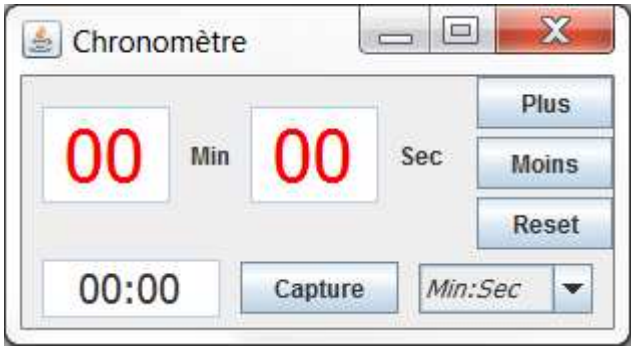
TP 3 : Actions associées à une interface

On veut ajouter des actions à l'interface du chronomètre du TP précédent.



Remarque : Pour les 2 premières questions on ne tiendra pas compte du mode de chronométrage choisi et on considèrera qu'il s'agit toujours du mode minutes et secondes (on modifiera cela plus tard).

1°) Ajouter l'action associée au bouton "Plus" qui fait avancer les secondes d'une unité. Bien entendu, quand les secondes passeront de 59 à 0 les minutes avanceront de 1 et si elles atteignent 59 elles repasseront à 0.
Pour convertir un entier en chaîne on utilisera String.valueOf
Pour convertir une chaîne en entier on utilisera Integer.parseInt qui peut lever une exception de classe NumberFormatException

2°) Ajouter l'action associée au bouton "Moins" qui fait reculer les secondes d'une unité. Bien entendu, quand les secondes passeront de 0 à 59 les minutes reculeront de 1 et si elles atteignent 0 elles repasseront à 59.

3°) Ajouter l'action associée au bouton "Capture" pour que le temps actuellement chronométré soit affiché dans cette zone, les 2 valeurs étant séparées par un ':'

4°) Ajouter l'action associée au bouton "Reset" pour que les 2 valeurs du temps chronométré et du temps intermédiaire reviennent à 0 (pour le temps intermédiaire c'est 0:0)

5°) Ajouter l'action associée à la liste déroulante de façon à modifier les étiquettes qui seront soit Min et Sec soit Sec et 1/100. Par ailleurs un changement de mode doit remettre les 2 valeurs du temps chronométré à 0 mais ne modifie pas le temps intermédiaire.

6°) Modifier les actions des 1°) et 2°) de façon à ce que l'avance ou le recul du temps affiché soit conforme au choix de mesure (entre 0 et 59 ou entre 0 et 100).

7°) Modifier la façon d'afficher les valeurs pour que les 2 zones d'affichage du temps chronométré soient toujours sur 2 chiffres (afficher 05 plutôt que 5). De même le temps intermédiaire doit aussi être toujours de la forme xx:xx (05:00 plutôt que 5:0)

