

Solution de l'examen Septembre 2000

**//Question 1**

```
import java.util.Vector;

public class Pile {
    Vector laPile;

    public Pile() {
        laPile=new Vector();
    }

    public void empiler(int x) {
        laPile.addElement(new Integer(x));
    }

    public int depiler() {
        int x = ((Integer) laPile.lastElement()).intValue();
        laPile.removeElement(laPile.lastElement());
        return( x)
    }

    public int sommet() {
        return ((Integer) laPile.lastElement()).intValue();
    }
}
```

**//Question B**

//Dans le classe Pile

```
public int depiler() throws PileVideErreur {
    if (laPile.isEmpty()) throw(new PileVideErreur());
    int x = ((Integer) laPile.lastElement()).intValue();
    laPile.removeElement(laPile.lastElement());
    return x;
}

public int sommet() throws PileVideErreur {
    if (laPile.isEmpty()) throw(new PileVideErreur());
    return ((Integer) laPile.lastElement()).intValue();
}
```

//On crée un nouvelle classe pour l'exception

```
public class PileVideErreur extends Exception {
    public PileVideErreur() { }
}
```

## //Deuxième partie

//Question C et D

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

public class Exam extends Frame implements ActionListener {
    Button chiffre[],enter,plus,moins,diviser,multi;
    Label resultat;
    Pile maPile;
    boolean endSaisie; //true si la valeur saisie a été validé par la touche Enter

    public Exam() {
        int i;
        super("Calculatrice");
        addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e){
                System.exit(0);
            }
        });

        maPile = new Pile();
        endSaisie = true;
        chiffre=new Button[10];
        for(i=0; i<10 ; i++) {
            chiffre[i] = new Button(String.valueOf(i));
            chiffre[i].addActionListener(this);
        }
        enter = new Button("Enter");
        plus = new Button("+");
        moins = new Button("-");
        multi = new Button("*");
        diviser = new Button("/");
        resultat=new Label("",Label.RIGHT);
        setLayout(new BorderLayout());
        add("North",resultat);
        Panel boutons=new Panel();
        boutons.setLayout(new GridLayout(4,4));
        for(i=1; i<10; i++) boutons.add(chiffre[i]);
        boutons.add(chiffre[0]);
        boutons.add(zero);
        boutons.add(new Panel());
        boutons.add(enter);
        boutons.add(diviser);
        enter.addActionListener(this);
        plus.addActionListener(this);
        moins.addActionListener(this);
        multi.addActionListener(this);
        diviser.addActionListener(this);
    }
}
```

```

        add("Center",boutons);
        pack();
    }

    public void actionPerformed(ActionEvent e) {
        int i;
        Object source = e.getSource();
        for(i=0; i<10; i++){
            if(source == chiffre[i]){
                if(endSaisie) resultat.setText(String.valueOf(i));
                else resultat.setText(resultat.getText()+ i );
                endSaisie = false;
                return;
            }
        }
        if (source == plus || source == moins || source == multi || source == diviser) {
            try {
                if(! resultat.getText().equals(""))
maPile.empiler(Integer.parseInt(resultat.getText()));
                int a=maPile.depiler();
                int b=maPile.depiler();
                int c;
                if (source == plus) c = a+b;
                else if (source == moins) c=b-a;
                else if (source == multi) c=b*a;
                else c= b/a;
                maPile.empiler(c);
                resultat.setText(String.valueOf(c));
                endSaisie = true;
            }
            catch(PileVideErreur erreur) {
                System.out.println("Erreur: la pile est vide");
            }
        }
        else if (source == enter) {
            maPile.empiler(Integer.parseInt(resultat.getText()));
            endSaisie = true;
        }
    }

    public static void main(String[] args) {
        Exam e = new Exam();
        e.show();
    }
}

```

**//Question E**

```

public class Exam extends Frame implements ActionListener {
    //... ancien attribut ...
    Button sto,rcl ;
    int memory;

    public Exam() {
        //... comme avant ...
        memory = 0;
        sto=new Button("STO");
        rcl=new Button("RCL");
        sto.addActionListener(this);
        rcl.addActionListener(this);

        boutons.setLayout(new GridLayout(5,4));
        //... comme avant ...
        boutons.add(sto);
        boutons.add(rcl);
        //... comme avant ...
    }

    public void actionPerformed(ActionEvent e) {
        //... comme avant ...

        else if ( source == sto) {
            try { memory=maPile.sommet(); }
            catch(PileVideErreur erreur) {
                System.out.println("Erreur: la pile est vide");
            }
        }
        else if(e.getSource() == rcl) {
            maPile.empiler(memory);
            resultat.setText(String.valueOf(memory));
            endSaisie = true;
        }
    }

    //...
}

```