Licence Informatique 1ère année
Module Info2b – Interfaces visuelles

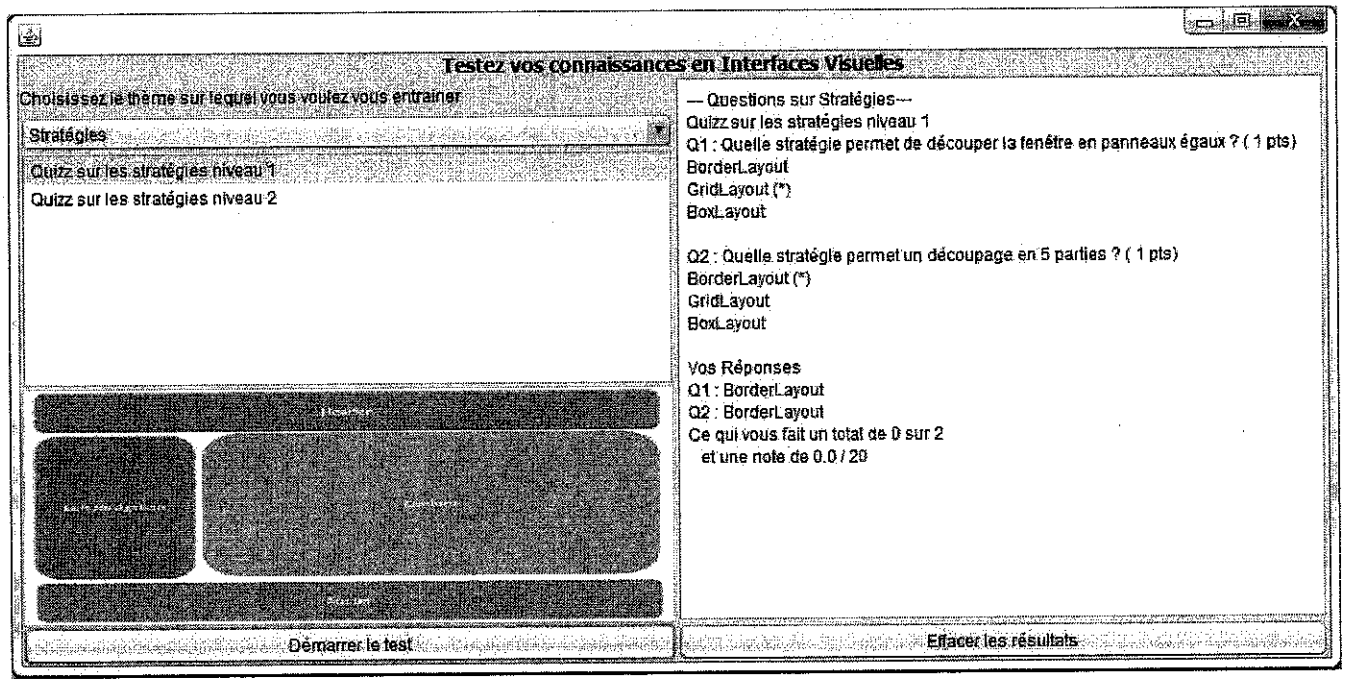
Examen session 2 – Durée 2h - Vendredi 22 juin 2018
Documents autorisés (polycopiés de CM, TD et TP)

Un enseignant de l'université de Bourgogne a décidé de proposer à ses étudiants de parfaire leurs connaissances en s'entrainant sur des questionnaires en ligne. Chaque questionnaire traite d'un thème et prend la forme d'un QCM. Tous les questionnaires sont construits sur le même modèle : une question, trois propositions et une seule réponse correcte.

Une question est caractérisée par un intitulé de type String, un ensemble de trois propositions stockées sous forme de chaînes de caractères, la position de la bonne réponse parmi les propositions (entier entre 0 et 2) et le nombre de points que rapporte la question. Une réponse, quant à elle, est caractérisée par la question à laquelle elle se rapporte ainsi que la position de la proposition choisie comme étant la bonne réponse.
L'ensemble des questions est géré par une liste « List<Question> » dans la classe « Questionnaire ». Cette classe possède aussi comme attributs le titre du questionnaire et le thème abordé par les questions (chaînes de caractères).
L'ensemble des réponses est géré par un tableau dans la classe « LesReponses ». Le nombre de réponses fournies est identique au nombre de questions du questionnaire auquel sont apportées les réponses.

L'annexe 1 propose un extrait de la classe « Question » qui décrit une question et le contenu de la classe « Reponse » qui décrit une réponse à une question. L'annexe 2 fournit un extrait de la classe « Questionnaire » et de la classe « LesReponses ».

