# Удамшил Ба Түүний Хэрэглээ (Лаборатори №6)

О. Ихбаяр

ХШУИС, Програм хангамж, 3-р түвшин, 17b1num2575@stud.num.edu.mn

# 1. ОРШИЛ

Удамшил гэж юу болох, түүний хэрэглээ болон үүргүүдийн талаар авч үзэх болно. Мөн удамшлын давуу тал болон төрлүүдийг нь судална.

# 2. ЗОРИЛГО

Удамшлын талаарх ерөнхий болон онолын ойлголтыг мэдэх ба удамшлыг зөв ашиглаж сурах болно.

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Удамшил

Удамшил гэдэг нь шинээр үүсэж буй класс нь өмнөх классынхаа шинж үйл хөдлөлийг өвлөн авахыг удамшил гэнэ. Өөрөөр хэлбэл байгаа классаас шинээр класс үүсгэх ойлголт юм. Шинээр үүсэж буй класс нь өмнөх классынхаа зарим эсвэл бүх шинжийг өвлөн авдаг. Мөн өөрийн шинжийг нэмж тодорхойлж өгч болно. Өөрөөс нь шинэ класс үүсэж байгаа классыг үндсэн, эх класс, шинээр үүсэх классыг удамшсан класс хэмээн нэрлэдэг. Удамшлын давуу тал нь нэгэнт бичигдсэн кодыг дахин ашиглах болон ерөнхийлж, нарийвчилж өгдөгөөрөө давуу талтай.

JAVA хэлний хувьд нийлмэл удамшил байдаггүй, удамшлын горим нь public байдаг бөгөөд устгагч функц болон delete оператор байхгүй ба тухайн санах ойгоос хамгийн сүүлийн заалтан хувьсагч арилах үед санах ойг чөлөөлдөг.

## 3.2 Удамшлын горим

Удамших класст ямар нэгэн горим зааж өгөөгүй үед private горимоор удамшдаг. Удамшлын гурван төрлийн горим байдаг. Үүнд:

* PUBLIC горимоор удамших үед эх классын гишүүн өгөгдөл, гишүүн функц хүүхэд класст хандалтын түвшин нь өөрчлөгдөхгүй.
* PRIVATE горимоор удамших үед эх классын гишүүн өгөгдөл, гишүүн функц хүүхэд класст хандалтын түвшин нь private болно.
* PROTECTED горимоор удамших үед эх классын гишүүн өгөгдөл, гишүүн функц хүүхэд класст хандалтын түвшин нь protected болж өөрчлөгдөнө.

## 3.3 Удамшлын төрөл

Удамшлын таван төрөл байдаг. Үүнд:

1. Нийлмэл удамшил – Удамших класс нь хоёроос цөөнгүй эх класстай байна. Тодорхойлохдоо “class + шинэ классын нэр : удамшлын горим + эх класс + удамшлын горим + эх класс” гэж тодорхойлно.
2. Энгийн удамшил – Удамшиж буй класс нь зөвхөн нэг эх класстай байна. Тодорхойлохдоо “class + шинэ классын нэр : удамшлын горим + эх класс” гэж тодорхойлно.
3. Олон түвшинт удамшил – Удамшиж буй классын эх класс нь өөр классаас удамшсан байвал олон түвшинт удамшил болно.
4. Шаталсан удамшил – Нэг эх классаас хоёроос цөөнгүй класс удамшсан байвал шаталсан удамшил болно.
5. Холимог удамшил – Шаталсан удамшил болон нийлмэл удамшил хоёрын нэгдлээс холимог удамшил үүснэ.

# 4. ХЭРЭГЖҮҮЛЭЛТ

## 4.1 Класс удамших

class shape {

public:

int \*y;

int \*x;

Тодорхойлолт: Shape классаас shape2d классыг удамшуулж байна. Ингэхдээ х, у-г shape2d класст дахин тодорхойлж өгөх шаардлагагүй. Учир нь shape2d класст shape-ийн х, у адил төрөлтэйгөөр ашиглана гэсэн үг.

