# Удамшил ба хийсвэрлэл (Лаборатори №8)

Я.Тэмүүжин

ХШУИС, МКУТ, ПХ 16B1SEAS3369, y.usott@gmail.com

# 1. ОРШИЛ

Объект хандалтат програмчлалын удамшил, хийсвэрлэл(abstraction) ойлголтыг судлаж тэдгээрийг ашиглан лаборатори 8-д хийсэн кодонд хэрэгжүүлнэ.

# 2. ЗОРИЛГО

Объект хандалтат програмчлалын удамшил, хийсвэрлэл(abstraction) ойлголтыг ашиглан лаборатори 8-д хийсэн кодонд хэрэгжүүлж гүйцэтгэнэ.

Үүний тулд дараах зорилтуудыг тавьж ажиллана.

1. Java болон C++ хэлэнд удамшил, хийсвэрлэлтийг хэрхэн хийдэг талаар онолын судалгаа хийх.
2. Хийсвэр функц, хийсвэр классын онолын судалгааг хийх
3. Санамсаргүй дүрс үүсгэх кодыг бичих
4. Санамсаргүй үүссэн дүрсүүдийг талбайгаар нь эрэмбэлэх код бичих

# 3. ОНОЛЫН СУДАЛГАА

## Хийсвэр функц гэж юу вэ? Тодорхойлолт, зарлалт, жишээ, давуу талыг тус тус бич.

Жава дээр функцүүд нь хийсвэр байдаг. Эх классд тодорхойлолт нь хийгдсэн ба хүүхэд класс эх классын тодорхойлолтыг ашиглаж болох ба эсвэл дахин программчилж ашиглаж болно.

**abstract** **<буцаахТөрөл>** <функцынНэр>();/

## Жинхэнэ хийсвэр функц гэж юу вэ? Тодорхойлолт, зарлалт, жишээ, давуу талыг тус тус бич.

Abstract түлхүүр үг ашиглаж тодорхойлолтгүй зарлагдсан функцыг жинхэнэ хийсвэр функц гэнэ.Удамшсан класс жинхэнэ хийсвэр классыг заавал дахин програмчлах ёстой. Хэрвээ охин классд зарлагдсан байх шаардлагатай функцүүдийг зарлаж өгөх мөн түүний охин классууд түүний тодорхойлолтыг өөр өөрийн хүссэнэээр тодорхойлж өгөхийг хүсвэл хийсвэр функцийг ашиглаж болно. Жишээ нь тус лабораторын ажилд хийсэн TwoDimensionalShape.java дотрох calcArea() функц нь охин классууд буюу удамшсан классууд болох triangle, circle, square дотор байх ёстой боловч тэдгээрийн тодорхойлолтууд нь өөр өөр байх хэрэгтэй болно. Ийм учраас түүнийг хийсвэр функц болгон ашигласан.

Хийсвэр функц зарлах:

**abstract** **<буцаахТөрөл>** <функцынНэр>();/

Жишээ нь: Дээр дурдсан TwoDimensionalShape.java дээр ажиллъя тус хийсвэр классд calcArea() хийсвэр функцыг зарлаж өгсөн байна

public abstract class TwoDimensionalShape extends Shape {

protected double x[],y[] ; // 2 hemjeesin x bolon y coordinate

public abstract double calcArea();

}

Харин түүнээс удамшсан circle класс calcArea() –г ингэж тодорхойлсон

@Override

public double calcArea() {

return pi \* Math.pow(this.length,2);

}

Харин square класс

@Override

public double calcArea() {

return Math.pow(this.length,2);

}

Гэх зэргээр өөр өөрсдийн хэрэгцээ шаардлагад нь тохирсон тодорхойлолтуудыг хийж өгдөг.

## Функц дахин программчлах гэж юу вэ? Эх классын дахин программчилсан функцыг хүүхэд классын функц дотроос хэрхэн дууддаг вэ?