Un 1er prototype a été développé par un stagiaire. L'interface de l'application principale « Examen2S_2018 » est de la forme suivante :



Cette interface comporte entre autre :

- un composant (JComboBox) pour choisir un thème parmi les différents thèmes abordés dans les questionnaires
- un composant (JList) pour afficher les titres des questionnaires du thème sélectionné et choisir celui à lancer
- un composant (JPanel) pour afficher l'image du thème choisi
- un composant (JButton) pour ouvrir la boîte de dialogue « QuestionDlg » afin de répondre au questionnaire sélectionné
- une zone (JTextArea) pour visualiser les questions du questionnaire choisi, les réponses fournies ainsi que le nombre de points acquis et la note sur 20
- un composant (JButton) pour effacer la zone de Texte « Edition »

L'utilisateur choisit un thème dans la liste déroulante, ce qui provoque la mise à jour de la liste avec les questionnaires portant sur le thème sélectionné et de l'image. L'utilisateur sélectionne le titre du questionnaire sur lequel il veut s'entrainer et clique sur le bouton « Démarrer le test » pour le lancer. Une nouvelle fenêtre s'ouvre pour y saisir les réponses. Lors du retour dans l'application principale, les résultats du test sont ajoutés dans la partie droite de la fenêtre. Il peut alors les effacer en cliquant sur le bouton prévu à cet effet.

Exercice 1 (5 pts) - *Classes « Question », « Reponse » et « LesReponses »*

En utilisant le code de la classe « Question » fourni en annexe 1,

a. (1 pt) Rédigez le code java du constructeur « *public Question(String intitu, int noC, int nbP)* » de la classe « Question » qui initialise tous les attributs.

En utilisant le code de la classe « Reponse » fourni en annexe 1,

b. (1 pt) Expliquez le contenu de la chaîne de caractères renvoyée par la méthode « toString() » de la classe « Reponse ».

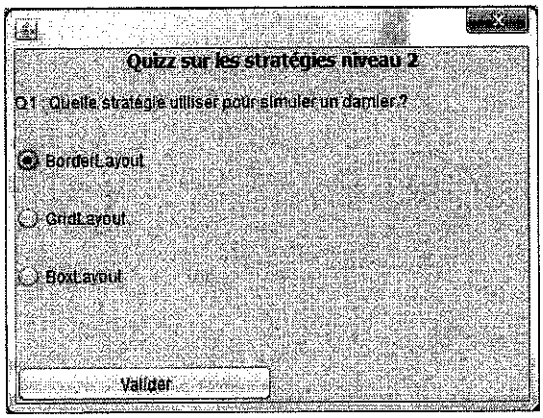
En utilisant le code de la classe « LesReponses » fourni en annexe 2,

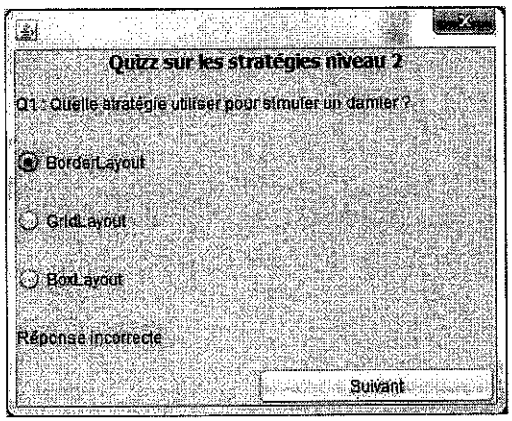
c. (1 pt) Expliquez le résultat renvoyé par la méthode « *getNbTotalPts()* » de la classe « LesReponses » par une phrase sans commenter chaque ligne de code.

d. (2 pts) Rédigez le code java de la méthode « getNbPoints() » de la classe « LesReponses » qui retourne le nombre de points obtenus en ne comptabilisant que les bonnes réponses. Pour mémoire, seul le numéro de la proposition est stocké dans la réponse.

Exercice 2 (5,5 pts) - *Boîte de dialogue « QuestionDlg »*

Le clic sur le bouton « Démarrer le test » de l'application principale ouvre une boîte de dialogue « QuestionDlg ». Cette dernière permet de proposer de répondre aux différentes questions qui composent le questionnaire préalablement choisi dans la fenêtre principale. L'interface de cette boîte de dialogue prend plusieurs apparences : elle propose d'abord de cocher la proposition considérée comme bonne puis de la valider, à la suite de la validation, le résultat est affiché (Réponse correcte / Réponse incorrecte) et un bouton permet de passer à la question suivante.





