**SUJET 1: HORLOGE ANALOGIQUE ET NUMERIQUE**

1. **Requêtes:**

* Implémenter une horloge qui suivant le clic sur la souris permet un affichage analogique ou numérique (l’affichage doit unclure l’heure les minutes les secondes. Il faut de plus au lancement de l’application demander à l’utilisateur de saisir l’heure (cad heure: minutes: secondes).
* Supplémentaire: il y a 3 options pour choisir l’affichage:
* Horloge analogique
* Horloge numérique
* Horloge compte à rebours pour la demande à l’utilisateur de saisir l’heure

1. **Application:**

Pour réexaminer et permettre aux étudiants d’en apprendre davantage comment d’écrire le programme eux-mêmes, appliquez les connaissances, acquises et faites une application simple, cet application est convenable. Elle affiche une horloge analogique sur le bureau de Windows, celle-ci est muni d’une option de configuration souple et fonctionne selon votre préférence qui sont les fonctions de base les plus simples à utiliser. Le choix entre horloge analogique et numérique dépend l’habitude d’observateur.

* 1. Horloge analogique:
* Définition: L’affichage de l’heure analogique en horlogerie est l’affichage le plus couramment utilisé depuis plusieurs siècles. Les montres analogiques ont la particularité d’avoir un système d’indicateur de temps qui se fait par l’intermédiaire du déplacement des aiguilles sur un cadran. Pour faciliter la lecture, les heures souvent indiquées sur le cadran (cadran divisé), des petits index pouvant même indiqués chaque minute sur le cadran via un petit point ou un trait. L’affichage analogique de l’heure utilise le plus souvent trois aiguilles (une pour les heures, une autres pour les minutes et une dernière pour les secondes.



***Figure 1:*** *Horloge de gare analogique, classique ou radiopilotée (cite sur l’Internet).*

* 1. Horloge numérique:
* Définition: Une horloge numérique est un appareil électronique qui dispose de plusieurs fonctions très précises, en plus de celle d’afficher l’heure bien évidemment. L’intérêt d’une horloge numérique, outre le fait d’être un équipement pas cher, c’est le fait qu’il prendra le rôle d’un véritable assistant personnel au quotidien puisqu’il s’occupera en parfaite autonomie et avec une extrême fiabilité.



***Figure 2:*** *Illustration de l’horloge numérique (cité sur l’Internet).*

* 1. Horloge compte à rebours:
* Un compte à rebours est un décompte jusqu’à l’heure d’un événement spécifique. Il vous permet de savoir le temps restant avant un événement spécifique. Il nous permet de savoir le temps restant avant un événement donné, comme le nouvel an, course sportive, l’examen, l’éducation (technique de pomodoro), etc.



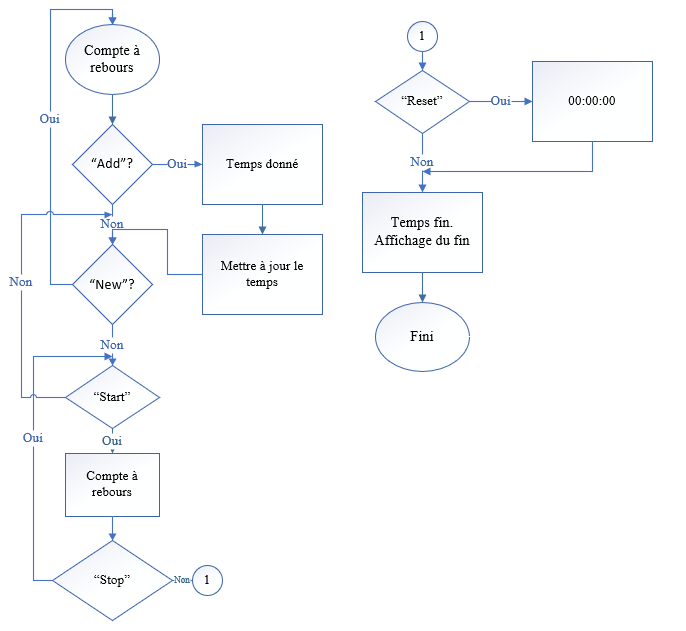
***Figure 3:*** *Illustration de l’horloge compte à rebours (cité sur l’Internet).*

1. **Le programme:**
   1. ***Diagramme du programme:***

À l’aide du diagramme du programme, on a une vue d’ensemble de toutes les étapes d’un programme.