Эх классаас удамшиж орж ирсэн классын тодорхойлолтыг өөрчлөн шинээр програмчлахыг функц дахин програмчлах гэнэ. Эх классын дахин програмчилсан функцыг superо обьектыг ашиглаж дууддаг. Жишээ нь эх класс calcArea() функцыг охин класс дахин програмчилсан бөгөөд эх классын calcArea()-г дуудхыг хүсвэл super.calcArea() гэж дуудна. Харин эх классын байгуулагчыг дуудхыг хүсвэл super() эсвэл super(a,x) болно.

## 3.4 Хийсвэр класс гэж юу вэ? Хэрхэн объект байгуулдаг вэ?

Abstract түлхүр үг ашиглаж үүсгэсэн классыг хийсвэр класс гэнэ. Хийсвэр класс нь хийсвэр функцүүдийг агуулж болно. Классд нэг л жинхэнэ хийсвэр функц байвал түүнийг хийсвэр класс болгох хэрэгтэй байдаг. Хийсвэр классаас обьект үүсгэж болдоггүй бөгөөд хийсвэр классыг ашиглахын тулд түүнийг өөр классд удамшуулж хийсвэр функцүүдийг дахин програмчилж ашиглана. Хийсвэр классаас удамшсан бол түүний бүх функцийг дахин программчилж өгөх ёстой байдаг.

## Удамшилд байгуулагч функц хэрхэн ашиглагддаг вэ? Удамшил ба байгуулагч функцийн хүрээнд үзсэн зүйлүүдийг бич.

Удашсан классаас обьект үүсхэд эхлээд түүний эх классын байгуулагч функ түүний дараа өөрийн байгуулагч функц нь дуудагддаг.

Жишээ нь shape.java классын анхдагч байгуулагч нь

1. Shape(){
2. setName("durs");
3. setLength(1);
4. *//Эх класс түрүүлж дуудаж байгаа эсэхийг шалгахын тулд тавьсан*
5. ***System.out.println("eh class");***
6. }

Бөгөөд түүнээс TwoDimensionalShape удамшина түүнээс удамшсан square классын обьект үүсхэд эхлээд eh class гэж хэвлэнэ. Энэ процесс нь эх классын байгуулагч түрүүлж дуудагдаж байгаа харуулж байна.

## 3.6 Удамшил ба устгагч функц хоёр ямар хамааралтай вэ?

Байгуулагч функц нь эхлээд эх классынх түрүүлж дуудагдаг дараа нь охин классын байгуулагч дуудагдаж байсан бол устгагч функцын хувьд эсэргээрээ байна. Тэгэхлээр эхлээд охин классын устгагч дуудагдаж үүний дараа эх классын устгагч дуудагддаг байна.

## 4. Хэрэгжүүлэлт

## 4.1 Массивт санамсаргүй дүрсийг санамсаргүй талбайтайгаар үүсгэж байна

1. TwoDimensionalShape[] shape = **new** TwoDimensionalShape[10];
3. **for**(**int** i=0;i<10;i++){
5. **int randomNumber = (int) (Math.random()\*3+1);**
6. **double** rLength = Math.random()\*50+1;
7. **switch**(randomNumber){
8. **case** 1:
9. **shape[i] = new Circle(1,1,rLength,"Circle-"+(int)rLength);break;**
10. **case** 2:
11. shape[i] = **new** Triangle(1,1,rLength,"Triangle-" + (**int**)rLength); **break**;
12. **case** 3:
13. shape[i] = **new** Square(1,1,rLength,"Square-" + (**int**)rLength); **break**;
14. **}**
15. }

## 4.2 Санамсаргүй үүсгэсэн дүрснүүдийг талбайгаар нь өсөх дарааллаар эрэмбэлж байна

1. **for** (**int** i = 0; i < 10 - 1; i++)
2. {
3. **int** index = i;
4. **for** (**int** j = i + 1; j < shape.length; j++){
5. **if (shape[j].calcArea() < shape[index].calcArea()){**
6. index = j;*//searching for lowest index*
7. }
8. }
9. TwoDimensionalShape smaller = shape[index];
10. **shape[index] = shape[i];**
11. shape[i] = smaller;
12. }