La boîte de dialogue « QuestionDlg » est composée :

- en haut d'un composant de type JLabel nommé « Titre »
- en bas d'un JPanel contenant deux boutons de type JButton nommés respectivement « Valider » et « Suivant »
- au centre d'un JPanel organisé en Grid Layout avec 5 lignes contenant successivement un composant de type JLabel « Intitule » pour placer l'intitulé de la question précédé de son numéro d'ordre dans le questionnaire, trois composants de type JRadioButton (nommés respectivement « RB1 », « RB2 », « RB3 ») pour les propositions et un dernier composant de type JLabel nommé « Resultat » pour afficher si la réponse est ou non correcte.

En utilisant l'annexe 4 qui décrit partiellement la classe « QuestionDlg »,

a. (1 pt) Expliquez les attributs de la classe « QuestionDlg » en indiquant leur utilité dans la gestion de la fenêtre.

b. (1 pt) Justifiez la raison de la présence, dans l'entête du constructeur de la classe « QuestionDlg », des différents paramètres.

c. (1,5 pts) Expliquez, ligne par ligne, par des phrases le code du constructeur de la classe « QuestionDlg ».

d. (2 pts) Rédigez le code java du gestionnaire de clic sur le bouton « Valider ». Ce gestionnaire permet de :
   - vérifier si la case cochée est la bonne
   - afficher le message adéquat (Réponse correcte / Réponse incorrecte) dans le composant « Resultat » (JLabel)
   - rendre le bouton « Suivant » visible et le bouton « Valider » invisible
   - modifier la réponse courante avec l'indice de la valeur choisie par l'utilisateur.

Exercice 3 ***(9,5 pts)*** *- Application principale*

En utilisant le code de la méthode « initComponents() » de la classe « Examen2S_2018 » fournie en annexe 3,

a. (2,5 pts) Représentez cette fenêtre de type JFrame sous forme d'une arborescence avec les types de composant, leur nom et si besoin leur valeur.

L'application principale comporte les attributs et le constructeur suivants :

```
// tableau des thèmes disponibles
        private String tabThemes[ ]={"Stratégies", "Containers", "Composants", };
// tableau des noms des fichiers (de type String) des images illustrant les thèmes
        private String tabFichImgThemes[ ]={"Strategies.png","Containers.png","Composants.png"};
// tableau des images (de type Image) des thèmes
        private Image tabImgThemes[ ];
// tableau des questionnaires disponibles
        private Questionnaire quizz[ ];
// image du thème affichée
        private Image img;

    public Examen2S_2018() {
      initComponents();
      DefaultListModel mod= new DefaultListModel();
      ListeQuizz.setModel(mod);
      Edition.setText("");
      initQuizz();     // initialise le tableau des quizz avec des questionnaires
      initCBThemes(); // initialise la liste déroulante avec les noms des thèmes
      initTabImageThemes(); // initialise le tableau des images associées aux thèmes
    }
```

Elle comporte également deux méthodes de recherche

- *public List<Questionnaire> rechTheme(String theme)* qui retourne la liste des questionnaires pour le thème passé en paramètre
- *public Questionnaire rechTitre(String titre)* qui retourne le questionnaire dont le titre en passé en paramètre

b. (1,5 pts) Rédigez le code de la méthode « initCBThemes() » qui initialise le composant JComboBox « CBThemes » avec les différents thèmes stockés dans la collection « tabThemes ».

c. (2,5 pts) Rédigez le code du gestionnaire lié à la sélection du thème dans la JComboBox nommée « CBThemes »

private void CBThemesActionPerformed(java.awt.event.ActionEvent evt)
qui récupère l'index du thème sélectionné, recherche les questionnaires portant sur ce thème, ajoute leurs titres dans le composant de type JList nommé « ListeQuizz » après avoir vidé cette liste avec méthode clear() initialise l'image avec celle du thème à afficher et demande le rafraîchissement de la fenêtre.

d. (1,5 pts) Donnez les deux instructions qui seraient nécessaires dans le code du gestionnaire de clic sur le bouton « Demarrer » pour créer, allouer et rendre visible la boîte de dialogue « QuestionDlg » si en considérant que le questionnaire choisi est référencé par la variable « leQ » (de type Questionnaire).

e. (1,5 pts) L'application principale comporte également la surcharge de la méthode « paint » comme indiqué ci-dessous, expliquer en une phrase (sans commenter le code ligne à ligne) ce que fait cette méthode et pourquoi il est nécessaire de la surcharger.

```
public void paint(Graphics g)
  { super.paint(g);
    if (img != null)
    { Graphics gg = PanImage.getGraphics();
      gg.drawImage(img, 0, 0, PanImage.getWidth(), PanImage.getHeight(), this);
    }
  }
```

