



MEMBERS









LE DUC ANH TUAN

DAO VAN TUNG BUI THANH TUNG NGUYEN ANH TUAN



Problem statement

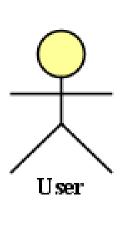


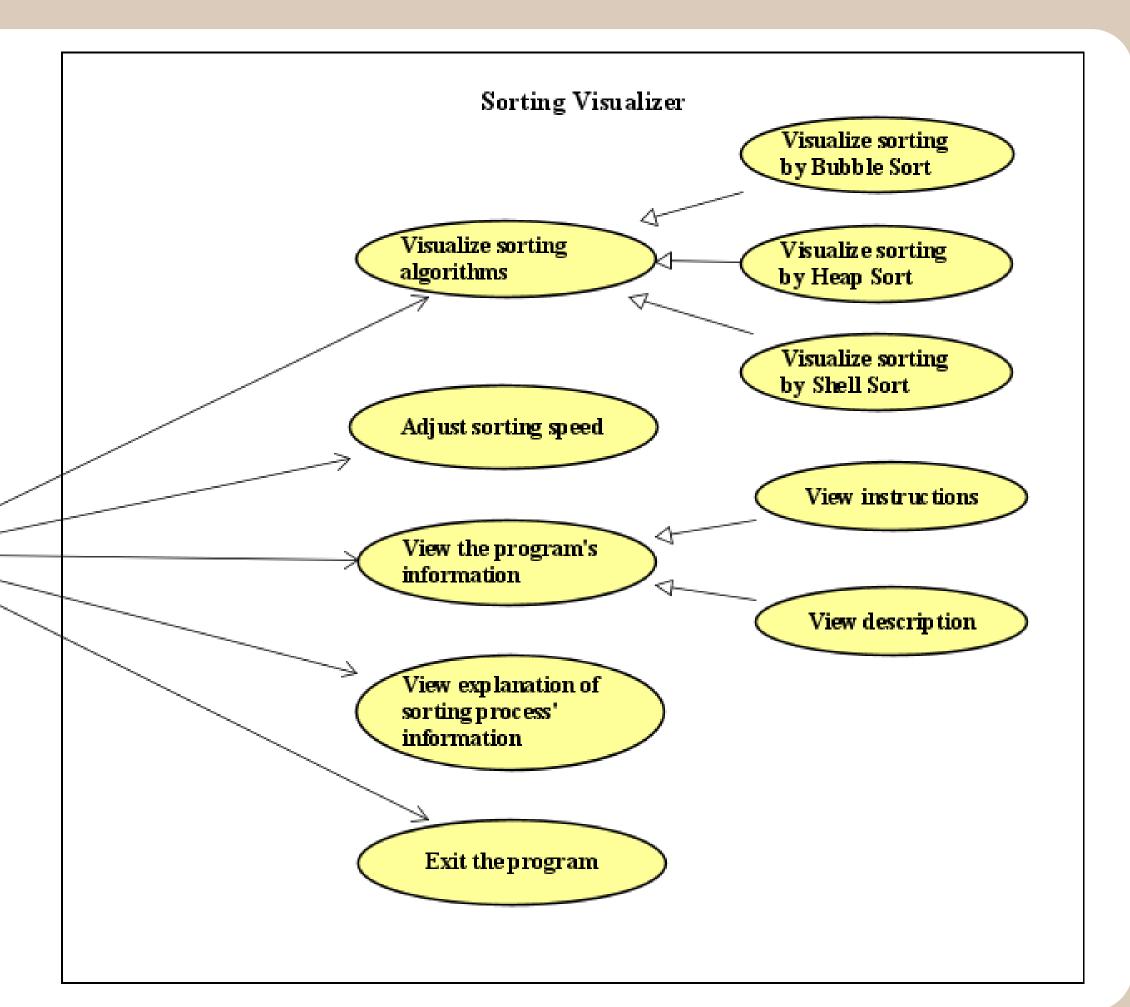
Array is the most basic structure of computer science. Most operations as well as other data structures are built and performed on an array.

