С++ хэлний удамшил

(Лаборатори №6)

Багийн нэр: Jelyfish

Багийн ахлагч: Ц.Тэнгис

Багийн гишүүд:

* МТЭС, МКУТ, [22B1NUM6249@stud.num.edu.mn](mailto:22B1NUM6249@stud.num.edu.mn) Ц.Тэнгис
* МТЭС, МКУТ, [22B1NUM0338@stud.num.edu.mn](mailto:22B1NUM0338@stud.num.edu.mn), Д.Хүрэлсүх
* МТЭС, МКУТ, [24B1NUM0308@stud.num.edu.mn](mailto:24B1NUM0308@stud.num.edu.mn), Э.Буяннэмэх

Багийн зорилго:

Энэхүү лабораторийн ажлаар бид С++ хэлний удамшлын зарчмыг ойлгож, анхан шатны удамшлын бүтэц ашиглан даалгаварт өгөгдсөн дасгал бодлогыг практикт хэрэгжүүлж хэрхэн ажиллаж байгааг мэдэж авах. Мөн багийн хамтын ажиллагааг харуулах зорилготой.

# Оршил

Энэхүү лабораторийн тайланд С++ хэлний удамшлын зарчим (inheritance) болон түүний хэрэгжилтийг судалж, Shape → 2DShape → Circle, Square, Triangle гэсэн шаталсан классуудын бүтцийг хэрэгжүүлсэн үр дүнг танилцуулна. Тайланг уншсанаар уншигч удамшлын үндсэн ойлголт болох эцэг болон хүүхэд классын харилцаа, кодын дахин ашиглалт, өргөтгөх боломжийг ойлгох бөгөөд С++ хэлэнд public, protected, private горимууд хэрхэн ялгаатай ажилладгийг мэдэж авах болно.

# Зорилго

Энэхүү лабораторийн тайлангийн зорилго нь С++ хэл дээр удамшлын зарчмыг онолын хүрээнд ойлгох бус мөн түүнийг ашиглан классын шаталсан бүтцийг зохион байгуулах, кодын давхардлыг багасгах, дахин ашиглалтыг сайжруулахад оршино. Түүнд хүрэхийн тулд дараах зорилтуудыг дэвшүүлэв:

* Удамшлын онолын үндсийг судалж, C++ хэл дээр хэрхэн хэрэгждэгийг тодорхойлох.
* public, protected, private удамшлын горимын ялгааг жишээгээр харуулах.
* Shape ба 2DShape классуудын нийтлэг шинж чанарыг ерөнхийлж, тэднээс Circle, Square, Triangle классуудыг удамшуулах.
* Хүүхэд классуудын тусгай гишүүн өгөгдөл ба функцуудыг тодорхойлж, талбай болон периметр/тойргийн урт тооцох аргуудыг хэрэгжүүлэх.

# Онолын судалгаа

## Удамшил гэж юу вэ? (тодорхойлолт, C++ хэл дээр хэрхэн хэрэгжүүлдэг талаар бичнэ)

Объект хандалтат програмчлалд удамшил нь нэгэнт бий болсон зүйлийг дахин хэрэглэхтэй холбоотой ойлголт [1] бөгөөд өөрөөр хэлвэл бэлэн байгаа классаас шинэ класс үүсгэх арга юм. Бэлэн байгаа класс нь өмнө олон туршигдаж ашиглагдсан байдаг тул ийм классыг өөрийн програмд удамшуулан хэрэглэх нь хийх ажлын хэмжээ, алдаа гаргалтыг багасгадаг бөгөөд хөгжүүлэлтийн хугацааг богиносгож өртөг бууруулдаг. шинэ класс (охин класс) нь хуучин класс (эх класс)-ын шинж чанар (attribute) болон үйлдэл (method)-ийг өвлөн авах боломжийг олгодог механизм юм. Өөрөөс нь шинэ класс үүсч байгаа классыг үндсэн класс, дээд класс, эх класс харин түүнээс үүсэж буй классыг дэд класс, охин класс гэнэ [1].