ANNEXE 1 : Extrait de la classe « Question » et contenu de la classe « Reponse »

```
public class Question {
  private String intitule;
  private String proposition[];
  private int nbProp;
  private int noCorrecte;
  private int nbPoints;

  public Question(String intitu,int noC, int nbP)  { // à compléter }

  public String getIntitule() {  return intitule;  }
  public int getNoCorrecte() {  return noCorrecte;  }
  public int getNbPoints() {  return nbPoints;  }
  public String getProposition(int i)   { if (0<=i && i<3) return proposition[i];   else return null;  }

  public boolean ajoutProposition(String p)
  { boolean res=true;
   if (nbProp<3)
       proposition[nbProp++]=p;
   else res=false;
   return res;
  }

  public String toString() {
     String s= intitule + " ( "+nbPoints+" pts)\n";
     for(int i=0; i<nbProp; i++)
     {  s+= proposition[i];
        if (i==noCorrecte) s+= " (*)";
        s+="\n";
     }
     return s;
  }
}



public class Reponse {
  private Question quest;
  private int numReponse;

  public Reponse(Question q)
  { quest=q;     numReponse=0;  }

  public Question getQuestion() {  return quest;  }
  public int getNumReponse() {  return numReponse;  }
  public void setNumReponse(int rep) {  numReponse=rep;  }

  public String toString() {   String p=quest.getProposition(numReponse);   return p ;}  // à expliquer
  }
```

ANNEXE 2 : Extrait de la classe « Questionnaire » et de la classe « LesReponses »

```
public class Questionnaire {

  private List<Question> lstQ;
  private String titre;
  private String theme;

  public Questionnaire(String t, String th)  { titre=t;   theme=th;   lstQ = new ArrayList<Question>(); }

  public int getNbQuestions(){ return lstQ.size();}
  public String getTitre() {  return titre;  }
  public String getTheme() {  return theme;  }

  public Question getQuestion(int i) {  if (0<=i && i<lstQ.size()) return lstQ.get(i);     else return null;  }

  public void ajoutQuestion(Question q)   { lstQ.add(q); }

  public double getNbPointsTotal()
  { int res=0;
    for(int i=0; i<lstQ.size();i++)      res+=lstQ.get(i).getNbPoints();
    return res;  }

  public String toString()
  { String s="--- Questions sur "+theme+"---\n"+titre;
    for(int i=0; i< lstQ.size();i++)    s+="\nQ"+(i+1)+" : "+ this.getQuestion(i);
    return s;  }
}

public class LesReponses {
  private Reponse tabR[];

  public LesReponses(int taille)  {    tabR = new Reponse[taille];  }

  public boolean ajoutReponse(Reponse r, int n)
  { boolean res=true;
    if (0<=n && n<tabR.length && tabR[n]==null)        tabR[n]=r;    else res=false;
    return res;  }

  public Reponse getReponse(int i)
  { if (0<=i && i<tabR.length)        return tabR[i];    else return null;        }

  public int getNbReponses() { return tabR.length;}

  public double getNbPoints()  { // à compléter  }

  public double getNbTotalPts()
  {  double res=0;
     for(int i=0; i<tabR.length;i++)    res+=tabR[i].getQuestion().getNbPoints();
     return res;  }

  public double getNoteGlobale()
  { double note=this.getNbPoints();
     note = note / this.getNbTotalPts() * 20;
     return note;  }

  public String toString()
  { String s="Vos Réponses\n";
    for(int i=0; i< tabR.length; i++)
          s+="Q"+(i+1) + " : "+tabR[i].getQuestion().getProposition(tabR[i].getNumReponse())+"\n";
    s+="Ce qui vous fait un total de "+this.getNbPoints()+" sur "+this.getNbTotalPts()+"\n";
    s+= "  et une note de "+this.getNoteGlobale()+" / 20";
    return s;
  }
}
```