***Figure 4:*** *Diagram de l’application.*



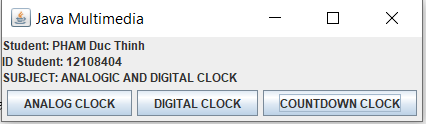
***Figure 5:*** *Diagram du compte-à-rebours d’interface.*

* 1. ***Afffichage du programme:***
     1. *Affichage générale pour visualiser les 3 boutons:*

Nous devons créer une fonction de code principale pour le fichier java, utiliser la bibliothèque de suppport pour créer une interface à l’aide de JLabel, JFrame et JPanel, voyant le fichier ci-dessous.

super(s);  
 /\*Using the ContentPane for put the visual object\*/  
 cont = this.getContentPane();  
 /\*Create the content on the interface\*/  
 JLabel StudentName = new JLabel("Student: PHAM Duc Thinh");  
 JLabel IDStudent = new JLabel("ID Student: 12108404");  
 JLabel Requirement = new JLabel("SUBJECT: ANALOGIC AND DIGITAL CLOCK");  
 /\*Panel1 content these infos above\*/  
 panel1 = new JPanel();  
 /\*Set the Layout include 3 rows and 1 column\*/  
 panel1.setLayout(new GridLayout(3,1));  
 /\*Put the elements in the panel 1\*/  
 panel1.add(StudentName);  
 panel1.add(IDStudent);  
 panel1.add(Requirement);  
 /\*Create 3 buttons: one for Countdown Clock, one for Digital Clock  
 \* and the last one for Analog Clock\*/  
 Digital\_Clock = new JButton("DIGITAL CLOCK");  
 Digital\_Clock.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JFrame clockdigital = new TextClockWindow();// TO DO CHECK HERE  
 clockdigital.setVisible(true);  
 clockdigital.pack();  
 }  
 });  
 // Add action for button  
 Countdown\_Clock = new JButton("COUNTDOWN CLOCK");  
 Countdown\_Clock.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JFrame cl = new TimerClass(); // cl abbreviation for countdown clock  
 cl.setVisible(true);  
 cl.setTitle("COUNTDOWN CLOCK");  
 cl.pack();  
 }  
 });  
 Analog\_Clock = new JButton("ANALOG CLOCK"); //TO DO CHECK HERE  
 Analog\_Clock.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JFrame ac = new Clock();  
 ac.setVisible(true); // ac abbreviation for analog clock  
 ac.setTitle("ANALOG CLOCK");  
 ac.pack();  
 ac.setSize(300, 300);  
 }  
 });  
 /\*Panel2 include 3 buttons\*/  
 panel2 = new JPanel();  
 panel2.add(Analog\_Clock);  
 panel2.add(Digital\_Clock);  
 panel2.add(Countdown\_Clock);  
  
  
 /\*Put 2 panels in the same ContentPane\*/  
 cont.add(panel1);  
 cont.add(panel2,"South");  
  
// Countdown\_Clock.addActionListener(this);  
// Digital\_Clock.addActionListener(this);  
 /\*Etablish the dimension to show\*/  
 this.pack();  
 this.setVisible(true);  
 }

Après executer le fichier, on a l’affichage ensuite:



