Group 9

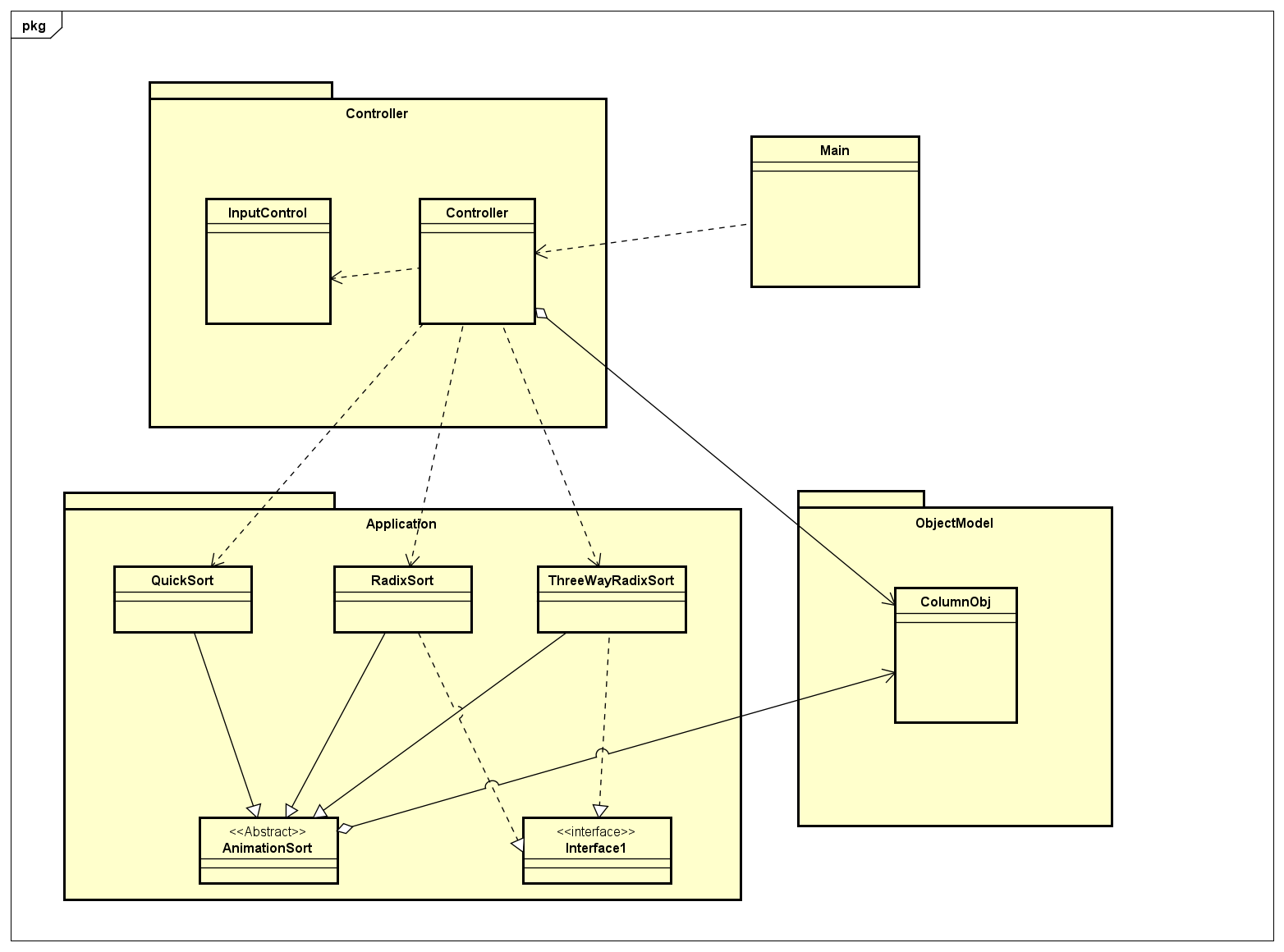
Mini-project Report

Topic: An application to (visually)demonstrate following sorting algorithms : Quick sort, Radix sort, 3-way Radix Quicksort.

