# Классийг Загварчлан Удамшуулах, Эх Класс Болон Удамшсан Классуудын Харилцаа (Лаборатори №6)

Ү.Мэндсайхан

МУИС, ХШУИС, Компьютерын ухаан, mendsaikhan@gurvanshiidel.mn

# 1. ОРШИЛ

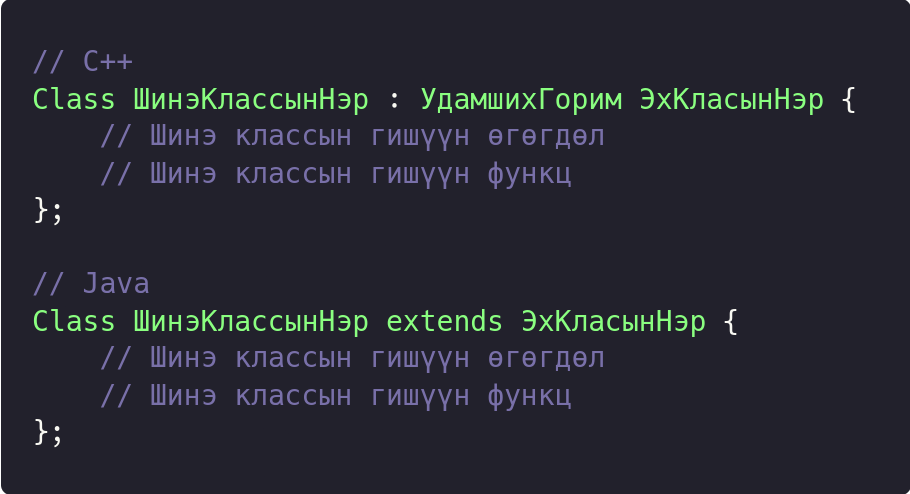
Удамшил гэж юу болох, хэрхэн хэрэгжүүлэх, давуу болон сул талын тухай судлан TwoDimensionalShape класс үүсгэн уг классаас Circle, Square, Triangle зэрэг хүүхэд классуудыг үүсгэсэн. Удамшлын горимуудыг ашиглан эх классын гишүүд рүү хандах, хүүхэд классын харилцааг харуулсан бөгөөд нэмэлтээр шинээр үүссэн классын байгуулагч функц хэрхэн ажиллаж буйг оруулав.

# 2. ЗОРИЛГО

Лабораторийн ажлын хүрээнд shape классыг удамшуулан TwoDimensionalShape классыг тодорхойлох бөгөөд энэхүү классаа дахин удамшуулан Circle, Square, Triangle гэх классуудыг тодорхойлох. Үүний тулд дараах дараах ойлголтуудыг судлах.  
 2.1 Удамшил гэж юу вэ?  
 2.2 Удамшлын горим  
 2.3 Удамшлын давуу талуудыг тоочин бичих  
 2.4 Удамшлын хэдэн төрөл байдаг вэ? Тус бүрийг тайлбарлан бичих

# 3. ОНОЛЫН СУДАЛГАА

## 3.1 Удамшил гэж юу вэ?

Удамшил гэдэг нь өмнө нь тодорхойлсон классыг ашиглан түүнийгээ өргөтгөн шинэ класс үүсгэх арга юм. Бид ямар нэгэн зүйлийн ерөнхий шинжээр класс үүсгэн тухайн классаа илүү олон өгөгдөл бүхий тодорхой нэг зүйлд зориулан хэрэглэхдээ тухайн классаа удамшуулж шинэ класс үүсгэдэг. Өөрөөс нь шинэ класс үүсэж байгаа классыг “Эх класс” бөгөөд шинээр үүсэж байгаа классыг “Хүүхэд класс” гэж нэрлэдэг. C++ болон Java хэл дээр удамших классыг тодорхойлохдоо:   