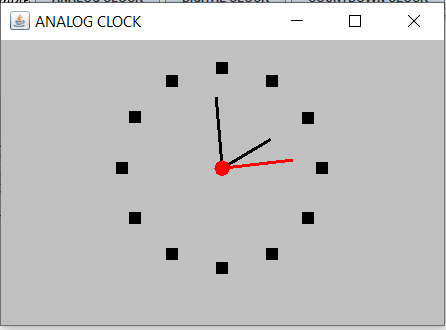
***Figure 5:*** *Affichage générale avec les 3 boutons.*

* + 1. *Affichage de l’excution du button “Analog Clock”:*

Pour l’horloge analogique, on a utilisé des tapis de référence pour dessiner les aiguilles des heures, des minutes, des secondes et la face de l’horloge. Les bibliothèques de Timer, JFrame, clockMinutes, Graphic 2D et Graphic Calendar sont appliqué pour construire l’interface du cadran.

1. public void paint(Graphics g) {  
     
    g.setColor(Color.*LIGHT\_GRAY*);  
    g.fillRect(0, 0, this.getWidth(), this.getHeight());  
    drawCardinals((Graphics2D) g);  
    drawHands((Graphics2D) g);  
     
   }  
     
   void clockMinutes(int startRadius, int endRadius, double theta) {  
    theta -= Math.*PI* / 2;  
    x[0] = (int) (getWidth() / 2 + startRadius \* Math.*cos*(theta));  
    y[0] = (int) (getHeight() / 2 + startRadius \* Math.*sin*(theta));  
    x[1] = (int) (getWidth() / 2 + endRadius \* Math.*cos*(theta));  
    y[1] = (int) (getHeight() / 2 + endRadius \* Math.*sin*(theta));  
   }  
     
   void drawCardinals(Graphics2D g) {  
    g.setStroke(new BasicStroke(12));  
    g.setColor(Color.*black*);  
     
    for (double theta = 0; theta < Math.*PI* \* 2; theta += Math.*PI* / 6) {  
    clockMinutes(100, 100, theta);  
    g.drawPolyline(x, y, 2);  
    }  
   }  
     
   public void drawHands(Graphics2D g) {  
    double h = 2 \* Math.*PI* \* (cal.get(Calendar.*HOUR*));  
    double m = 2 \* Math.*PI* \* (cal.get(Calendar.*MINUTE*));  
    double s = 2 \* Math.*PI* \* (cal.get(Calendar.*SECOND*));  
     
    g.setStroke(new BasicStroke(3)); // Size of each clockwise  
     
    clockMinutes(0, 55, h / 12 + m / (60 \* 12));  
    g.setColor(Color.*BLACK*);  
    g.drawPolyline(x, y, 2);  
     
    clockMinutes(0, 70, m / 60 + s / (60 \* 60));  
    g.setColor(Color.*black*);  
    g.drawPolyline(x, y, 2);  
     
    clockMinutes(0, 70, s / 60);  
    g.setColor(Color.*red*);  
    g.drawPolyline(x, y, 2);  
     
    g.fillOval(getWidth() / 2 - 8, getHeight() / 2 - 8, 16, 16);  
   }

Après executer le fichier, on a l’affichage ensuite:



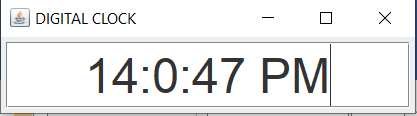
***Figure 6:*** *Affichage si on s’appuie le button “Analog clock”.*

1. *Affichage de l’excution du button “Digital Clock”:*

Nous devons créer une fonction de code principale pour activer le button “Digital Clock”, l’utilisation de la bibliothèque de suppport pour créer une interface à l’aide de la classe Calendar en java est une classe abstraite (abstract) qui fournit une méthode pour convertir des dates entre une date précise dans le temps et un ensemble de champs de calendrier tels que MOIS, ANNÉE, HEURE,… Elle hérite de la classe l’interface Comparable. L’horloge numérique est plus simple que l’horloge analogique.