Mini-project description:

* Show the way that the sorting algorithms run step by step, in order to the viewer can understand that algorithms.
* On the screen app, the user enters input - an array of integer in the text field.
* The user can select one type of sorting algorithms (Quick sort, Radix sort or 3-way Radix Quicksort).
* After choose the option, click “Run” button to play simulation.
* When the algorithm is running on screen, we can pause the process by a button “Pause”, and click again to continue the process.

Explanation of the design ideas



* Class “Main” use to run whole project. Render ViewFXML that mean screen of application. Call method Run on Controller to run program.
* Package Controller
  + Choose the sort algorithm and visualize its.
  + Classes
    - InputControl:
      * Check the input correctly
    - Controller:
      * Navigate the program to sort algorithms visualization
* Package ObjectModel
  + Class ColumnObj
    - The object describes the column respect to each numbers of input
* Package Application
  + Abstract Class AnimationSort
    - Contain methods for making the animation.
  + Class QuickSort
    - Extends abstract class AnimationSort
    - Visualizing quicksort algorithms
  + Class RadixSort
    - Extends abstract class AnimationSort
    - Visualizing radix sort algorithms
  + Class ThreeWayRadixQuickSort
    - Extends abstract class AnimationSort
    - Visualizing 3Way radix quicksort algorithms
  + Interface ColumnColor
    - Interact to column color

Assignment of members:

***Package ObjectModel:***

* Class **ColumnObj**: (Lê Đình Quyền 20176859 + Trần Xuân Đức 20176720 + Vũ Ngọc Nam 20176832)  
  => Create the main Object of project. The object include a rectangle like a column and a label have number input on it.
  + **private** **int** Xcurrent; => x coordinate in the current time of process
  + **private** **int** xCol; => layout X of Object
  + **private** **int** yCol; => layout Y of Object
  + **private** **int** width; => the width of column
  + **private** **int** height; => the height of column
  + **private** String idCol; => The id to set for rectangle
  + **private** String idlabel; => the id to set for label
  + **private** ArrayList<Animation> AnimaCol = **new** ArrayList<Animation>();
    - **list of animations set for rectangle in timeline**
  + **private** ArrayList<Animation> AnimaLab = **new** ArrayList<Animation>();
    - **list of animations set for label in timeline.**
  + **private** SequentialTransition SeqAnimaCol = **new** SequentialTransition();
    - **Sequence of rectangle’s animation – like action of object**
  + **private** SequentialTransition SeqAnimaLab = **new** SequentialTransition();
    - **Sequence of Label’s animation – like action of object**
  + Public Rectangle rectangle()
    - To create a rectangle like a column corresponding to value of number input.
  + Public Rectangle getRectangle(Pane parent)
    - To get element by IdCol in Pane and return exactly the rectangle of this object in the Pane on Screen.
  + Public Label label()
    - To create a Label like a ground floor of object corresponding to value of number input.
  + Public Label getLabel(Pane parent )
    - To get element by IdLabel in Pane and return exactly the label of this object in the Pane on Screen.
  + Public void BeginPlayAnimationColumn(Pane parent)
    - To get object on screen and set play() animation of this object.
  + Public void Pause()
    - To pause the sequential animation of object.
  + Public void Play()
    - To continue the sequential animation of object.

***Package Controller***

* Class InputControl
  + Public static boolean TestInput(String str) - (Lê Đình Quyền 20176859)
    - Check string input get from text field on screen

If string null or having any characters is not digit or space return false and show a dialog error into screen.

* + Public static int[] toArray(String str) - (Lê Đình Quyền 20176859)
    - To change string input integer to an array of integer.
  + public static int getNumberCharacters(ArrayList<ColumnObj> columnObjList) - (Vũ Ngọc Nam - 20176832)
    - Get the maximal number of characters of a number.
  + public static String toStringColumnHeight(int height, int numberCharacters) - (Vũ Ngọc Nam - 20176832).
    - When you know the maximal number of characters of maximal number of the array, each numbers will be inserted ‘0’ into the first position such that the length of each number equals to this.

