Joshua McCann  
OOP Principles – LAB 5

1. Create three Helper methods in the StringHelper class

|  |
| --- |
| package common.helpers;  */\*\*  \* Created by Joshua.McCann on 6/26/2017.  \*/* public class StringHelper {   //Notes: this checks a string if it is null or empty returns TRUE, otherwise returns FALSE.  public static boolean isNullOrEmpty(String s){  return s==null || s.length()==0;  }   public static String compareAlphabetical(String s1, String s2){  if (s1.charAt(0) < s2.charAt(0))  return s1;  else  return s2;  }   public static String sortString(String s){  char tempChar;  char[] tempArray = s.toCharArray();  StringBuilder returnString = new StringBuilder();   for(int i = 0; i!=s.length()-1; i++) {  if (i <= s.length() - 1 && tempArray[i] > tempArray[i + 1]) {  tempChar = tempArray[i];  tempArray[i] = tempArray[i + 1];  tempArray[i + 1] = tempChar;  i = -1;  }  }  for(char test : tempArray) {  returnString.append(Character.*toString*(test));  }  return returnString.toString();  } } |

1. Create three Helper methods in the MathHelper class

|  |
| --- |
| package common.helpers;  */\*\*  \* Created by Joshua.McCann on 6/26/2017.  \*/* public class MathHelper {   public static final double *E* = 2.71;  public static final double *PI* = 3.14;   public static int square(int val) {  return val \* val;  }   public static int biggerInt(int int1, int int2){  if (int1>int2)  return int1;  else  return int2;  }   public static int smallerInt(int int1, int int2){  if (int1<int2)  return int2;  else  return int1;  } } |

1. Come up with a scenario where using interfaces would be useful.

Since both of the classes vehicle and person have the names made up of multiple strings, it might be worth utilizing the full name of both the person (made up of the first and last names) and the full name of the vehicle (make and model) in a single string.

1. Create the interface, and at least 2 concrete implementations of the interface.

|  |
| --- |
| package com.astontech.bo.interfaces;  */\*\*  \* Created by Joshua.McCann on 6/27/2017.  \*/* public interface IGetFullName {  public String getFullName(); } |
| @Override public String getFullName(){  if(StringHelper.*isNullOrEmpty*(this.FirstName) && StringHelper.*isNullOrEmpty*(this.LastName)){  return "No Name Set";  }  else{  if(StringHelper.*isNullOrEmpty*(this.FirstName))  return this.LastName;  else if (StringHelper.*isNullOrEmpty*(this.LastName))  return this.FirstName;  else  return this.FirstName + " " + this.LastName;  }  } |
| @Override public String getFullName(){  return this.getVehicleMakeName() + " " + this.getVehicleModelName(); } |

1. Create a method in Main class that accepts the interface as a parameter.

|  |
| --- |
| public static void fullNameInterface(IGetFullName IgetFullName, String s){  System.*out*.println(s + ": " + IgetFullName.getFullName());  } |

1. Write a class that implements the CharSequence interface found in the java.lang package. Your implementation should return the string backwards.

|  |
| --- |
| package com.astontech.bo;  import java.util.ArrayList; import java.util.List;  */\*\*  \* Created by Joshua.McCann on 6/27/2017.  \*/* public class OopPrinciplesLab5 implements CharSequence {   //region PROPERTIES  private String objectString;  //endregion   //region CONSTRUCTORS  public OopPrinciplesLab5() {}   public OopPrinciplesLab5(String s){  this.objectString = s;  }  //endregion   //region GETTERS / SETTERS   public String getObjectString() {  return objectString;  }   public void setObjectString(String objectString) {  this.objectString = objectString;  }   //endregion   @Override  public CharSequence subSequence(int start, int end) {  if(start>end) {  StringBuilder tempString = new StringBuilder();  while (start >= end) {  tempString.append(Character.*toString*(this.getObjectString().charAt(start)));  start--;  }  return tempString;  }  else {  return this.getObjectString().subSequence(start, end);  }  }   @Override  public char charAt(int index) {  return this.getObjectString().charAt(index);  }   @Override  public int length() {  return this.getObjectString().length();  } } |

1. Implement Comparable on your Person and Vehicle class

|  |
| --- |
| @Override public int compareTo(Object o) {   Person temp = (Person) o;   if (temp.getLastName().toLowerCase().charAt(0) < this.getLastName().toLowerCase().charAt(0))  return -1;  else if (temp.getLastName().toLowerCase().charAt(0) > this.getLastName().toLowerCase().charAt(0))  return 1;  else  return 0; } |
| @Override public int compareTo(Object o) {  Vehicle temp = (Vehicle) o;   if(temp.getVehicleMakeName().toLowerCase().charAt(0) < this.getVehicleMakeName().toLowerCase().charAt(0))  return 1;  else if(temp.getVehicleMakeName().toLowerCase().charAt(0) > this.getVehicleMakeName().toLowerCase().charAt(0))  return -1;  else  return 0; } |