class TextClockWindow extends JFrame{ //from class JFrame extend class TExtClockWindow  
 private JTextField timeField;  
 public TextClockWindow() {  
 timeField = new JTextField(10);  
 timeField.setFont(new Font("sansserif", Font.*PLAIN*, 48));  
  
 Container content = this.getContentPane();  
 content.setLayout(new FlowLayout());  
 content.add(timeField);  
  
 this.setTitle("DIGITAL CLOCK");  
 this.pack();  
  
 javax.swing.Timer t = new javax.swing.Timer(1000,  
 new ActionListener() {  
 public void actionPerformed(ActionEvent e) {  
  
 Calendar calendar = new GregorianCalendar();  
 String am\_pm;  
  
  
 Calendar now = Calendar.*getInstance*();  
 int h = now.get(Calendar.*HOUR\_OF\_DAY*);  
 int m = now.get(Calendar.*MINUTE*);  
 int s = now.get(Calendar.*SECOND*);  
  
  
 if (calendar.get(Calendar.*AM\_PM*) == 0) {  
 am\_pm = "AM";  
 } else {  
 am\_pm = "PM";  
  
 } // Code to Determine whether the time is AM or PM  
  
 timeField.setText("" + h + ":" + m + ":" + s + " " + am\_pm);  
 timeField.setHorizontalAlignment(JTextField.*CENTER*);  
 // Center the text  
 timeField.getCaret().setVisible(false);  
 // Hide the Cursor in JTextField  
 }  
 });  
 t.start();  
 }  
}

Après executer le fichier, on a l’affichage ensuite:

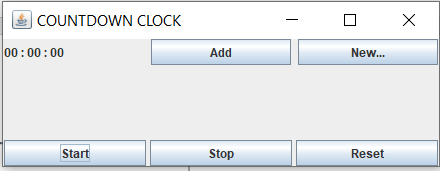


***Figure 7:*** *Affichage si on s’appuie le button “Digital clock”.*

1. *Affichage de l’excution du button “Countdown Clock”:*

Pour le “Countdown Clock” bouton, c’est un peu plus compliqué avec beaucoup de fonctionnalités et de boutons à utiliser. J’ai appris la mise en page d’une interface à l’aide de la bibliothèque JPanel, à changer de bouton avec Jbutton, à utiliser ActionListener pour appeler la fonction, à utiliser la fonction Timer pour régler l’heure, ci-dessous fait partie du programme.

public TimerClass(){  
 setLayout(new BorderLayout());  
 JPanel panel1 = new JPanel(new GridLayout(1, 3, 7, 7));  
 JPanel panel2 = new JPanel(new GridLayout(1, 3, 4, 4));  
 timerLabel = new JLabel("00 : 00 : 00");  
 originalColor = timerLabel.getForeground();  
 secTimer = new Timer(1000, new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 if (started) {  
 if (seconds == 0 && minutes == 0 && hours == 0) {  
 timerLabel.setForeground(originalColor);  
 JOptionPane.*showMessageDialog*(TimerClass.this, "Time Up", "Time's Up!", JOptionPane.*INFORMATION\_MESSAGE*);  
 secTimer.stop();  
 addButton.setEnabled(true);  
 }  
 //Change the color when 5 seconds remain  
 if ((seconds <= 5 && seconds > 0) && minutes == 0 && hours == 0) {  
 timerLabel.setForeground(Color.*RED*);  
 }  
 if (seconds > 0) {  
  
 seconds--;  
 timerLabel.setText(String.*format*("%02d : %02d : %02d", hours, minutes, seconds));  
 } else {  
 if (minutes != 0 && hours == 0) {  
 minutes--;  
 seconds = 60;  
 seconds--;  
 } else if (minutes != 0 && hours != 0) {  
 minutes--;  
 seconds = 60;  
 seconds--;  
 } else if (minutes == 0 && hours != 0) {  
 minutes = 60;  
 minutes--;  
 seconds = 60;  
 seconds--;  
 hours--;  
 }  
 timerLabel.setText(String.*format*("%02d : %02d : %02d", hours, minutes, seconds));  
 }  
 }  
 }  
 });  
  