C++ хэл дээр хэрэгжүүлэх хэлбэр:

class <шинэКлассНэр> : <public, private, protected> <эхКлассынНэр>{

// ШинэКлассын гишүүн өгөгдөл

// ШинэКлассын гишүүн функцүүд

};

Жишээ:

class Hun{

public:

string ner;

void taniltsuulah() {

cout<<"sain baina uu namaig "<<ner<<" gedeg."<<endl;

}

};

class Oyutan : public Hun{

public:

string surguuli;

void suraltsah() {

cout<<ner<<" ni "<<surguuli<<" surguuld surdag."<<endl;

}

};

int main(){

Oyutan o;

o.ner="bat";

o.surguuli="NUM";

o.taniltsuulah();

o.suraltsah();

}



Зураг 1 Жишээ гаралт 1

## Удамшлын горим. public, private, protected горимын талаар тайлбарлаж жишээгээр батална.

|  |  |  |  |
| --- | --- | --- | --- |
| Удамшлын горим | Эх классын public гишүүн | Эх классын protected гишүүн | Эх классын private гишүүн |
| **public** | public | protected | удамшдаггүй |
| **protected** | protected | protected | удамшдаггүй |
| **private** | private | private | удамшдаггүй |

*Private удамшлын горим*

Эх класс private горимоoр удамшихад түүний public болно protected гишүүн өгөгдөл охин классд private хандалтын түвшинтэй гишүүн өгөгдөл болж удамшдаг эдгээр функцүүдрүү удамших классын гишүүн функцээр дамжиж хандана. Тус горимоор удашихад эх классын private өгөгдөл охин классд удамшихгүй.

С++ удамших Классын тодорхойлолт загвар

class <шинэКлассНэр> : private <эхКлассынНэр>{

ШинэКлассын шиүүн өгөгдөл

ШинэКлассын гишүүн функцүүд

};

Жишээ С++ хэл дээр Eej класс private горимоор ohin классд удамшиж байна. Дүрмийн дагуу protected, public өгөгдлүүд удамшиж буй ohin классд private өгөгдөл болж орсон байна.

1. class Eej{
2. public:
3. int aa;
4. protected:
5. **int bb;**
6. private:
7. int cc;
8. };
9. class ohin : private Eej{
10. **private:**
11. int aa;
12. int bb;
13. }

*Public удамшлын горим*

Эх класс public горимоор удамших үед түүний public гишүүн өгөгдөл охин классын public гишүүн болно мөн эх классын protected гишүүн өгөгдөл охин классын protected гишүүн болно.

С++ удамших Классын тодорхойлолт загвар

class <шинэКлассНэр> : public <эхКлассынНэр>{

ШинэКлассын шиүүн өгөгдөл

ШинэКлассын гишүүн функцүүд

};

Public горимоор удамшиж буй жишээ с++ хэл дээр

1. class Eej{
2. public:
3. int aa;
4. protected:
5. **int bb;**
6. private:
7. int cc;
8. };
9. class ohin : public Eej{
10. **public:**
11. int aa;
12. Protected:
13. int bb;

}

*Protected удамшлын горим*

Эх класс protected горимоор удамших үед түүний public гишүүн өгөгдөл охин классын protected гишүүн болно мөн эх классын protected гишүүн өгөгдөл охин классын protected гишүүн болно.

С++ удамших Классын тодорхойлолт загвар

class <шинэКлассНэр> : protected <эхКлассынНэр>{

ШинэКлассын шиүүн өгөгдөл

ШинэКлассын гишүүн функцүүд

};

Public горимоор удамшиж буй жишээ с++ хэл дээр

1. class Eej{
2. public:
3. int aa;
4. protected:
5. **int bb;**
6. private:
7. int cc;
8. };
9. class ohin : protected Eej{
10. **protected:**
11. int aa;
12. int bb;

}

## Удамшлын давуу талуудыг тоочин бичиж бодит жишээн дээр тайлбарла.

Удамшлын давуу тал

* Алдаа гаралт бага
* Кодыг дахин давтаж бичихгүй
* Хөгжүүлэлт хурд бага
* Өргөтгөхөд хялбар

Алдаа гаргалт бага

* Удамших класс нь өмнө нь бичигдсэн олон удаа шалгагдсан байх боломжтой тул алдаа гаргалт бага байна.