};

// shapees 2d shape udamshih

class shape2d:public shape {

// a ni taliin urt

public:

int a;

float area();

// premetr

int pr();

};

## 4.3 Холимог удамшил ашиглалт

class shape {

public:

int \*y;

int \*x;

};

// shapees 2d shape udamshih

class shape2d:public shape {

// a ni taliin urt

public:

Тайлбар: Shape классаас shape2d класс удамшиж байгаа бөгөөд shape2d класаас Square, Circle, … гэх мэт классууд удамшиж байгаа нь холимог удамшлыг үүсгэж байна. Доорх жишээ нь холимог удашлаас үүссэн хүүхэд классуудад утга оноож өгөөд бодолт хийсэн жишээ юм.

int a;

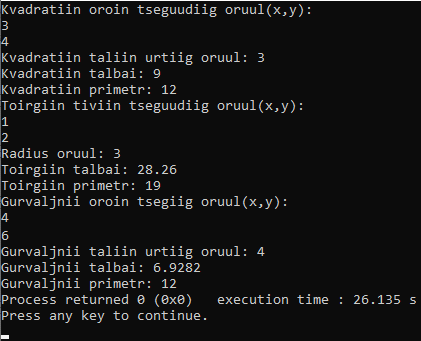
float area();

// premetr

int pr();

};

class Square:public shape2d{

 public:

int \*x = new int[4];

int \*y = new int[4];

float area();

int pr();

};

class Circle : public shape2d {

public:

int x;

int y;

float area();

int pr();

};

# 5. ДҮГНЭЛТ

Энэхүү лабораторын ажиллаар удамшил гэж юу болох, хэрхэн хэрэглэх талаар болон удамшил ашиласны давуу талууд болон үүрэг, төрөл болон удамшлын горимуудын талаарх мэдлэгийг авсан.

# 6. АШИГЛАСАН МАТЕРИАЛ

1. Объект хандлагат технологийн С++ програмчлал, Ж.Пүрэв, 2008, Улаанбаатар.

# 7. ХАВСРАЛТ

#include<iostream>

#include<math.h>

using namespace std;

class shape {

public:

int \*y;

int \*x;

};

// shapees 2d shape udamshih

class shape2d:public shape {

// a ni taliin urt

public:

int a;

float area();

// premetr

int pr();

};

int shape2d::pr(){

return 0;

}

float shape2d::area(){

return 0;

}

class Square:public shape2d{

public:

\*x = new int[4];

\*y = new int[4];

float area();

int pr();

};

float Square::area(){

return a\*a;

}

int Square::pr(){

return a\*4;

}

class Circle : public shape2d {

public:

int x;

int y;

float area();

int pr();

};

float Circle::area(){

return 3.14\*a\*a;

}

int Circle::pr(){

return 2\*3.24\*a;

}

class Triangle : public shape2d {

public:

int \*x = new int[3];

int \*y = new int[3];

float area();

int pr();

};

float Triangle::area(){

return (a\*a\*sqrt(3))/4;

}

int Triangle::pr(){

return a\*3;

}

main(){

/\*Kvadratiin oroin tseg bolon taliin urtiig garaas oruulan busad

oroinuud bolon primetr talbaig bodoj olood hevlej bna\*/

cout<<"Kvadratiin oroin tseguudiig oruul(x,y): \n";

Square s1;

Circle c1;

Triangle t1;

cin>>s1.x[0]>>s1.y[0];

cout<<"Kvadratiin taliin urtiig oruul: ";

cin>>s1.a;

//end kvadratiin busad oroig olj bna

s1.x[1]=s1.x[0]+s1.a;

s1.y[1]=s1.y[0];

s1.y[2]=s1.y[0]+s1.a;

s1.x[2]=s1.x[0];

s1.y[3]=s1.y[0]+s1.a;

s1.x[3]=s1.x[1]+s1.a;

cout<<"Kvadratiin talbai: "<<s1.area();

cout<<"\nKvadratiin primetr: "<<s1.pr();

/\*Toirgiin toviin tseg bolon radius urtiig garaas oruulan

primetr ba talbaig bodoj olood hevlej bna\*/

cout<<"\nToirgiin tiviin tseguudiig oruul(x,y): \n";

cin>>c1.x>>c1.y;

cout<<"Radius oruul: ";

cin>>c1.a;

cout<<"Toirgiin talbai: "<<c1.area();

cout<<"\nToirgiin primetr: "<<c1.pr();

/\*Gurvaljnii oroin tseg bolon taliin urtiig garaas oruulan busad

oroinuud bolon primetr talbaig bodoj olood hevlej bna\*/

cout<<"\nGurvaljnii oroin tsegiig oruul(x,y): \n";

cin>>t1.x[0]>>t1.y[0];

cout<<"Gurvaljnii taliin urtiig oruul: ";

cin>>t1.a;

//Gurvaljnii busad oroig end olj bna

t1.y[1]=t1.a\*sqrt(3)/2;

t1.x[1]=t1.x[0]-a/2;

t1.y[2]=t1.a\*sqrt(3)/2;

t1.x[2]=t1.x[0]+a/2;

cout<<"Gurvaljnii talbai: "<<t1.area();

cout<<"\nGurvaljnii primetr: "<<t1.pr();

}