat)
            etatSauvegarde.setText("sauvegarde réussie");
    }
}

```

```

        else
            etatSauvegarde.setText("erreur de sauvegarde ");
    }
}

class Restaurer {
    private String nomFichier;
    private boolean etat ;
    public Restaurer(String nomFichier) {
        this.nomFichier = nomFichier;
    }
    public void go() {
        etat = true;
        try {
            maFile.restaurer(nomFichier);
        } catch (IOException ioe) {
            etat = false;
        }
        if (etat) {
            resultat.setText(maFile.toString());
            etatSauvegarde.setText("restauration réussie");
        } else
            etatSauvegarde.setText("erreur de restauration ");
    }
}

public static void main(String[] args) {
    javax.swing.SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new ManipulationFile();
        }
    });
}
}

```

Question A :

J'essaye le bouton Sauvegarder Catastrophe !
 Que faut-il ajouter et à quel endroit pour corriger cela ?
 Ci-dessous la trace de l'erreur.

```

$ java df.file.ManipulationFile java.io.NotSerializableException: df.file.MaFile
at java.io.ObjectOutputStream.writeObject0(ObjectOutputStream.java:1184)
at java.io.ObjectOutputStream.writeObject(ObjectOutputStream.java:348)
at df.file.MaFile.sauvegarder(MaFile.java:49)
at df.file.ManipulationFile$Sauvegarder.go(ManipulationFile.java:114)
at df.file.ManipulationFile$4.actionPerformed(ManipulationFile.java:82)
at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:2022)
at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2348)
at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:402)
at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:259)
at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:252)
at java.awt.Component.processMouseEvent(Component.java:6533)
at javax.swing.JComponent.processMouseEvent(JComponent.java:3324)
at java.awt.Component.processEvent(Component.java:6298)
at java.awt.Container.processEvent(Container.java:2236)
at java.awt.Component.dispatchEventImpl(Component.java:4889)
at java.awt.Container.dispatchEventImpl(Container.java:2294)
at java.awt.Component.dispatchEvent(Component.java:4711)
at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4888)
at java.awt.LightweightDispatcher.processMouseEvent(Container.java:4525)
at java.awt.LightweightDispatcher.dispatchEvent(Container.java:4466)
at java.awt.Container.dispatchEventImpl(Container.java:2280)
at java.awt.Window.dispatchEventImpl(Window.java:2746)
at java.awt.Component.dispatchEvent(Component.java:4711)
at java.awt.EventQueue.dispatchEventImpl(EventQueue.java:758)
at java.awt.EventQueue.access$500(EventQueue.java:97)
at java.awt.EventQueue$3.run(EventQueue.java:709)
at java.awt.EventQueue$3.run(EventQueue.java:703)
at java.security.AccessController.doPrivileged(Native Method)
at

```

```

java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:76)
    at
java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:86)
    at java.awt.EventQueue$4.run(EventQueue.java:731)
    at java.awt.EventQueue$4.run(EventQueue.java:729)
    at java.security.AccessController.doPrivileged(Native Method)
    at
java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:76)
    at java.awt.EventQueue.dispatchEvent(EventQueue.java:728)
    at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:201)
    at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:116)
    at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:105)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
    at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)

```

Question B :

Nous voudrions que la File ne comporte que des éléments de même type.

Que faut-il modifier dans le fichier MaFile pour que cette classe soit générique ?

Question C :

A force d'appuyer sur le bouton "défiler", l'erreur ci-dessous se produit.

La prise en compte des Exceptions, lorsque la file est vide, n'est pas programmée.

```

$ java df.file.ManipulationFileException in thread "AWT-EventQueue-0" java.lang.IndexOutOfBoundsException:
Index: 0, Size: 0
    at java.util.LinkedList.checkElementIndex(LinkedList.java:555)
    at java.util.LinkedList.remove(LinkedList.java:525)
    at df.file.MaFile.defiler(MaFile.java:25)
    at df.file.ManipulationFile$2.actionPerformed(ManipulationFile.java:46)
    at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:2022)
    at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2348)
    at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:402)
    at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:259)
    at javax.swing.plaf.basic.BasicButtonListener.mouseReleased(BasicButtonListener.java:252)
    at java.awt.Component.processMouseEvent(Component.java:6533)
    at javax.swing.JComponent.processMouseEvent(JComponent.java:3324)
    at java.awt.Component.processEvent(Component.java:6298)
    at java.awt.Container.processEvent(Container.java:2236)
    at java.awt.Component.dispatchEventImpl(Component.java:4889)
    at java.awt.Container.dispatchEventImpl(Container.java:2294)
    at java.awt.Component.dispatchEvent(Component.java:4711)
    at java.awt.LightweightDispatcher.retargetMouseEvent(Container.java:4888)
    at java.awt.LightweightDispatcher.processMouseEvent(Container.java:4525)
    at java.awt.LightweightDispatcher.dispatchEvent(Container.java:4466)
    at java.awt.Container.dispatchEventImpl(Container.java:2280)
    at java.awt.Window.dispatchEventImpl(Window.java:2746)
    at java.awt.Component.dispatchEvent(Component.java:4711)
    at java.awt.EventQueue.dispatchEventImpl(EventQueue.java:758)
    at java.awt.EventQueue.access$500(EventQueue.java:97)
    at java.awt.EventQueue$3.run(EventQueue.java:709)
    at java.awt.EventQueue$3.run(EventQueue.java:703)
    at java.security.AccessController.doPrivileged(Native Method)
    at
java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:76)
    at
java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:86)
    at java.awt.EventQueue$4.run(EventQueue.java:731)
    at java.awt.EventQueue$4.run(EventQueue.java:729)
    at java.security.AccessController.doPrivileged(Native Method)
    at
java.security.ProtectionDomain$JavaSecurityAccessImpl.doIntersectionPrivilege(ProtectionDomain.java:76)
    at java.awt.EventQueue.dispatchEvent(EventQueue.java:728)
    at java.awt.EventDispatchThread.pumpOneEventForFilters(EventDispatchThread.java:201)
    at java.awt.EventDispatchThread.pumpEventsForFilter(EventDispatchThread.java:116)
    at java.awt.EventDispatchThread.pumpEventsForHierarchy(EventDispatchThread.java:105)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:101)
    at java.awt.EventDispatchThread.pumpEvents(EventDispatchThread.java:93)
    at java.awt.EventDispatchThread.run(EventDispatchThread.java:82)

```

Re-programmez les méthodes `getPremier()` et `defiler()` pour qu'elles génèrent, quand nécessaire, une `java.lang.UnsupportedOperationException`.

Re-programmez, en conséquence, les méthodes des classes internes anonymes écouteurs des boutons "Premier" et "Défiler".

Question D :

La programmation événementielle des boutons "Sauvegarder" et "Restaurer" est minable : les lectures et écritures dans un fichier sont des opérations longues, or les opérations longues ne doivent pas monopoliser le thread graphique.

Re-programmez, en conséquence, uniquement "Sauvegarder", car le traitement de "Restaurer" est identique. Attention à ce que les opérations "graphiques" soient bien traitées par le thread graphique.