Кодыг дахин давтаж бичихгүй

* Эх классд тодорхойлж өгсөн public, protected гишүүнийг охин классд дахин бичиж өгөх шаардлага байдаггүй.

Өргөтгөхөд хялбар

* Шинэ функц нэмэхэд бүх удамшсан классд үйлчилнэ

1. class Eej{
2. public:
3. int aa;
4. protected:
5. **int bb;**
6. private:
7. int cc;
8. };
9. class ohin : protected Eej{
10. **protected:**
11. int aa;

int bb;

## Удамшлын хэдэн төрөл байдаг вэ? Тус бүрийг тайлбарлан бич.

Удамшлын 5 төрөл байдаг.

* 1. Нэг-нэг буюу дан, энгийн удамшил
  2. Олон-Нэг буюу нийлмэл удамшил
  3. Олон түвшинт удамшил
  4. Шаталсан удамшил
  5. Холимог удамшил

**Нэг-нэг буюу энгийн удамшил.**

Удамших класс зөвхөн нэг эх класстай байна. Өөрөөр хэлвэл нэг эх нэг охинтой байх тохиолдол юм.

Жишээ нь:

Class Khun{};

Class oyutan: public Khun{}

**Олон-нэг буюу нийлмэл будамшил**

Удамших класс нь нэгээс олон эх класстай байж болно. Жишээ нь оюутан нь хүн классаас мөн иргэн классаас удамшсан байж болно.

Жишээ нь:

Class Khun{};

Class Irgen{};

Class oyutan: public Khun, public Irgen{}

**Олон түвшинт удамшил**

Удамших класс өөрөө удамших классаас үүснэ. Жишээ нь : А классаас Б удамшсан бөгөөд Б классаас В класс удамшсан байхыг хэлнэ. Өөрөөр хэлвэл Өвөө классаас Аав класс удамшсан бол аав классаас хүүхэд класс удамшсан байж болно.

Class Khun{};

Class Irgen: public Khun{};

Class oyutan: public Irgen{}

**Шаталсан удамшил**

Нэг эх классаас хоёроос цөөнгүй класс үүсэж байвал ийм удамшлыг шаталсан удамшил гэнэ. Жишээ нь Амьтан классаас сээр нуруутан, жигүүртэн, махан идэштэн зэрэг олон класс удамшиж болно. Ийм төрлийн удамшлыг шаталсан удамшил гэнэ.

Class Khun{};

Class Irgen: public Khun{};

Class oyutan: public Khun{};

**Холимог удамшил**

Холимог удамшил бол олон-нэг буюу нийлмэл удамшил, шаталсан хоёрын нийлэмж удамшил юм.

Class Khun{};

Class Irgen: public Khun{};

Class Jolooch: public Khun{};

Class oyutan: public Irgen, public Jolooch{};

# 4. Хэрэгжүүлэлт

Shape классаас 2DShape гэдэг класс удамшина. 2DShape классаас Circle, Square, Triangle классууд удамшина. Доор хүүхэд классуудын шинж болон функцууд өгөгдсөн бол Shape, 2DShape классуудын шинжийг ерөнхийллийн аргаар зохиомжилж удамшуул.

class shape{            //shape үндсэн класс тодорхойлолт

    public:             //public хандалтын түвшинтэй

        float a;        //талын урт, радиус хадгалах хувьсагч

        char \*name;     //дүрсийн нэрийг хадгалах хаяган хувьсагч

};

## 4.1 Тойрог:

  гишүүн өгөгдөл-тойргийн төвийн координат, радиус, нэр;

    гишүүн функц-талбай олох, тойргийн урт тооцох

class circle:public twod{        //2dshape классаас удамшина

   float area();                 //талбай олох функц

   float perimeter();            //периметр олох функц

   void coordinate();            //координатын цэгүүдийг олох функц

};

float circle::perimeter(){      //периметр функц тодорхойлолт

    return 2\*3.14\*a;            //периметрийг олон буцааж байна

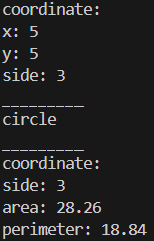
}

float circle::area(){           //талбай олох функцийн тодорхойлолт

        return 3.14\*a\*a;        //талбайг олон буцааж байна

    }

void circle::coordinate(){



Зураг 2 Жишээ гаралт 2

## 4.2 Квадрат:

    гишүүн өгөгдөл-оройнуудын координат, талын урт, нэр;

