Exercices de révision - Solutions

R.1: MiniLotto

```
public class MiniLotto {

    private int number1;
    private int number2;
    private int number3;

    public void generateNumbers() {
        number1 = number2 = number3 = 0;
        while (number1 == number2 || number1 == number3 || number2 == number3) {
            number1 = (int) (Math.random() * 49) + 1;
            number2 = (int) (Math.random() * 49) + 1;
            number3 = (int) (Math.random() * 49) + 1;
        }
        // for debug purposes
        System.out.println(number1 + " " + number2 + " " + number3);
    }

    public boolean contains(int number) {
        return (number == number1 || number == number2 || number == number3);
    }

    public int play(int n1, int n2, int n3) {
        if (n1 == n2 || n1 == n3 || n2 == n3) {
            System.out.println("Erreur, il faut indiquer 3 nombres différents ! ");
            return -1;
        }
        generateNumbers();
        int result = 0;
        if (contains(n1)) {
            result++;
        }
        if (contains(n2)) {
            result++;
        }
        if (contains(n3)) {
            result++;
        }
        System.out.println("Vous avez deviné correctement " + result + "
nombre(s)");
        return result;
    }
}
```

```
public class MainFrame extends javax.swing.JFrame {
    private MiniLotto miniLotto = new MiniLotto();

    public MainFrame() {
        initComponents();
    }
    private void playButtonActionPerformed(java.awt.event.ActionEvent evt) {
        int guess1 = Integer.valueOf(guess1TextField.getText());
        int guess2 = Integer.valueOf(guess2TextField.getText());
        int guess3 = Integer.valueOf(guess3TextField.getText());
        int result = miniLotto.play(guess1, guess2, guess3);
        if (result == -1) {
            outputLabel.setText("Erreur, il faut indiquer 3 nombres différents !");
        } else {
            outputLabel.setText("Vous avez deviné correctement " + result + "
nombre(s)");
        }
    }
```

R.2: Voting System

```
public class Candidate {
    private String name;
    private String firstname;
    private int votes = 0;

    public Candidate(String name, String firstname) {
        this.name = name;
        this.firstname = firstname;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getFirstname() {
        return firstname;
    }

    public void setFirstname(String firstname) {
        this.firstname = firstname;
    }

    public int getVotes() {
        return votes;
    }

    public void voteFor() {
        votes++;
    }

    @Override
    public String toString() {
        String pre = String.valueOf(votes);
        while (pre.length() < 5)
            pre = " " + pre;
        return pre+" : " + name + " " + firstname;
    }

}
```

```
public class Party {
    private String name;

    private ArrayList<Candidate> alCandidates = new ArrayList<>();

    public Party(String name) {
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public int size() {
        return alCandidates.size();
    }

    public Candidate get(int i) {
        return alCandidates.get(i);
    }

    public boolean add(Candidate e) {
        return alCandidates.add(e);
    }

    public Candidate remove(int i) {
        return alCandidates.remove(i);
    }

    public void clear() {
        alCandidates.clear();
    }

    public Object[] toArray() {
        return alCandidates.toArray();
    }

    public int indexOf(Object o) {
        return alCandidates.indexOf(o);
    }

    public int getTotalVotes()
    {
        int result = 0;

        for (int i = 0; i < alCandidates.size(); i++) {
            Candidate get = alCandidates.get(i);
            Result += get.getVotes();
        }

        return result;
    }
```

```
    public Candidate findByName(String name) {
        for (int i = 0; i < alCandidates.size(); i++) {
            Candidate get = alCandidates.get(i);
            if (name.equals(get.getName()))
                return get;
        }
        return null;
    }

    /*
    public boolean voteFor(String name)
    {
        Candidate candi = findByName(name);

        if (candi==null) {
            return false;
        } else {
            candi.voteFor();
            return true;
        }
    }*/

    public boolean voteFor(int index) {
        if (index<0 || index>=alCandidates.size()) {
            return false;
        } else {
            alCandidates.get(index).voteFor();
            return true;
        }
    }

    public void sortByVotes()  {
        for (int i=0; i<alCandidates.size()-1; i++) {
            int posmax = i;
            for (int j=i+1; j<alCandidates.size(); j++) {
                if (alCandidates.get(j).getVotes()>alCandidates.get(posmax).getVotes())
                    posmax=j;
            }
            if (posmax!=i){
                Candidate tmp = alCandidates.get(i);
                alCandidates.set(i, alCandidates.get(posmax));
                alCandidates.set(posmax, tmp);
            }
        }
    }

}
```

```
public class MainFrame extends javax.swing.JFrame {

    private Party party;

    public MainFrame() {
        initComponents();

        party = initParty("dp");
        updateView();
    }

    public void updateView() {
        partyLabel.setText(party.getName());
        candidatesLabel.setText(party.size()+"");
        votesLabel.setText(party.getTotalVotes()+"");

        party.sortByVotes();
        candidatesList.setListData(party.toArray());
    }

    private Party initParty(String name) {
        ...
    }

    private void voteButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // E
        int index = candidatesList.getSelectedIndex();
        if(index<0) return;

        // T
        party.get(index).voteFor();

        // S
        updateView();
    }

    private void searchTextFieldActionPerformed(java.awt.event.ActionEvent evt) {
        searchButtonActionPerformed(evt);
    }

    private void searchButtonActionPerformed(java.awt.event.ActionEvent evt) {
        // E
        String search = searchTextField.getText();

        // T
        Candidate c = party.findByName(search);
        int index = party.indexOf(c);

        // S
        candidatesList.setSelectedIndex(index);
        if(index==-1) candidatesList.clearSelection();
    }
```