Lab2 O-O PProgrammin II

1. Polymorphism

Superclass RectangleFigure

1. **public** **class** RectangleFigure {
2. **private** **double** length;
3. **private** **double** width;
5. **public** RectangleFigure(){
6. length = 0;
7. width = 0;
8. }
10. **public** RectangleFigure(**double** l, **double** w) {
11. setDimension(l, w);
12. }
14. **public** **void** setDimension(**double** l, **double** w) {
15. **if**(l >= 0)
16. length = l;
17. **else**
18. length = 0;
20. **if**(w >= 0)
21. width = w;
22. **else**
23. width = 0;
24. }
26. **public** **double** getLength() {
27. **return** length;
28. }
30. **public** **double** getWidth() {
31. **return** width;
32. }
34. **public** **double** area() {
35. **return** length \* width;
36. }
38. **public** **double** perimeter() {
39. **return** 2 \* (length + width);
40. }
42. **public** String toString() {
43. **return** "Length = " + length + "; Width = " + width
44. + "\nArea = " + **this**.area();
45. }
46. }

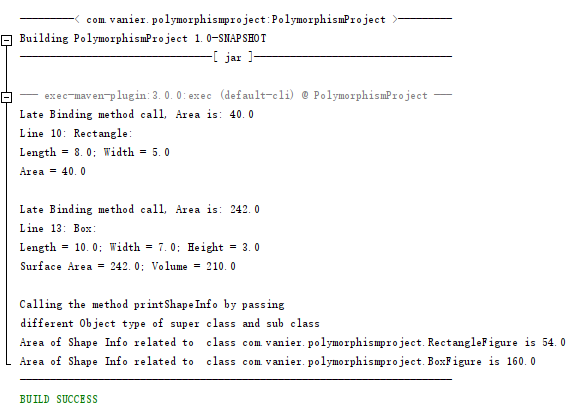
Subclass BoxFigure

1. **public** **class** BoxFigure **extends** RectangleFigure {
2. **private** **double** height;
4. **public** BoxFigure() {
5. **super**();
6. height = 0;
7. }
9. **public** BoxFigure(**double** l, **double** w, **double** h) {
10. **super**(l, w);
11. height = h;
12. }
14. **public** **void** setDimension(**double** l, **double** w, **double** h) {
15. **super**.setDimension(l, w);
17. **if**(h >= 0)
18. height = h;
19. **else**
20. height = 0;
21. }
23. **public** **double** getHeight() {
24. **return** height;
25. }
27. **public** **double** area() {
28. **return** 2 \* (getLength() \* getWidth()
29. + getLength() \* height
30. + getWidth() \* height);
31. }
33. **public** **double** volume() {
34. **return** **super**.area() \* height;
35. }
37. **public** String toString() {
38. **return** "Length = " + **super**.getLength() + "; Width = " + **super**.getWidth()
39. + "; Height = " + height
40. + "\nSurface Area = " + **this**.area()
41. + "; Volume = " + **this**.volume();
42. }
43. }

Java Program TestPolymorphism.java

1. **public** **class** PolymorphismProject {
3. **public** **static** **void** main(String[] args) {                  //Line 3
4. //Line 4
5. RectangleFigure rectangle, shapeRef;                  //Line 5
6. BoxFigure box;                                     //Line 6
8. rectangle = **new** RectangleFigure(8, 5);                //Line 7
9. box = **new** BoxFigure(10, 7, 3);                         //Line 8
11. shapeRef = rectangle;                                  //Line 9
13. System.out.println("Late Binding method call, Area is: "
14. + shapeRef.area());
16. System.out.println("Line 10: Rectangle:\n"
17. + shapeRef);                          //Line 10
18. System.out.println();                                 //Line 11
20. shapeRef = box;                                       //Line 12
22. System.out.println("Late Binding method call, Area is: "
23. + shapeRef.area());
25. System.out.println("Line 13: Box:\n"
26. + shapeRef);
27. System.out.println();
29. //Calling the method printShapeInfo by passing
30. //  different Object type as argument
31. //  to Increase Reusablity pattern
32. System.out.println("Calling the method printShapeInfo by passing\n"
33. + "different Object type of super class and sub class");
34. RectangleFigure myRectangle2 = **new** RectangleFigure(9, 6);
35. BoxFigure myBox2 = **new** BoxFigure(12, 4, 2);
37. shapeRef = myRectangle2;
38. printShapeInfo(shapeRef);
40. shapeRef = myBox2;
41. printShapeInfo(shapeRef);
42. }   // end of main                                          //Line 15
44. **public** **static** **void** printShapeInfo(RectangleFigure shapeObj) {
45. System.out.println("Area of Shape Info related to  " + shapeObj.getClass()
46. + " is " + shapeObj.area());
48. }
49. }   // End of class                                          //Line 16

