Luke Fitzgerald

Treadmill Program

Class Definitions

class Treadmill

**Interface Data Type Description**

speedLabel JLabel Label for speed text field

inclineLabel JLabel label for incline text field

quickInclineLabel JLabel label for quick incline buttons

quickSpeedLabel JLabel label for quick speed buttons

paceLabel JLabel label for pace text field

timerLabel JLabel label for time text field

speedTF JTextField speed text field

inclineTF JTextField incline text field

distanceTF JTextField distance text field

paceTF JTextField pace text field

timeTF JTextField time text field

onOrOff JButton On/Off button

start JButton Start button

stopReset JButton Stop/Reset button

inclineUp JButton Increase Incline button

inclineDown JButton Decrease Incline button

speedUp JButton Increase Speed button

speedDown JButton Decrease Speed button

quickIncline1 JButton Quick Incline 1 button

quickIncline2 JButton Quick Incline 2 button

quickIncline4 JButton Quick Incline 4 button

quickIncline6 JButton Quick Incline 6 button

quickSpeed3 JButton Quick Speed 3 button

quickSpeed4 JButton Quick Speed 4 button

quickSpeed6 JButton Quick Speed 6 button

quickSpeed7 JButton Quick Speed 7 button

**Instance Variables Type Description**

w2 Window Window 2

w3 Window Window 3

bhandle Button Handler Listens for Action Event

clock1 Clock Sets up clock used for displaying time and distance on screen

treadmill1 TreadmillModel2 Creates treadmill object

stopwatch[] Timer Array of stopwatches

timer1 javax.swing.Timer Timer used for displaying the time on screen

displayDistanceTimer javax.swing.Timer Timer used for displaying the distance on screen

runTracker1 RunTracker Creates RunTracker object to track distance of the run

onOffControl int controls whether the On/Off button is on or off

displayDistance DistanceTimer Handles the displaying of distance on screen

f Font font used for On/Off button

**Methods Type Params Description**

Treadmill none none Constructor – sets up main interface

class Window2

**Interface Data Type Description**

weight JLabel Label for Weight Textfield

enterWeight JTextField place for user to enter weight

enter JButton Button to accept a user’s weight

gender JComboBox user picks gender

genderArray[] String array for the gender combo box

**Instance Variables Type Description**

weightAmount int handles the user’s weight

genderNum int handles the user’s gender

bhandle ButtonHandler listens for an action event

**Methods Type Params Description**

Window2 none None Constructor - Builds window 2

class Window3

**Interface Data Type Description**

time JLabel Label for time field

distance JLabel Label for distance field

caloriesBurned JLabel label for calories burnt

printLabel JLabel label for printing results

timeTF JTextField displays time

distanceTF JTextField displays distance

caloriesTF JTextField displays calories

yes JRadioButton option yes to print

no JRadioButton option no to print

enter JButton button to print or not print results

yesNo ButtonGroup group for the yes/no buttons

**Instance Variables Type Description**

bhandle Buttonhandler listens for an action event

timeString String final time

distanceString String final distance

calories String calories burned

**Methods Type Params Description**

Window3 none None Constructor - Builds window 3

class TreadmillModel1

**Instance Variables Type Description**

treadmillOn Boolean tells whether the treadmill is on or off

start Boolean starts a treadmill run

reset Boolean stops a treadmill run

speed double keeps track of how fast treadmill is going

**Methods Type Params Description**

TreadmillModel1 none None Constructor - Sets all fields to default value

increaseSpeed void none increases speed by .1

decreaseSpeed void none decreases speed by .1

setTreadmillOn void none sets treadmillOn to true

setTreadmillOff void none sets treadmillOn to false

setStart void none sets start to true

setResetOn void none sets reset to true

setResetOff void none sets reset to false

setSpeed void MPH sets speed to MPH

getSpeed double none returns speed

getFormattedSpeed String none returns formatted speed as a string

class TreadmillModel2

**Instance Variables Type Description**

incline double tracks the incline of the treadmill

pace double tracks pace of the treadmill

weight int weight of the user

calories int calories burned by user

gender int whether the user is male or female

**Methods Type Params Description**

TreadmillModel2 none none Constructor –sets data to default values

increaseIncline void none increases incline by .5

decreaseIncline void none decreases incline by .5

setIncline void percentIncline sets incline to percentIncline

getIncline double none returns incline

setWeight void weight sets weight

getCalories int none returns calories

getFormattedPace String none returns formatted pace as a string

setCaloriesBurned void distance calculates calories burned

setGender void gender sets gender to gender parameter

class Timer

**Instance Variables Type Description**

isRunning Boolean tells whether the treadmill is running

startTime long the start time of the clock

elapsedTime long how much time has passed

stopTime long end time of the clock

sec long seconds

min long minutes

formattedTime String formatted time in a string

**Methods Type Params Description**

Timer none none Constructor – sets data to default values

start void none starts the timer/sets isRunning to true

stop void none stops the timer/sets isRunning to false

getElapsedTime long none Calculates amount of time passed

getIsRunning Boolean none returns isRunning

formatTime String none returns time as a formatted string

resetTime void none resets all values to 0

class RunTracker

**Instance Variables Type Description**

speed double speed of treadmill

pace double pace of treadmill

distance double distance traveled on treadmill

**Methods Type Params Description**

RunTracker none none Constructor – sets data to 0

getPace double none returns pace

getSpeed double none returns speed

setSpeed void speed sets speed to parameter speed

increaseSpeed void none increases speed by .1/calls setPace

decreaseSpeed void none decreases speed by .1/ calls setPace

setPace void none formats and sets pace

calculateDistance void endTime calculates the distance traveled

getDistance String none returns formatted distance as a string

resetDistance void none resets distance to 0

private class Clock

**Methods Type Params Description**

actionPerformed void ActionEvent displays the time on the main interface

private class Distance Timer

**Methods Type Params Description**

actionPerformed void ActionEvent displays the distance on the main interface

private class ButtonHandler

**Methods Type Params Description**

actionPerformed void ActionEvent controls all of the components of the main user interface