Sorta Awesome

David Frid, Karina Ionkina, Mark Shafran

APCS2 pd05
HW48 -- On Target
2017-06-07

Sorta Awesome UML Diagram

Driver Class

SA

int rect1X, rect1Y, rect2X, rect2Y, rect3X, rect3Y
int rect4X, rect4Y, rect5X, rect5Y, rect6X, rect6Y
int recW, recH;
color rectColor, rect2Color, rect3Color, fillcolor;
boolean Over1, Over2, Over3, Over4, Over5;
boolean Over6, Over7, Over8;
int state;
BubbleSortLarge bubL;
SelectionSortLarge selL;
InsertionSortLarge insL;
InsertionSortSmall insS;
SelectionSortSmall selS;
BubbleSortSmall bubS;
BogoSortSmall bogS;
BogoSortLarge bogL;
HeapsortSmall heaS;
HeapSortLarge heaL;

void setup();
void draw();
void makeText(String, int, int, int);
void update();
void mouseClicked();
boolean overRect(int,int);
void keyPressed();

For Small Scale Visualization

HeapSortSmall

int [] arr;
float levelX, levelY;
int arrSize, boxWidth, boxHeight, rootY, rootX, i, counter
boolean filling, odd;
theSort arr1;

void HeapSortSmall();
int log(int, int)
void drawHSS();
void display(int[ ])

theSort

int [] arr;
ArrayList<int[ ]> order;
boolean nyoom;
ArrayList<Integer> sorted;

void theSort(int[ ]);
void heapSortV();
int maxChildPos(int[ ], int, int);
int[ ] dupe(int[ ])
void swap(int[ ], int, int)

node

float ex, why;

void node (float, float, int, float, boolean)

InsertionSortSmall

int [] arr;
float boxWidth, boxHeight;
int firstBox, secondBox, arrSize, i , maxPos, pass;

void InsertionSortSmall();
void drawIS();

BogoSortSmall

int [] arr;
float boxWidth, boxHeight;
int firstBox, secondBox, arrSize, a, b, partition, i, old
boolean swapping

void BogoSortSmall();
void drawBGS();
bolean sorted(int[ ])

SelectionSortSmall

int [] arr;
float boxWidth, boxHeight;
int firstBox, secondBox, arrSize, a, b, partition, i, old
boolean swapping, last

void SelectionSortSmall();
void drawSS();

BubbleSortSmall

int [] arr;
float boxWidth, boxHeight;
int firstBox, secondBox, arrSize, a, b, partition, i, old
boolean swapping, last

void BubbleSortSmall();
void drawBS();

For Large Scale Visualization

rArrIS

ArrayList<Integer> count
ArrayList<Rectangles[ ]> order;
Rectangles[ ] rawr;
int[ ] arr;

rArrIS(Rectangles[ ]);
Rectangles[ ] dupe(Rectangles[ ]);
void InsertionSortV()
void drawIL(int)
void swap(Rectangles[ ], int, int)

InsertionSortLarge

int counter;
rArrIS rects;
int[] arr

InsertionSortLarge();
void drawIL();
void setArr();

Rectangles

float x, y, w, h;
color c;

Rectangles(float,float,float,float,color);
void draw(int,int,boolean)
float getHeight();
void setHeight(float);
Rectangles dupe();

SelectionSortLarge

int counter;
rArrSS rects;
int[] arr

SelectionSortLarge();
void drawSL();
void setArr();

BubbleSortLarge

int counter;
rArrBS rects;
int[] arr

BubbleSortLarge();
void drawBL();
void setArr();

rArrBS

ArrayList<Integer> count
ArrayList<Rectangles[ ]> order;
Rectangles[ ] rawr;
int[ ] arr;

rArrBS(Rectangles[ ]);
Rectangles[ ] dupe(Rectangles[ ]);
void BubbleSortV()
void drawBL(int)
void swap(Rectangles[ ], int, int)

BogoSortLarge

int counter;
ArrayList<Integer> arr

void BogoSortLarge();
void drawBGL();
void setArr();
boolean notSorted (ArrayList< Integer>);

rArrSS

ArrayList<Integer> count
ArrayList<Rectangles[ ]> order;
Rectangles[ ] rawr;
int[ ] arr;

rArrIS(Rectangles[ ]);
Rectangles[ ] dupe(Rectangles[ ]);
void SelectionSortV()
void drawSL(int)
void swap(Rectangles[ ], int, int)

HeapSortLarge

int counter;
rArrHS rects;
int[] arr

HeapSortLarge();
void drawHS();
void setArr();

rArrHS

ArrayList<Integer> count
ArrayList<Rectangles[ ]> order;
Rectangles[ ] rawr;
int[ ] arr;

rArrHS(Rectangles[ ]);
Rectangles[ ] dupe(Rectangles[ ]);
void heapSortV()
void drawH(int)
void swap(Rectangles[ ], int, int)
void setArr();
int maxChildPos (Rectangles[ ], int, int)