Output screenshots



1. Complete Project SportProject from Lab 1:

Java Program Sport.java

1. **public** **class** Sport {
2. **private** String name;
3. **private** **double** number\_hour;
4. **private** **int** number\_week;
6. **public** **static** **double** cost\_hour;
7. **public** **double** cost\_training;
9. **public** Sport() {
10. name = "";
11. number\_hour = 0.00;
12. number\_week = 0;
13. }
14. **public** Sport(String n, **double** h, **int** w) {
15. name = n;
16. number\_hour = h;
17. number\_week = w;
18. }
20. **public** **boolean** setName(String n) {
21. **try** {
22. name = n;
23. **return** **true**;
24. } **catch**(Exception e) {
25. **return** **false**;
26. }
27. }
29. **public** **boolean** setNumberHour(**double** hour) {
30. **try** {
31. number\_hour = hour;
32. **return** **true**;
33. } **catch**(Exception e) {
34. **return** **false**;
35. }
36. }
38. **public** **boolean** setNumberWeek(**int** week) {
39. **try** {
40. number\_week = week;
41. **return** **true**;
42. } **catch**(Exception e) {
43. **return** **false**;
44. }
45. }
47. **public** String getName() {
48. **return** name;
49. }
51. **public** **double** getNumberHour() {
52. **return** number\_hour;
53. }
55. **public** **int** getNumberWeek() {
56. **return** number\_week;
57. }
59. **public** **double** CalculateCostTraining() {
60. cost\_training = cost\_hour \* number\_hour \* number\_week;
61. **return** cost\_training;
62. }
64. **public** **double** CalculateCostTraining(**double** costHour) {
65. cost\_hour = costHour;
66. cost\_training = cost\_hour \* number\_hour \* number\_week;
67. **return** cost\_training;
68. }
70. **public** String toString() {
71. **return** name + "//"
72. + String.format("%.2f", number\_hour) + "//"
73. + String.format("%d", number\_week) + "//"
74. + String.format("%.2f$", cost\_hour);
75. }
76. }