* Class Controller(Trần Xuân Đức 20176720)
  + Public void Run(ActionEvent e)
    - Run application
  + Pubic void action1(ActionEvent e)
    - Handler event when user choose Quick Sort in MenuItem, change some attribute of controls such as title, menuButton.
  + Public void action2(ActionEvent e)
    - Handler event when user choose Radix Sort in MenuItem, change some attribute of controls such as title, menuButton.
  + Public void action3(ActionEvent e)
    - Handler event when user choose 3-way Radix Sort in MenuItem, change some attribute of controls such as title, menuButton.

***Package Application***

* Abstract Class AnimationSort - (Lê Đình Quyền – 20176859)
  + Public void Objpicked(ColumnObj X)
    - choose the object add animation on UI to show that the column is picked.
  + Public void ObjPause(ColumnObj X, int timePause)
    - add animation Pause for object.
  + Public void SwapUI (ColumnObj A, ColumnObj B)
    - swap 2 object on UI
  + Public void PivotUPAnimation(ColumnObj X)
    - Animation for object is chosen as Pivot
  + Public void PivotDownAnimation(ColumnObj X)
    - Animation for object Pivot is done.
  + Private boolean isContain(int A[], int k)
    - consider k in array A[]?
  + Public void SetPauseTheRest(ArrayList<ColumnObj> List, int... ExceptIndex )
    - Set pause for all object that not in ExceptIndex
  + Public swap(ArrayList<ColumnObj> list, int i, int j)
    - use in quick sort and 3-way radix quick sort.
* Interface ColumnColor (Vũ Ngọc Nam - 20176832 + Trần Xuân Đức –20176720)
* Class QuickSort (Lê Đình Quyền – 20176859)
  + Public void Objpicked(ColumnObj X, Color color)
    - Choose the column object show in screen what number is picked.
  + Private int partition(ArrayList<ColumnObj> List, int low, int high)
    - A part of quick sort algorithm. When swap value on memory, We set animation to swap object on UI, respectively.
  + Public void quicksort(ArrayList<ColumnObj> List, int low, int high)
    - Main function of quick sort algorithm.
  + Private Label Note(String text, String color, int x, int y)
    - Create the notation for i, j, pivot index
  + Public QuickSort(Pane P, Arraylist<ColumnObj> Data)
    - Run program to show Quick sort algorithm on screen.
* Class RadixSort (Trần Xuân Đức -20176720)
  + Extends AnimationSort
    - Get the basic animation of AnimationSort
  + Implements ColumnColor
    - Override method for interact with column color
  + Public int getMax()
    - the Max value of integer input List
  + Public void getXColumn()
    - Store XColumn of all Column in ListObj to array xColumn[int]
  + Public void moveObj(ColumnObj X, int newX)
    - Move the column X to new posotion with setToX(newX)
  + Public void setColumnColor(ColumnObj X, Color color)
    - Override the method setColumnColor in the Interface and change color of a Column and n change color of Label
  + Public void down(ColumnObj X)
    - Down position of ColumnObj
  + Public void countSort(int exp)
    - A part of RadixSort, a function to do counting sort the height of ListObj according to the digit represented by exp. In this time, add corresponding animation for ColumnObj.
  + Public RadixSort(Pane pane, ArrayList<ColumnObj> ListObj)
    - Run program to show Radix Sort algorithm on screen.

* Class ThreeWayRadixSort - (Vũ Ngọc Nam - 20176832)
  + Extends AnimationSort
    - Get the basic animation of AnimationSort
  + Implements ColumnColor
    - override method for interact with column color
  + private void radixSort3Way(ArrayList<ColumnObj> ListObj, int low, int high, int index, int numberCharacter)
    - Using sort algorithm and animation for visualizing.
  + private void setColorForCurrentArray(ArrayList<ColumnObj> listObj, int lt, int gt, Color color)
    - Set the color for the list column with its range is between lt and gt
  + public void setColumnColor(ColumnObj columnObj, Color color)
    - Override the method setColumnColor in the Interface ColumnColor and set the color for a column

***Package ViewFXML*** (Trần Xuân Đức -20176720)

Using JavaFX and SceneBuilder to build GUI for application