 addButton = new JButton("Add");  
 addButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) { // Action to run a new function which has been declared  
 new TimerSpinner();  
 }  
 });  
 //creates a new timer that runs in separate thread.  
 newButton = new JButton("New...");  
 newButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 JFrame newFrame = new TimerClass();// TO DO CHECK HERE  
 newFrame.setVisible(true);  
 }  
 });  
 panel1.add(timerLabel);  
 panel1.add(addButton);  
 panel1.add(newButton);  
 startButton = new JButton("Start");  
 stopButton = new JButton("Stop");  
 resetButton = new JButton("Reset");  
 startButton.setEnabled(false);  
 stopButton.setEnabled(false);  
 resetButton.setEnabled(false);  
 panel2.add(startButton);  
 panel2.add(stopButton);  
 panel2.add(resetButton);  
 startButton.addActionListener(new ButtonHandler());  
 stopButton.addActionListener(new ButtonHandler());  
 resetButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 initHr = initMin = initSec = 00; //TO DO  
 timerLabel.setText(String.*format*("%02d : %02d : %02d", initHr, initMin, initSec));  
 hours = initHr;  
 minutes = initMin;  
 seconds = initSec;  
 secTimer.restart();  
  
 }  
 });  
  
 add(panel1, BorderLayout.*NORTH*);  
 add(panel2, BorderLayout.*SOUTH*);  
}



***Figure 8:*** *Affichage si on s’appuie le button “Countdown clock”.*

**4. Connaissances acquises:**

* En plus des connaissances présentées dans les sections ci-dessus lors de la demande, on veut ajouter un peu sur l’environnement IntelliJ IDEA 2021.3.1 pour programmer et comment executer Java et l’option pour créer l’application .jar.
* L’environnement IntelliJ IDEA 2021.3.1: est une IDEA environnement, inaugure la prise en charge du développement à distance (en bêta) ainsi qu’un nouveau moyen de dépanner les problèmes de l’IDE grâce à l’action Repair IDE. Il nous permet de nous connecter à une machine distante exécutant le backend d’IntelliJ IDEA depuis n’importe où dans le monde.
* Comment l’option pour créer un fichier .jar avec IntelliJ IDEA:
  + Voici la méthode pour créer un fichier “jar” avec IntelliJ IDEA, incluant les dépendances (autres fichiers “jar” utilisés), peut être trouvé à [5].
    - Faire “File → Project Structure… → Artifacts”
    - Cliquer sur le “+”
    - Sélectionner “JAR → From modules with dependencies…”
    - Sélectionner la “Main Class” (point d’entrée de l’application)
    - Cliquer sur “OK”
    - Dans la fenêtre suivante, cliquer sur “Apply”, puis “OK”
    - Dans la barre de menus, faire “Build → Build Artifacts…”
    - Dans la liste “Action”, sélectionnez “Build”
    - Le fichier “jar” est généré dans le répertoire que vous choisiez.

**Les points à améliorer du programme:**

* Tailler inflexible.
* La décoration de l’application est simple et à l’ancienne.
* Un cas qui n’a pas été mentionné, peut-être, la saisie manuelle de l’heure avant de démarrer l’horloge.

1. **Reference:**

[1] For countdown clock: https://www.daniweb.com/programming/software-development/threads/463090/what-code-can-i-use-to-create-a-countdown-timer-in-a-dialog-box

[2] For interface: https://codelearn.io/sharing/lam-mot-may-tinh-bang-java-swing

[3] For Analog clock: https://ispycode.com/Java/Swing/Clocks/Analog-Clock-3

[4] Digital Clock: <https://www.c-sharpcorner.com/UploadFile/16154c/digital-clock-using-swing-in-java/>

[5] Create file .jar: <https://stackjava.com/intellij-idea/huong-dan-tao-build-export-file-jar-trong-intellij-idea.html>