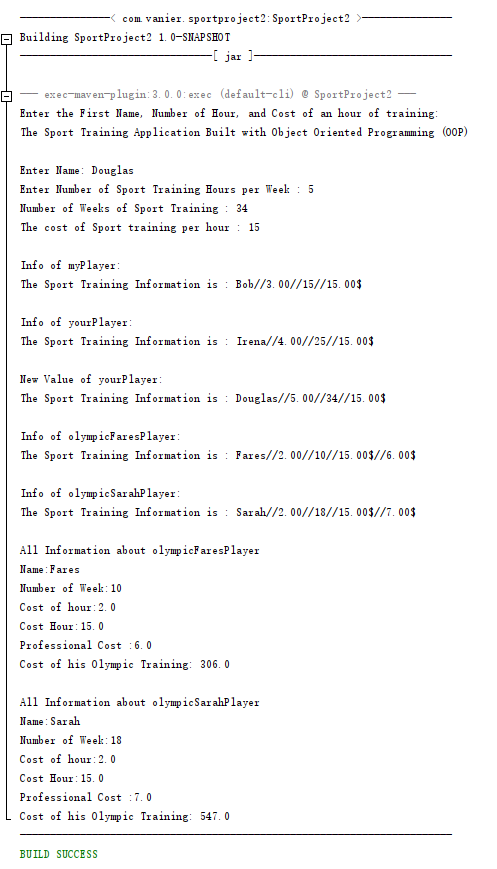
Java Program OlympicSport.java

1. **public** **class** OlympicSport **extends** Sport{
2. **private** **double** cost\_pro;
4. **public** OlympicSport(String n, **double** numberHour, **int** numberWeek, **double** costPro){
5. **super**(n, numberHour, numberWeek);
6. cost\_pro = costPro;
7. }
9. **public** **double** getCostPro() {
10. **return** cost\_pro;
11. }
13. **public** **void** setCostPro(**double** costPro) {
14. **if**(costPro>0) {
15. cost\_pro = costPro;
16. } **else** {
17. cost\_pro = 0.0;
18. }
19. }
21. **public** **double** CalculateCostTraining() {
22. cost\_training = **super**.CalculateCostTraining() + cost\_pro;
23. **return** cost\_training;
24. }
26. **public** **double** CalculateCostTraining(**double** costHour) {
27. cost\_training = **super**.CalculateCostTraining(costHour) + cost\_pro;
28. **return** cost\_training;
29. }
31. **public** String toString() {
32. **return** **super**.toString() + "//"
33. + String.format("%.2f$", cost\_pro);
34. }
35. }

Java Program SportProject2.java

1. **public** **class** SportProject2 {
3. **public** **static** **void** main(String[] args) {
4. Scanner input = **new** Scanner(System.in);
5. String name;
6. **double** number\_hour;
7. **int** number\_week;
8. **int** cost\_hour;
10. Sport myPlayer = **new** Sport("Bob", 3, 15);
11. Sport yourPlayer = **new** Sport("Irena", 4, 25);
13. System.out.println("Enter the First Name, Number of Hour, and Cost of an hour of training:");
14. System.out.println("The Sport Training Application Built with Object Oriented Programming (OOP)");
16. System.out.printf("\nEnter Name: ");
17. name = input.next();
18. System.out.printf("Enter Number of Sport Training Hours per Week : ");
19. number\_hour = input.nextDouble();
20. System.out.printf("Number of Weeks of Sport Training : ");
21. number\_week = input.nextInt();
22. System.out.printf("The cost of Sport training per hour : ");
23. cost\_hour = input.nextInt();
25. myPlayer.cost\_hour = cost\_hour;
26. System.out.println();
27. System.out.println("Info of myPlayer:");
28. System.out.println("The Sport Training Information is : " + myPlayer);
30. System.out.println();
31. System.out.println("Info of yourPlayer:");
32. System.out.println("The Sport Training Information is : " + yourPlayer);
34. yourPlayer.setName(name);
35. yourPlayer.setNumberHour(number\_hour);
36. yourPlayer.setNumberWeek(number\_week);
37. System.out.println();
38. System.out.println("New Value of yourPlayer:");
39. System.out.println("The Sport Training Information is : " + yourPlayer);
41. OlympicSport olympicFaresPlayer = **new** OlympicSport("Fares", 2, 10, 6);
42. System.out.println();
43. System.out.println("Info of olympicFaresPlayer:");
44. System.out.println("The Sport Training Information is : " + olympicFaresPlayer);
46. OlympicSport olympicSarahPlayer = **new** OlympicSport("Sarah", 2, 18, 7);
47. System.out.println();
48. System.out.println("Info of olympicSarahPlayer:");
49. System.out.println("The Sport Training Information is : " + olympicSarahPlayer);
51. System.out.println();
52. System.out.println("All Information about olympicFaresPlayer");
53. System.out.println("Name:" + olympicFaresPlayer.getName());
54. System.out.println("Number of Week:" + olympicFaresPlayer.getNumberWeek());
55. System.out.println("Cost of hour:" + olympicFaresPlayer.getNumberHour());
56. System.out.println("Cost Hour:" + olympicFaresPlayer.cost\_hour);
57. System.out.println("Professional Cost :" + olympicFaresPlayer.getCostPro());
58. System.out.println("Cost of his Olympic Training: " + olympicFaresPlayer.CalculateCostTraining());
60. System.out.println();
61. System.out.println("All Information about olympicSarahPlayer");
62. System.out.println("Name:" + olympicSarahPlayer.getName());
63. System.out.println("Number of Week:" + olympicSarahPlayer.getNumberWeek());
64. System.out.println("Cost of hour:" + olympicSarahPlayer.getNumberHour());
65. System.out.println("Cost Hour:" + olympicSarahPlayer.cost\_hour);
66. System.out.println("Professional Cost :" + olympicSarahPlayer.getCostPro());
67. System.out.println("Cost of his Olympic Training: " + olympicSarahPlayer.CalculateCostTraining());
69. }
70. }