    гишүүн функц-талбай олох, приметр тооцох

class square:public twod{        //2dshape классаас удамшина

   float area();                 //талбай олох функц

   float perimeter();            //периметр олох функц

   void coordinate();            //координатын цэгүүдийг олох функц

};

float square::perimeter(){      //периметр функц тодорхойлолт

        return 4\*a;                 //периметрийг олон буцааж байна

    }

    float square::area(){           //талбай олох функцийн тодорхойлолт

        return a\*a;                 //талбайг олон буцааж байна

    }

    void square::coordinate(){      //координатын цэгүүдийг олох функц тодорхойлолт

        xy[1][0]=xy[0][0]+a;        //х у цэгүүдийг олж байна

        xy[1][1]=xy[0][1];

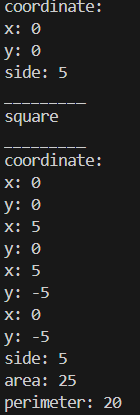
        xy[2][0]=xy[0][0]+a;

        xy[2][1]=xy[0][1]-a;

        xy[3][0]=xy[0][0];

        xy[3][1]=xy[0][1]-a;

    }



Зураг 3 Жишээ гаралт 3

## 4.3 Зөв гурвалжин:

    гишүүн өгөгдөл-оройнуудын координат, талуудын урт, нэр;

    гишүүн функц-талбай, приметр тооцох

class triangle:public twod{        //2dshape классаас удамшина

   float area();                 //талбай олох функц

   float perimeter();            //периметр олох функц

   void coordinate();            //координатын цэгүүдийг олох функц

};

float triangle::perimeter(){        //периметр функц тодорхойлолт

    return 3\*a;                     //периметрийг олон буцааж байна

}

float triangle::area(){             //талбай олох функцийн тодорхойлолт

    float h=sqrt(a\*a-a/2\*a/2);      //гурвалжны өндөрийг олж байна

    return a\*h/2;                   //талбайг олон буцааж байна

}

void triangle::coordinate(){        //координатын цэгийг олох функц тодорхойлолт

    float h=sqrt(a\*a-a/2\*a/2);

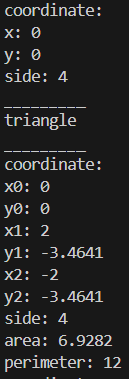
    xy[1][0]=xy[0][0]+a/2;          //х у цэгүүдийг олж байна

    xy[1][1]=xy[0][1]-h;

    xy[2][0]=xy[0][0]-a/2;

    xy[2][1]=xy[0][1]-h;

}



Зураг 4 Жишээ гаралт 4

## Тайлбар:

Тойргийн төвийн цэг, радиусд утга онооно. Квадратын талын урт, зүүн дээд оройн координатад утга оноогоод бусад оройг тооцоолж классын харгалзах гишүүн өгөгдөлд хадгална. Зөв гурвалжингийн талын урт, дээд оройн координатыг утга оноогоод бусад оройг тооцоолно

# 5. Дүгнэлт

Багийн гишүүд тэгш оролцоотой байсан. Хийх ажилаа тэнцүү хуваасан. Хэрхэн хуваасан бэ гэвэл 3 нь ширхэг хүүхэд классаа нэг нэгээр нь аваад доорх байдлаар хуваалаа.

1. Тойрог + онолын хэсэг
2. Квадрат + оршил, дүгнэлт, зорилго
3. Зөв гурвалжин + үндсэн код

Үндсэн хэрэглэгдхүүн Github ашигласан. Тулгарсан асуудал docx файл буюу тайлангийн материалаа merge хийх байсан. Дараа нь 3-уулаа үндсэн нэг ойлголттой болох.

Энэ багийн ажлын хүрээнд юу сурсан бэ гэвэл дээр дурьдсан удамшлын зарчмыг судалж практикт хэрэгжүүлсэн. Мөн багийн гишүүдийн хийх ажилаа эрт хувааж хангалттай төлөвлөх, хийх цаг гаргах.