## 4.3 Хэвлэх давталт

1. **for**(**int** i=0;i<10;i++){
2. System.out.println(shape[i].getName()+" "+shape[i].calcArea()+"**\n**");
3. **if**(shape[i] **instanceof** Triangle){
4. *// System.out.println(((Triangle) shape[i]).getX1()+" "+shape[i].calcArea()+"\n");*
5. **}**
6. }

## 7. Хавсралт

## 7.1 Shape class (Shape.java)

1. **package** lab8;
2. **abstract** **class** Shape{
3. *//дүрсний нэрийг хадгалах гишүүн өгөгдөл*
4. **protected** String name;
5. ***//дүрсний нэрийг хадгалах гишүүн өгөгдөл***
6. **protected** **double** length;
7. **public** **abstract** **double** calcPerimeter();
9. Shape(){
10. **setName("durs");**
11. setLength(1);
12. *//Эх класс түрүүлж дуудаж байгаа эсэхийг шалгахын тулд тавьсан*
13. *// System.out.println("eh class");*
14. }
16. Shape(**double** length,String name){
17. setName(name);
18. setLength(length);
19. }
21. *// Дүрсэнд нэр олгох функц*
22. **public** **void** setName(String name) {
23. **this**.name = name;
24. }
26. *// Дүрсний нэрийг буцаах функц*
27. **public** String getName() {
28. **return** **this**.name;
29. }
31. *// Дүрсний талын уртад утга оноох функц*
32. **public** **void** setLength(**double** length){
33. **this**.length = length;
34. }
36. *// Дүрсний талын уртыг буцаах*
37. **public** **double** getLength(){
38. **return** **this**.length;
39. }
41. }

## 7.2 TwoDimentional class(TwoDimentional.java)

1. **package** lab8;
2. **public** **abstract** **class** TwoDimensionalShape **extends** Shape {
3. **protected** **double** x[],y[] ; *// 2 hemjeesin x bolon y coordinate*
4. **public** **abstract** **double** calcArea();
6. }

## 7.3 Гурвалжин класс (Triangle.java)

1. **package** lab8;
3. **import** java.lang.Math;
5. **public class Triangle extends TwoDimensionalShape {**

8. **public** Triangle(**double** x,**double** y,**double** length,String name)
9. **{**
10. **X= new double[3];**
11. **Y= new double[3];**
12. setName(name);
13. setLength(length);
14. setX(x);
15. setY(y);
16. **setX1();**
17. setY1();
18. setX2();
19. SetY2();
20. }

23. *// tuhain dursni x coordinated utga onoo function*
24. **public** **void** setX(**double** x)
25. {
26. **this.x[0] = x;**
27. }
29. *//tuhain dursni y coordinated utga onoo function*
30. **public** **void** setY(**double** y)
31. **{**
32. **this**.y[0] = y;
33. }
35. *//tuhain dursni x coordinatin utgig butsaah function*
36. **public double getX()**
37. {
38. **return** **this**.x[0];
39. }
41. ***//tuhain dursni y coordinatin utgig butsaah function***
42. **public** **double** getY()
43. {
44. **return** **this**.y[0];
45. }
46. *// Zuun dood oroin X coordinatad utga onooh*
47. **public** **void** setX1()
48. **{**
49. **this**.x[1] = **this**.x[0] - (**this**.length)/2;
50. }
52. *// Zuun dood oroin X coordinatig tootsooloh*
53. **public double getX1()**
54. {
55. **return** **this**.x[1];
56. }
58. ***// Zuun dood oroin X coordinatad utga onooh***
59. **public** **void** setY1()
60. {
61. **double** h;
62. h = Math.pow(**this**.length, 2) - Math.pow((**this**.length/2), 2);
63. **h = Math.sqrt(h);**
64. **this**.y[1] = **this**.y[0] - h;
65. }
67. *// Zuun dood oroin X coordinatig tootsooloh*
68. **public double getY1()**
69. {
70. **return** **this**.y[1];
71. }
73. ***// baruun dood oroin X coordinatad utga onooh***
74. **public** **void** setX2()
75. {
76. **this**.x[2] = **this**.x[0] + (**this**.length)/2;
77. }
79. *// baruun dood oroin X coordinatig tootsooloh*
80. **public** **double** getX2()
81. {
82. **return** **this**.x[2];
83. **}**
85. *// Baruun dood oroin y coordinatig tootsooloh*
86. **public** **void** SetY2()
87. {
88. **double h;**
89. h = Math.pow(**this**.length, 2) - Math.pow((**this**.length/2), 2);
90. h = Math.sqrt(h);
91. **this**.y[2] = **this**.y[0] - h;
92. }
94. *// baruun dood oroin Y coordinatig tootsooloh*
95. **public** **double** getY2()
96. {
97. **return** **this**.y[2];
98. **}**
100. @Override
101. **public** **double** calcArea() {
102. **double** x;
103. **x = Math.toRadians(60);**
104. **double** area = (Math.pow(**this**.length, 2) \* Math.sin(x))/2;
105. **return** area;
106. }
108. **@Override**
109. **public** **double** calcPerimeter() {
110. **return** 3 \* **this**.length;
111. }