Output screenshots



1. Answer the following questions:

a) Give an example of super class and sub class in your own stated project (different than

Geometry, Sport projects).

**Answer:**

Super class Products

Sub class Books extends Products

b) Define then data attributes of the specified super class. (give at least two data attributes)

**Answer:**

Super class Products

int product\_ID;

String product\_name;

int stock;

double price;

c) Define then data attributes of the specified sub class. (give at least one data attribute)

**Answer:**

Sub class Books extends Products

String author;

String publisher;

d) Define then methods members of the specified super class. (at least one method)

**Answer:**

Super class Products

public boolean addCart(int number) { } // add number products to user’s cart

public void inStock(int number) { } // restock products in the stock

public void showDetail() { // display the detail of product

System.out.println(product\_name);

}

e) Define then an overriding methods member of the specified sub class. (at least one method)

**Answer:**

Sub class Books

@Override

public void showDetail() { // display the detail of book

System.out.println(product\_name);

System.out.println(author);

System.out.println(publisher);

}

f) Write then a Java statement to instantiate an object from super class

**Answer:**

Products newproduct = new Products();

g) Write then a Java statement to instantiate an object from sub class

**Answer:**

Books newbook = new Books();

h) Apply polymorphism with the super class object from question (f) to invoke the overriding method of sub class defined in question (e). Write then the appropriate Java statements on how to use polymorphism.

**Answer:**

Products newproduct = new Products();

Books newbook = new Books();

newproduct.showDetail();

newbook.showDetail();

4. True or False and why:

a) It is Valid to write the following assignment in PolymorphismProject:

box = shapeRef;

**Answer:**

No.

Because the scenario that occurs the super class and sub class with inheritance relationship (is a), in Java, in addition to the reference of the class pointing to the object of this class, then the reference of the parent class can be made to point to the object of the sub class, but the reference of the subclass cannot be made to point to the super class Object

b) It is Valid to write the following statements in PolymorphismProject:

RectangleFigure shapeRef;

shapeRef =rectangle2; //assume rectangle2 is an object of RectangleFigure class type

System.out.println("Late Binding method call, Volume is:" + shapeRef.volume());

**Answer:**

No.

Because class RectangleFigure has no method volume().

c) It is Valid to write the following statements in PolymorphismProject:

RectangleFigure shapeRef;

shapeRef =box2; // assume box2 is an object of BoxFigure class type

System.out.println("Late Binding method call, Volume is:" + shapeRef.volume());

**Answer:**

Yes.

Because the reference of the parent class can be made to point to the object of the sub class, box2 produced from is the sub class BoxFigure, its super class is RectangleFigure

d) Assume you have the method called printShapeInfo(RectangleFigure shapeObj): is thefollowing call valid:

printShapeInfo(rectangle2); //rectangle2 is an object of RectangleFigure class type

**Answer:**

Yes.

The object instantiated from class can use the method of this class.

e) Assume you have the method called printShapeInfo(BoxFigure shapeObj): is the following call valid:

printShapeInfo(rectangle2); //rectangle2 is an object of RectangleFigure class type

**Answer:**

No.

Because the object instantiated from super class cannot use the method of sub class before referenced.

f) Assume you have the method called printShapeInfo(RectangleFigure shapeObj): is the

following call valid:

printShapeInfo(box2); // box2 is an object of BoxFigure class type

**Answer:**

Yes.

Because the object instantiated from sub class can use the method of super class.