# 6.Ашигласан материал

|  |  |
| --- | --- |
| [1] | B. Stroustrup, *Programming: Principles and Practice Using C++ 2th edition*, Addison Wesley, 2014. |
| [2] | П. Ж, Обьект хандлагат технологийн С++ програмчилал, Улаанбаатар: USI Books, 2008. |
| [3] | S. Bjarne , The C++ Programming Language, Addison-Wesley, 2013. |

# 7. Хавсралт

Shapes.h

#ifndef \_\_shape\_\_

#define \_\_shape\_\_

class shape{            //shape үндсэн класс тодорхойлолт

    public:             //public хандалтын түвшинтэй

        float a;        //талын урт, радиус хадгалах хувьсагч

        char \*name;     //дүрсийн нэрийг хадгалах хаяган хувьсагч

};

#endif

2d.h

#include "shape.h"                          //shape.h header file холбож өгнө

#ifndef \_\_2d\_\_

#define \_\_2d\_\_

class twod: public shape{                   //shape ээс удамшсан 2Dshape класс тодорхойлолт

    public:                                 //public хандалтын түвшинтэй

        virtual float area()=0;             //ингэснээр удамшсан функцууд даин тодорхойлолт хийж болно

        virtual float perimeter()=0;

        float xy[4][2];                     //координатын цэгүүдийг хадгалах хувьсагч

        void get\_data(void);                //гараас утга оноох функц

        void show\_data(void);               //мэдээлэл дэлгэцэлж харуулах функц

        virtual void coordinate()=0;        //координатын цэгүүдийг олох функц

};

#endif

2d.cpp

#include "2d.h"

#include <iostream>

#include<string.h>

using namespace std;

void twod::get\_data(void){                                          //get data функц тодорхойлолт

            cout<<"coordinate: "<<endl;

            cout<<"x: ";

            cin>>xy[0][0];                                          //xy[0][0] буюу х цэгийг гараас оруулна

            cout<<"y: ";

            cin>>xy[0][1];                                          //xy[0][1] буюу у цэгийг гараас оруулна

            cout<<"side: ";

            cin>>a;                                                 //талын уртыг гараас оруулна

};

void twod::show\_data(void){                                         //мэдээлэл дэлгэцэлж харуулах функц тодорхойлолт

            cout<<"\_\_\_\_\_\_\_\_\_"<<endl<<name<<endl<<"\_\_\_\_\_\_\_\_\_"<<endl;

            cout<<"coordinate: "<<endl;

            if(strcmp(name,"triangle")==0){                         //нэр нь гурвалжин байвал 3 цэгийн координатыг л хэвлэж харуулна

                for(int i=0;i<3;i++){

                cout<<"x"<<i<<": ";

                cout<<xy[i][0]<<endl;

                cout<<"y"<<i<<": ";

                cout<<xy[i][1]<<endl;

                }

            }else if(strcmp(name,"square")==0){                     //нэр нь дөрвөлжин байвал 4 цэгийн мэдээллийг хэвлэж харуулна

                for(int i=0;i<4;i++){

                cout<<"x: ";

                cout<<xy[i][0]<<endl;

                cout<<"y: ";

                cout<<xy[i][1]<<endl;

                }

            }

            cout<<"side: ";

            cout<<a<<endl;

            cout<<"area: ";

            cout<< area()<<endl;

            cout<<"perimeter: ";

            cout<< perimeter()<<endl;

        }

Circle.h

#include "2d.h"                  //2d.h header file холбож байна

#ifndef \_\_circle\_\_

#define \_\_circle\_\_

class circle:public twod{        //2dshape классаас удамшина

   float area();                 //талбай олох функц

   float perimeter();            //периметр олох функц

   void coordinate();            //координатын цэгүүдийг олох функц

};

#endif

Circle.cpp

#include "circle.h"             //circle.h header file холболт

#include <math.h>               //math сан холболт

float circle::perimeter(){      //периметр функц тодорхойлолт

    return 2\*3.14\*a;            //периметрийг олон буцааж байна

}

float circle::area(){           //талбай олох функцийн тодорхойлолт

        return 3.14\*a\*a;        //талбайг олон буцааж байна

    }

void circle::coordinate(){

}

Square.h

#include "2d.h"                  //2d.h header file холбож байна

#ifndef \_\_square\_\_

#define \_\_square\_\_

class square:public twod{        //2dshape классаас удамшина

   float area();                 //талбай олох функц

   float perimeter();            //периметр олох функц

   void coordinate();            //координатын цэгүүдийг олох функц

};