In this project, we make an application GUI to explain three sorting algorithms on an array: bubble sort, heap sort, shell sort.

Use case diagram

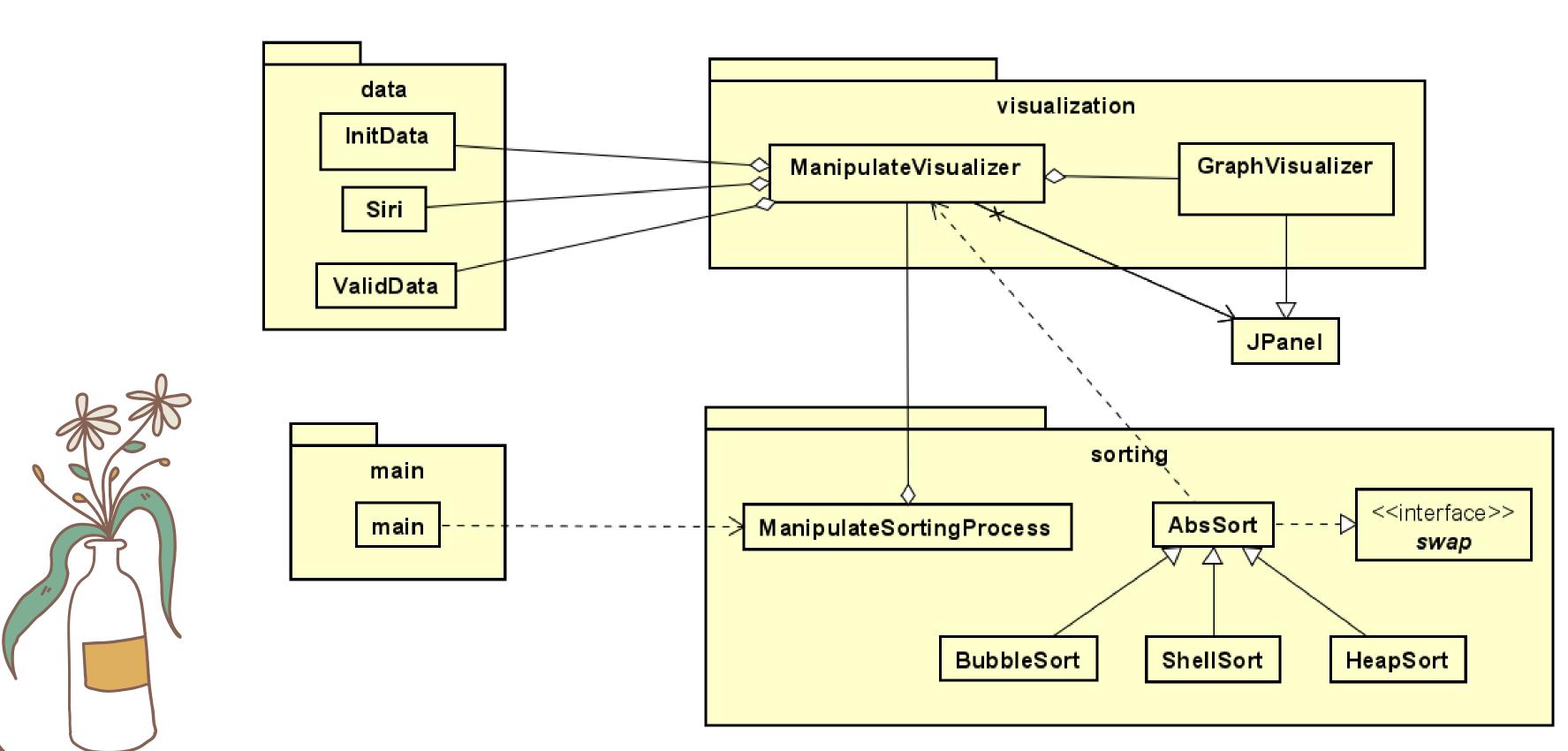


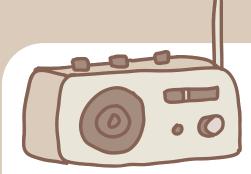




General class diagram









Main package

main

+ main(args : String[]) : void

Data package

InitD ata

- length : int - array : int[]
- isSorting : boolean - isPause : boolean

- isStop : boolean

- +InitData(len:int,isSorting:boolean,isPause:boolean,isStop:boolean)
- + setSorting(isSorting : boolean) : void
- +isSorted() boolean
- + basicArray(): void
- + swap(p1 : int, p2 : int) : void + genRandomArr() : void

ValidD ata

- + checkNullOrEmpty(str : String) : boolean
- + checkNumber(str : String) : boolean

Siri

- + Siri()
- + deleteNewLineTabSpaces(str : String) : String
- + StrTo Arr(str : String , sep Char : String) : int[]
- + ArrayToString(arr:int[], sepChar:String):String

Sorting package

ManipulateSortingProcess

- manipulateVisualizer : ManipulateVisualizer
- current int
- length int
- array : int[]
- isSorting | boolean
- isPause : boolean - isStop : boolean
- -speed int
- curAlg:int