SOLUTION

```
package tp3;

import javax.swing.JFrame;

public class Chronometre extends JFrame {
	private JTextField minutes;
	private JTextField secondes;
	private JTextField tempsIntermediaire;
	private JButton start, stop, capture, reset;
	private JComboBox<String> mode;
	private JLabel unite1, unite2;
	private boolean minsec;

	public Chronometre() {
		super("Chronomètre");
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		minsec=true;
		getContentPane().setLayout(new BorderLayout(0, 0));

		JPanel panel = new JPanel();
		getContentPane().add(panel, BorderLayout.CENTER);
		FlowLayout fl_panel = new FlowLayout(FlowLayout.CENTER, 10, 15);
		panel.setLayout(fl_panel);

		minutes = new JTextField();
		minutes.setToolTipText("Minutes ou secondes");
		minutes.setHorizontalAlignment(SwingConstants.CENTER);
		minutes.setText("00");
		minutes.setForeground(Color.RED);
		minutes.setBackground(Color.WHITE);
		minutes.setFont(new Font("Tahoma", Font.PLAIN, 36));
		minutes.setEditable(false);
		panel.add(minutes);
		minutes.setColumns(2);

		unite1 = new JLabel("Min");
		panel.add(unite1);

		secondes = new JTextField();
		secondes.setToolTipText("Secondes ou 1/100 de seconde");
		secondes.setText("00");
		secondes.setHorizontalAlignment(SwingConstants.CENTER);
		secondes.setForeground(Color.RED);
		secondes.setBackground(Color.WHITE);
		secondes.setEditable(false);
		secondes.setFont(new Font("Tahoma", Font.PLAIN, 36));
		panel.add(secondes);
		secondes.setColumns(2);

		unite2 = new JLabel("Sec  ");
		panel.add(unite2);

		JPanel panel_1 = new JPanel();
		getContentPane().add(panel_1, BorderLayout.EAST);
		panel_1.setLayout(new GridLayout(3, 1, 1, 4));

		start = new JButton("Plus");
		start.addActionListener(new ActionPlus());
```

```
		start.setToolTipText("Avancer");
		panel_1.add(start);

		stop = new JButton("Moins");
		stop.addActionListener(new ActionMoins());
		stop.setToolTipText("Reculer");
		panel_1.add(stop);

		reset = new JButton("Reset");
		reset.addActionListener(new ActionReset());
		reset.setToolTipText("Remise \u00E0 z\u00E9ro du chronom\u00E8tre");
		panel_1.add(reset);

		JPanel panel_2 = new JPanel();
		getContentPane().add(panel_2, BorderLayout.SOUTH);
		FlowLayout fl_panel_2 = new FlowLayout(FlowLayout.CENTER, 10, 5);
		panel_2.setLayout(fl_panel_2);

		tempsIntermediaire = new JTextField();
		tempsIntermediaire.setToolTipText("Temps interm\u00E9diaire");
		tempsIntermediaire.setText("00:00");
		tempsIntermediaire.setHorizontalAlignment(SwingConstants.CENTER);
		tempsIntermediaire.setBackground(Color.WHITE);
		tempsIntermediaire.setEditable(false);
		tempsIntermediaire.setFont(new Font("Tahoma", Font.PLAIN, 20));
		panel_2.add(tempsIntermediaire);
		tempsIntermediaire.setColumns(5);

		capture = new JButton("Capture");
		capture.addActionListener(new ActionCapture());
		capture.setToolTipText("Capture du temps interm\u00E9diaire");
		panel_2.add(capture);

		mode = new JComboBox<String>();
		mode.addActionListener(new ActionChoix());
		mode.setToolTipText("Choix du mode (min:sec ou sec:1/100)");
		mode.setFont(new Font("Tahoma", Font.ITALIC, 13));
		mode.addItem("Min:Sec");
		mode.addItem("Sec:1/100");
		panel_2.add(mode);

		pack();
		setVisible(true);
	}

	private class ActionPlus implements ActionListener {
		public void actionPerformed(ActionEvent e) {
			int limite;
			if (minsec) limite = 60; else limite = 100;
			int sec;
			try { sec=Integer.parseInt(secondes.getText()); }
			catch (NumberFormatException nfem) {sec=-1;}
			sec++;
			if (sec==limite) {
				sec=0;
				int m;
				try { m=Integer.parseInt(minutes.getText()); }
				catch (NumberFormatException nfem) {m=-1;}
				m++;
```

```
                        if (m==limite) {
                                m=0;
                        }
                        if (m<10) minutes.setText("0"+String.valueOf(m));
                        else minutes.setText(String.valueOf(m));
                }
                if (sec<10) secondes.setText("0"+String.valueOf(sec));
                else secondes.setText(String.valueOf(sec));
        }
}

private class ActionMoins implements ActionListener {
        public void actionPerformed(ActionEvent e) {
                int limite;
                if (minsec) limite = 60; else limite = 100;
                int sec;
                try { sec=Integer.parseInt(secondes.getText()); }
                catch (NumberFormatException nfem) {sec=1;}
                sec--;
                if (sec==-1) {
                        sec=limite-1;
                        int m;
                        try { m=Integer.parseInt(minutes.getText()); }
                        catch (NumberFormatException nfem) {m=1;}
                        m--;
                        if (m==-1) {
                                m=limite-1;
                        }
                        minutes.setText(String.valueOf(m));
                }
                secondes.setText(String.valueOf(sec));
        }
}

private class ActionReset implements ActionListener {
        public void actionPerformed(ActionEvent e) {
                minutes.setText("00");
                secondes.setText("00");
                tempsIntermediaire.setText("00:00");
        }
}

private class ActionCapture implements ActionListener {
        public void actionPerformed(ActionEvent e) {
                String v1,v2;
                v1 = minutes.getText();
                v2 = secondes.getText();
                tempsIntermediaire.setText(v1+":"+v2);
        }
}

private class ActionChoix implements ActionListener  {
        public void actionPerformed(ActionEvent e) {
                switch (mode.getSelectedIndex()) {
                case 0: unite1.setText("Min");
                        unite2.setText("Sec");
                        minsec=true;
                        break;
                case 1: unite1.setText("Sec");
```

```
                unite2.setText("1/100");
                minsec=false;
                break;
                }
                minutes.setText("00");
                secondes.setText("00");
          }
    }

    /**
     * @param args
     */
    public static void main(String[] args) {
          // TODO Auto-generated method stub
          new Chronometre();
    }

}
```