114. }

## 7.4 Квадрат класс (Square.java)

1. **package** lab8;
3. **public** **class** Square **extends** TwoDimensionalShape {

6. *// Parametertai baiguulach punkts*
7. **public** Square(**double** x,**double** y, **double** length,String name)
8. **{**
9. **X= new double[4];**
10. **Y= new double[4];**
11. setName(name);
12. setLength(length);
13. setX(x);
14. setY(y);
15. **setX1();**
16. setY1();
17. setX2();
18. setY2();
19. setX3();
20. **setY3();**
21. }
23. *// tuhain dursni x coordinated utga onoo function*
24. **public** **void** setX(**double** x)
25. **{**
26. **this**.x[0] = x;
27. }
29. *//tuhain dursni y coordinated utga onoo function*
30. **public void setY(double y)**
31. {
32. **this**.y[0] = y;
33. }
35. ***//tuhain dursni x coordinatin utgig butsaah function***
36. **public** **double** getX(){
37. **return** **this**.x[0];
38. }
40. ***//tuhain dursni y coordinatin utgig butsaah function***
41. **public** **double** getY(){
42. **return** **this**.y[0];
43. }
45. ***// Baruun deed oroin X coordinatig utga onooh***
46. **public** **void** setX1()
47. {
48. **this**.x[1] = **this**.x[0] + **this**.length;
49. }
51. *// baruun deed oroin x coordinatig butsaah*
52. **public** **double** getX1()
53. {
54. **return** **this**.x[1];
55. **}**
57. *// baruun deed oroin Y coordinatad utga onooh*
58. **public** **void** setY1()
59. {
60. **this.y[1] = this.y[0];**
61. }
63. *// baruun deed oroin y coordinatig butsaah*
64. **public** **double** getY1()
65. **{**
66. **return** **this**.y[1];
67. }