#endif

Square.cpp

#include "square.h"                 //square.h header file холбож байна

    float square::perimeter(){      //периметр функц тодорхойлолт

        return 4\*a;                 //периметрийг олон буцааж байна

    }

    float square::area(){           //талбай олох функцийн тодорхойлолт

        return a\*a;                 //талбайг олон буцааж байна

    }

    void square::coordinate(){      //координатын цэгүүдийг олох функц тодорхойлолт

        xy[1][0]=xy[0][0]+a;        //х у цэгүүдийг олж байна

        xy[1][1]=xy[0][1];

        xy[2][0]=xy[0][0]+a;

        xy[2][1]=xy[0][1]-a;

        xy[3][0]=xy[0][0];

        xy[3][1]=xy[0][1]-a;

    }

Triangle.h

#include "2d.h"                  //2d.h header file холбож байна

#ifndef \_\_triangle\_\_

#define \_\_triangle\_\_

class triangle:public twod{        //2dshape классаас удамшина

   float area();                 //талбай олох функц

   float perimeter();            //периметр олох функц

   void coordinate();            //координатын цэгүүдийг олох функц

};

#endif

Triangle.cpp

#include "triangle.h"               //triangle.h header file холбож байна

#include <math.h>

float triangle::perimeter(){        //периметр функц тодорхойлолт

    return 3\*a;                     //периметрийг олон буцааж байна

}

float triangle::area(){             //талбай олох функцийн тодорхойлолт

    float h=sqrt(a\*a-a/2\*a/2);      //гурвалжны өндөрийг олж байна

    return a\*h/2;                   //талбайг олон буцааж байна

}

void triangle::coordinate(){        //координатын цэгийг олох функц тодорхойлолт

    float h=sqrt(a\*a-a/2\*a/2);

    xy[1][0]=xy[0][0]+a/2;          //х у цэгүүдийг олж байна

    xy[1][1]=xy[0][1]-h;

    xy[2][0]=xy[0][0]-a/2;

    xy[2][1]=xy[0][1]-h;

}

Main.cpp

#include "circle.h"                         //circle.h file холбож байна

#include "triangle.h"                       //triangle.h file холбож байна

#include "square.h"                         //square.h file холбож байна

#include<iostream>                          //iostream сан холбож байна

#include<string.h>

using namespace std;

main(){

    twod \*shapes[3];                        //2dshape хэлбэрийн 3 хаяган хувьсагч авах массив зарлалт

    shapes[0]=new circle;                   //эхний объектед circle хэлбэрийн санах ой нөөцлөж байна.

    shapes[0]->name=new char[7];            //нэрэнд нь 7 хэмжээтэй санах ой нөөцөлж байна.

    strcpy(shapes[0]->name,"circle");       //нөөцөлсөн санах ой доо circle утгыг хуулж байна

    shapes[1]=new triangle;                 //эхний объектед circle хэлбэрийн санах ой нөөцлөж байна.

    shapes[1]->name=new char[9];            //нэрэнд нь 9 хэмжээтэй санах ой нөөцөлж байна

    strcpy(shapes[1]->name,"triangle");     //нөөцөлсөн санах ой доо triangle утгыг хуулж байна

    shapes[2]=new square;                   //эхний объектед circle хэлбэрийн санах ой нөөцлөж байна.

    shapes[2]->name=new char[7];            //нэрэнд нь 7 хэмжээтэй санах ой нөөцөлж байна

    strcpy(shapes[2]->name,"square");       //нөөцөлсөн санах ой доо square утгыг хуулж байна

    for(int i=0;i<3;i++){                   //давталт гүйлгэж бүх объектийн гишүүн өгөгдлүүдэд утга оруулна

        shapes[i]->get\_data();

        shapes[i]->coordinate();            //мэдээллийг харуулахаас өмнө coordinate функцийг дуудаж ажиллуулж байна

        shapes[i]->show\_data();             //мэдээллийг хэвлэж харуулна

    }

}