ANNEXE 3 : Extrait de la méthode « initComponents() »

```
private void initComponents() {
    jLabel1 = new javax.swing.JLabel();
    jPanel3 = new javax.swing.JPanel();
    Gauche = new javax.swing.JPanel();
    jPanel1 = new javax.swing.JPanel();
    jLabel2 = new javax.swing.JLabel();
    CBThemes = new javax.swing.JComboBox<>();
    jPanel4 = new javax.swing.JPanel();
    jScrollPane2 = new javax.swing.JScrollPane();
    ListeQuizz = new javax.swing.JList<>();
    PanImage = new javax.swing.JPanel();
    Droit = new javax.swing.JPanel();
    jScrollPane1 = new javax.swing.JScrollPane();
    Edition = new javax.swing.JTextArea();
    jPanel2 = new javax.swing.JPanel();
    Demarrer = new javax.swing.JButton();
    Effacer = new javax.swing.JButton();
...
    jLabel1.setText("Testez vos connaissances en Interfaces Visuelles");
    getContentPane().add(jLabel1, java.awt.BorderLayout.NORTH);
    jPanel3.setLayout(new java.awt.GridLayout(1, 2));
    Gauche.setLayout(new java.awt.BorderLayout());
    jPanel1.setLayout(new java.awt.GridLayout(2, 1));
    jLabel2.setText("Choisissez le thème sur lequel vous voulez vous entrainer");
    jPanel1.add(jLabel2);
    CBThemes.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        CBThemesActionPerformed(evt);
      } });
    jPanel1.add(CBThemes);
    Gauche.add(jPanel1, java.awt.BorderLayout.NORTH);
    jPanel4.setLayout(new java.awt.GridLayout(2, 1));
    ListeQuizz.addListSelectionListener(new javax.swing.event.ListSelectionListener() {
      public void valueChanged(javax.swing.event.ListSelectionEvent evt) {
        ListeQuizzValueChanged(evt);
      } });
    jScrollPane2.setViewportView(ListeQuizz);
    jPanel4.add(jScrollPane2);
    PanImage.setLayout(new java.awt.GridLayout(1, 1));
    jPanel4.add(PanImage);
    Gauche.add(jPanel4, java.awt.BorderLayout.CENTER);
    jPanel3.add(Gauche);
    Droit.setLayout(new java.awt.BorderLayout());
...
    jScrollPane1.setViewportView(Edition);
    Droit.add(jScrollPane1, java.awt.BorderLayout.CENTER);
    jPanel3.add(Droit);
    getContentPane().add(jPanel3, java.awt.BorderLayout.CENTER);
    jPanel2.setLayout(new java.awt.GridLayout(1, 2));
    Demarrer.setText("Démarrer le test");
    Demarrer.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        DemarrerActionPerformed(evt);
      } });
    jPanel2.add(Demarrer);
    Effacer.setText("Effacer les résultats");
    Effacer.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
        EffacerActionPerformed(evt);
      }});
    jPanel2.add(Effacer);
    getContentPane().add(jPanel2, java.awt.BorderLayout.SOUTH);   }
```

ANNEXE 4 : Contenu de la classe « QuestionDlg »

```
public class QuestionDlg extends javax.swing.JDialog {

  private Questionnaire lstQ;
  private LesReponses lstR;
  private int numQc;          // numéro de la question courante
  private Reponse rc;         // réponse courante

  public QuestionDlg(Frame parent, boolean modal, Questionnaire q) { // à expliquer
    super(parent, modal);
    initComponents();
    lstQ=q;
    lstR=new LesReponses(lstQ.getNbQuestions());
    numQc=0;
    Titre.setText(lstQ.getTitre());
    initQuestion();
    rc=new Reponse(lstQ.getQuestion(numQc));
  }

  public LesReponses getLesReponses() { return lstR;}

  private void initQuestion()
  {  Question qc=lstQ.getQuestion(numQc);
     Intitule.setText("Q"+(numQc+1)+" : "+qc.getIntitule());
     for(int i=0; i<3;i++)
     {  JRadioButton rb=(JRadioButton) Questions.getComponent(i+1);
        rb.setText(qc.getProposition(i));
     }
     RB1.setSelected(true);
     Suivant.setVisible(false);
     Valider.setVisible(true);
  }

  private void ValiderActionPerformed(java.awt.event.ActionEvent evt)
  { // à compléter  }

  private void SuivantActionPerformed(java.awt.event.ActionEvent evt) {
    lstR.ajoutReponse(rc, numQc);
    if (numQc==lstQ.getNbQuestions() - 1)
        this.setVisible(false);
    else
      { if (numQc==lstQ.getNbQuestions() - 2)
            Suivant.setText("Terminer");
        numQc++;
        initQuestion();
        Resultat.setText("");
      }
  }

  private javax.swing.ButtonGroup GroupeBoutons;
  private javax.swing.JLabel Intitule;
  private javax.swing.JPanel Questions;
  private javax.swing.JRadioButton RB1;
  private javax.swing.JRadioButton RB2;
  private javax.swing.JRadioButton RB3;
  private javax.swing.JLabel Resultat;
  private javax.swing.JButton Suivant;
  private javax.swing.JLabel Titre;
  private javax.swing.JButton Valider;
  private javax.swing.JPanel jPanel2;
}
```