- + ManipulateSortingProcess(len: int, array: int[], isSorting: boolean, isPause: boolean, isStop: boolean, curAlg: int, speed: int, current: int, check: int)
- + isSorted(): boolean
- + init(): void
- + sorting(): void
- + pause() : void
- + reset() : void
- + delay() : void
- + Update() : void

<<interface>> swap

+ swap(il int i2 int) void

AbsSort

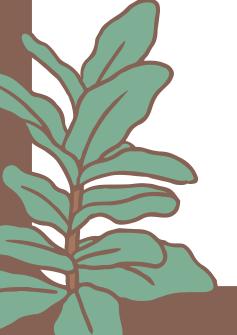
- manipulateVisualizer : ManipulateVisualizer
- length int
- array int[]
- isSorting: boolean
- isPause : boolean
- isStop : boolean
- + swap(i1 : int, i2 : int) : void

HeapSort

- current int = -1
- + InsertionSort(length; int, array; int[], isSorting; boolean, isPause; boolean, isStop; boolean, mv;
- ManipulateVisualizer)
- + sort(start : int. end : int) : void

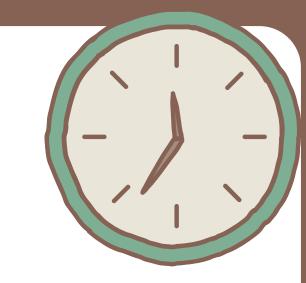
ShellSort

- current: int = -1
- check : int = -1
- + BubbleSort(length int, array int
- [], isSorting | boolean, isPause | boolean, isStop | boolean, mv |
- ManipulateVisualizer)
- + sort(start: int, end: int): void



BubbleSort

- current: int = -1
- -check:int=-1
- + QuickSort(length int, array int [], isSorting boolean, isPause boolean, isStop boolean, mv
- ManipulateVisualizer)
- + sort(start int, end int) void