70. ***// baruun dood oroin X coordinatid utga onooh***
71. **public** **void** setX2()
72. {
73. **this**.x[2] = **this**.x[0] + **this**.length;
74. }
76. *// baruun dood oroin x coordinatig butsaah*
77. **public** **double** getX2()
78. {
79. **return** **this**.x[2];
80. **}**
82. *// Baruun dood Y coordinatad utga onooh*
83. **public** **void** setY2()
84. {
85. **this.y[2] = this.y[0] - this.length;**
86. }
88. *// baruun dood oroin y coordinatig butsaah*
89. **public** **double** getY2()
90. **{**
91. **return** **this**.y[2];
92. }
94. *// zuun dood oroin Y coordinatad utga onooh*
95. **public void setY3()**
96. {
97. **this**.y[3] = **this**.y[0] - **this**.length;
98. }
100. ***// zuun dood oroin x coordinatig butsaah***
101. **public** **double** getY3()
102. {
103. **return** **this**.y[3];
104. }
106. *// Zuun dood oroin X coordinatig utga onooh*
107. **public** **void** setX3()
108. {
109. **this**.x[3] = **this**.x[0] ;
110. **}**
112. *// zuun dood oroin x coordinatig butsaah*
113. **public** **double** getX3()
114. {
115. **return this.x[3];**
116. }
117. */\**
118. *\* Талбай олох функцыг дахин тодорхойлж байна*
119. *\*/*
120. **@Override**
121. **public** **double** calcArea() {
122. **return** Math.pow(**this**.length,2);
123. }
124. */\**
125. ***\* Периметер олох функцыг дахин тодорхойлж байна***
126. *\*/*
127. @Override
128. **public** **double** calcPerimeter() {
129. **return** 4 \* **this**.length;
130. **}**

133. }

## 7.5 Дугуй (Circle.java)

1. **package** lab8;
2. **public** **class** Circle **extends** TwoDimensionalShape {
3. **double pi = 3.14;**
4. *// Duguin baiguulach puntkts*
5. **public** Circle(**double** x,**double** y,**double** length,String name)
6. {
7. **X= new double[4];**
8. **Y= new double[4];**
9. setX(x);
10. **setY(y);**
11. setLength(length);
12. setName(name);
13. }


17. *// tuhain dursni x coordinated utga onoo function*
18. **public** **void** setX(**double** x)
19. {
20. **this.x[0] = x;**
21. }
23. *//tuhain dursni y coordinated utga onoo function*
24. **public** **void** setY(**double** y)
25. **{**
26. **this**.y[0] = y;
27. }
29. *//tuhain dursni x coordinatin utgig butsaah function*
30. **public double getX()**
31. {
32. **return** **this**.x[0];
33. }
35. ***//tuhain dursni y coordinatin utgig butsaah function***
36. **public** **double** getY()
37. {
38. **return** **this**.y[0];
39. }
41. @Override
42. **public** **double** calcArea() {
43. **return** pi \* Math.pow(**this**.length,2);
44. }
46. @Override
47. **public** **double** calcPerimeter() {
48. **return** 2 \* pi \* **this**.length;
49. }


53. }

7.6 Main (Main.java)

1. **package** lab8;
2. **public** **class** MainShape {
3. **public static void main(String[] args) {**
4. TwoDimensionalShape[] shape = **new** TwoDimensionalShape[10];
5. **for**(**int** i=0;i<10;i++){
6. **int** randomNumber = (**int**) (Math.random()\*3+1);
7. **double** rLength = Math.random()\*50+1;
8. **switch**(randomNumber){
9. **case 1:**
10. shape[i] = **new** Circle(1,1,rLength,"Circle-"+(**int**)rLength);**break**;
11. ё**ase** 2:
12. shape[i] = **new** Triangle(1,1,rLength,"Triangle-" + (**int**)rLength); **break**;
13. **case** 3:
14. **shape[i] = new Square(1,1,rLength,"Square-" + (int)rLength); break;**
15. }
16. }
18. **for** (**int** i = 0; i < 10 - 1; i++)
19. **{**
20. **int** index = i;
21. **for** (**int** j = i + 1; j < shape.length; j++){
22. **if** (shape[j].calcArea() < shape[index].calcArea()){
23. index = j;*//searching for lowest index*
24. **}**
25. }
26. TwoDimensionalShape smaller = shape[index];
27. shape[index] = shape[i];
28. shape[i] = smaller;
29. **}**
31. **for**(**int** i=0;i<10;i++){
32. System.out.println(shape[i].getName()+" "+shape[i].calcArea()+"**\n**");
33. **if**(shape[i] **instanceof** Triangle){
34. ***// System.out.println(((Triangle) shape[i]).getX1()+" "+shape[i].calcArea()+"\n");***
35. }
36. }
38. }


42. }