Main Code for Super-Hero Polymorphism Assignment (Class: MainSuperPoly)

Key: Yellow for Polymorphic Variables

Dark Green for Polymorphic Methods

Blue for Polymorphic Method Calls

// Michael Jay Marajas Superhero Polymorphism Assignment

// COP 2800C

// 4/21/2020

// Purpose: I tested the Menu as well as the List and Sort elements within that menu using Polymorphic Calls, Methods, and Interfaces.

**import** java.util.Scanner;

//I needed to import this tool to instantiate a Scanner object for the user interaction

**public** **class** MainSuperPoly {

**boolean** exit;

**public** **static** **void** main(String[] args) {

MainSuperPoly menu = **new** MainSuperPoly();

menu.runMenu();

}

**public** **void** runMenu() {

//I called the Abstract method using a Polymorphic call

openingMessage OpenMess = **new** PolyInterfaceImp();

headerMenu();

OpenMess.openMess();

**while** (!exit) {

tabMenu();

@SuppressWarnings("unused")

**int** choice = getInput();

performAction(choice);

}

}

**private** **void** headerMenu() {

System.***out***.println("+---------------------------+");

System.***out***.println("| SuperHero |");

System.***out***.println("| Main Menu |");

System.***out***.println("+---------------------------+");

}

**private** **void** tabMenu() {

System.***out***.println("\n Please select an option");

System.***out***.println("1) List the details of 20 Superheroes");

System.***out***.println("2) Sort the Superheroes by Type");

System.***out***.println("0) Exit");

}

**private** **int** getInput() {

@SuppressWarnings("resource")

Scanner user = **new** Scanner(System.***in***);

**int** choice = -1;

**while**(choice < 0 || choice > 2) {

**try** {

System.***out***.print("Enter the number of the option you wish: ");

choice = Integer.*parseInt*(user.nextLine());

}

**catch** (NumberFormatException e){

System.***out***.println("You typed an Invalid Option. Please try Again.");

}

}

**return** choice;

}

**private** **void** performAction(**int** choice) {

//I instantiated the Polymorphic objects to access the Abstract methods in PolyInterface Imp

detailList detailedlist = **new** PolyInterfaceImp();

sortList sortedlist = **new** PolyInterfaceImp();

//I used a switch statement to route the user in the 3 possible choices

**switch**(choice) {

**case** 0:

exit = **true**;

System.***out***.println("Thank You for Using this Application");

**break**;

**case** 1:

System.***out***.println("\n");

System.***out***.println("This is the Superhero Details List");

System.***out***.println("\n");

listSuperhero();

System.***out***.println("\n");

detailedlist.listDetails();

**break**;

**case** 2:

System.***out***.println("\n");

System.***out***.println("Sorted Groups:");

System.***out***.println("\n");

sortSuperhero();

System.***out***.println("\n");

sortedlist.listSort();

**break**;

**default**:

System.***out***.println("An unknown error has occurred");

}

}

//This is where the Polymorphic Array is Stored

**public** **void** listSuperhero() {

Person people[] = **new** Superhero[100];

people[0] = **new** Elementalist("Dr. Manhattan", 27, "to Warp Reality", "The Watchmen");

people[1] = **new** Gadgeteer("Spiderman", 25, "my Web-Shooters and Spider Powers", "the Avengers");

people[2] = **new** Tank("The Thing", 30, "my raw physical Prowess and Strength", "the Fantastic Four");

people[3] = **new** Gadgeteer("Iron Man", 25, "my Suits and Technological Prowess", "the Avengers");

people[4] = **new** MartialArtist("Daredevil", 29, "my Superhuman Senses and Physical Skills", "The Defenders");

people[5] = **new** Tank("Hulk", 30, "my ability to turn into a Monster", "the Avengers");

people[6] = **new** Elementalist("Storm", 25, "Controlling the Weather", "X-Men");

people[7] = **new** MartialArtist("Iron Fist", 27, "my Superlevel-Fists", "The Defenders");

people[8] = **new** Gadgeteer("Batman", 25, "my Martial Arts and Unique Gadgets", "the Justice League");

people[9] = **new** Elementalist("The Human Torch", 23, "Controlling the Fire Element", "the Fantastic Four");

people[10] = **new** Mage("Dr. Strange", 23, "my ability to use the powers of the Sorcere Supreme", "the Avengers");

people[11] = **new** Tank("She-Hulk", 27, "my ability to turn into a She-Monster", "S.H.I.E.L.D");

people[12] = **new** Mage("Zatana", 23, "to use the art of Magic", "Young Justice");

people[13] = **new** Tank("Collosus", 31, "my Metallic Physical Prowess", "X-Men");

people[14] = **new** Gadgeteer("Green Arrow", 25, "my accuracy and unique Bows", "the Justice League");

people[15] = **new** Elementalist("Aquaman", 23, "Controls Ocean and Inhabitants of the Sea", "the Justice League");

people[16] = **new** MartialArtist("Nightwing", 29, "my Acrobatical Physical Skills", "the Justice League");

people[17] = **new** Tank("Hercules", 30, "to have immense Super-Strength", "the Olympians");

people[18] = **new** Mage("Raven", 23, "to use the art of Magic", "Teen Titans");

people[19] = **new** Gadgeteer("Green Lantern", 26, "my Green Ring that can transform to any tools", "the Justice League");

people[20] = **new** Elementalist("Thor", 33, "Controls Mjolnir and the element of Lightning", "the Avengers");

//For Loop to print the polymorphic objects and methods

**for** (**int** x = 0; x < 21; x++) {

System.***out***.println(people[x]);

//This will print out each method for the superhero type

people[x].doThis();

System.***out***.println("\n");

}

}

//This is the class that sorts the superhero using the 'equals' method in each class

**public** **void** sortSuperhero() {

Superhero sortList[] = **new** Superhero[5];

sortList[0] = **new** Mage("Mage", 1, "Mage Powers", "Mages");

sortList[1] = **new** Tank("Tank", 1, "Tank Powers", "Tanks");

sortList[2] = **new** Elementalist("Elementalist", 1, "Elements", "Elementalists");

sortList[3] = **new** Gadgeteer("Gadgeteer", 1, "Gadgets", "Gadgeteers");

sortList[4] = **new** MartialArtist("MartialArtist", 1, "Martial Arts", "Martial Artists");

**for** (**int** x = 0; x < sortList.length; x++) {

//This will print out each 'equals' method for the superhero type

sortList[x].equals();

System.***out***.println("\n");

}

}

}