Visualization



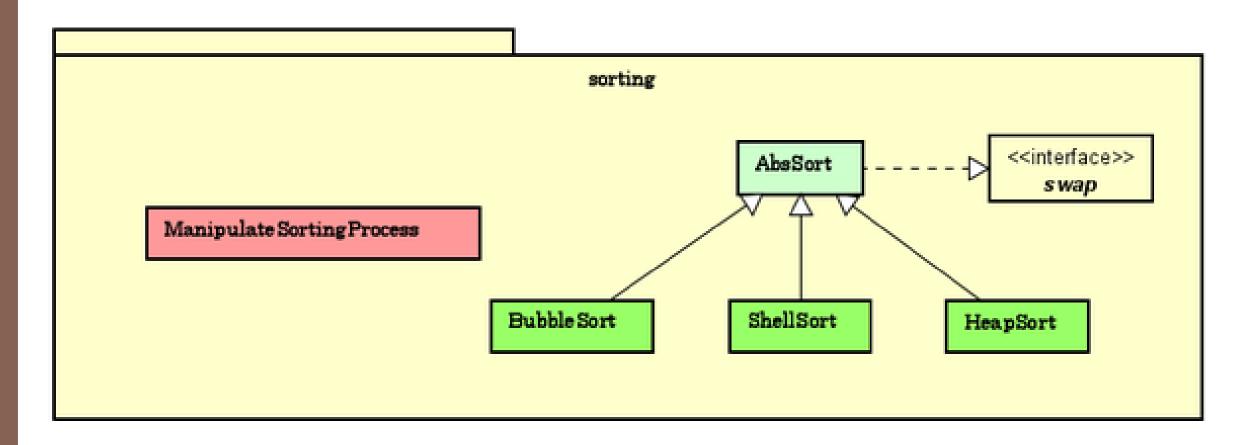
Manipul ateVisualizer

- validator : ValidData
- helpers Siri
- canvas : GraphVisualizer
- r : Random
- current int
- check int
- length int
- array int[]
- isSorting boolean
- isPause boolean
- isStop boolean
- speed : int = 50
- compared, arrayAccessed, curInputDataOption, curAlg : int = 0
- sortingProcessListMsg | String[] = { "Bubble sort" | "Heap sort" | "Shell sort" }
- sortingProcessMsg : String = sortingProcessListMsg[curAlg]
- · ABOUT_MESSAGE : String = "Sorting V ...
- HELP_INSTRUCTION_MESSAGE : String = "Here is t...
- MAX ARRAY LENGTH: int = 300
- displayTextArea String = "[\n \n]"
- GRAPH SIZE int = 600
- rectangle width int
- genDataOptions, helpGenDataMsg, algorithmOptions, algorithmListInfo: String[]
- iframe JFrame
- genDataPane, controlsPane: JPanel
- delayLabel.speedLabel.comparedLabel.arrayAccessedLabel.genDataOptionLabel.arrayLengthLabel. arrayLengthErrorLabel, algorithmOptionLabel, algorithmInfoLabel, sortingProcessLabel; JLabel
- genDataOptionComboBox, algorithmComboBox; JComboBox
- inputArrayArea, algorithmInfoArea: JTextArea
- inputArrayScrollPane : JScrollPane
- btnGenerateArray, btnHelpGenerateArray, btnStartSort, btnPauseSort, btnStopSort, btnResumeSort, btnHelpComparison, htnHelpAccess, btnHelpIntructions, btnAbout : JButton
- arrayLengthInput JTextField
- speedSlider : JSlider
- loweredEtched : Border
- + ManipulateVisualizer(length:int, array:int[], isSorting:boolean, isPause:boolean, isStop:boolean, cur:int, check:int)
- + initialize(): void
- + updateWhenSortDone(): void
- + updateProcess(length:int, array:int[], current:int, check:int): void
- + updateArrayAccessed(i:int):void
- + delay() : void

GraphVisualizer

- rectangle_width int
- length int
- array int[]
- current int
- check int
- + getCurrent() int
- + setCurrent (current : int) : void
- + getCheck():int
- + setCheck (check : int) : void
- + getArrav() : int[]
- + setArray(array : int[]) : void
- + getRectangle_width():int
- + setRectangle_width(rectangle_width:int):void
- + getLength():int
- + setLength (length : int) : void
- + GraphVisualizer(recWid:int, length:int, ar:int[], cur:int, check:int)
- + paintComponent(g: Graphics): void

Polymorphism + Inheritance



Inhertitance: AbstSort class inherit from swap interface

Polymorphism: 3 class sort extend class Abs for override from method sort abs

Demo Video