## 3.2 Удамшлын горимууд

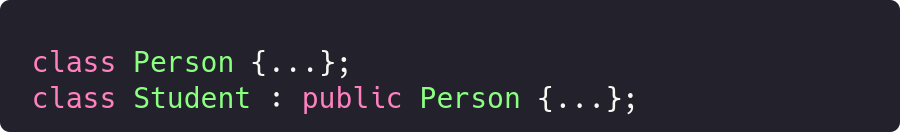
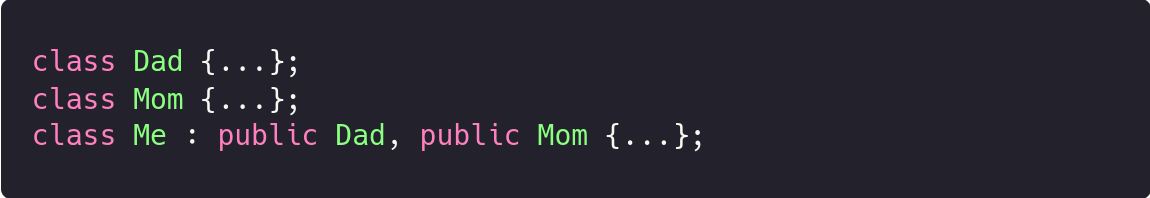
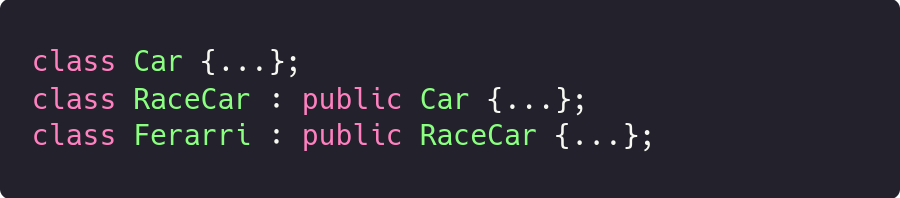
Объектыг удамшуулахын тулд өмнөх эх классын гишүүн өгөгдлүүдийг/функцуудын “Хүүхэд классаас” хандах горимыг тодорхойлж өгдөг бөгөөд үүнийг удамшлын горим гэдэг. Гурван удамшлын горим байдаг ба тэдгээр нь public, private болон protected юм.

* Public  
  Классыг public горимоор удамшуулж өгснөөр “Хүүхэд класс”-аас эх класст буй гишүүдэд ямар нэгэн хандалтын горимын нэмэлт өөрчлөлтгүй шууд байгаа горимоор нь хандаж болдог.   
   — Public гишүүдэд шууд хандаж болно.  
   — Private гишүүдэд шууд хандаж мэдээж болохгүй.  
   — Protected гишүүдэд зөвхөн “Хүүхэд класс”-аас хандах боломжтой.
* Private  
  Private горимоор удамшуулсан “Хүүхэд класс” нь “Эх класс”-руу шууд хандах боломжгүй байдаг бөгөөд бүх гишүүд нь private хэлбэртэй болж удамшдаг.
* Protected  
  Protected горим нь гаднаас шууд хандах боломжгүй ч “Хүүхэд класс”-аас хандах боломжтой.   
   — Public & Protected гишүүдэд шууд хандах боломжгүй  
   — Private гишүүн рүү ямар ч горимоор удамшуулсан шууд хандах боломжгүй

## 3.2 Удамшлын давуу талуудыг тоочин бичих

Удамшил буюу эх классыг өөрчлөхгүйгээр түүнээс удамших шинэ класс үүсгэж хэрэглэх нь дараах сайн талуудтай. Үүнд:  
 1. Удамшлыг ашигласнаар ерөнхий ойлголт бүхий классыг тодорхойлон шинэ класс үүсгэн өмнөх классын мэдээллийг өөрчлөлгүйгээр дахин ашиглах боломжийг бий болгодог.   
 2. Тухайн эх классаас үүдэлтэй олон хүүхэд класс үүсгэх бол удамшлыг ашиглан илүү бага кодын тусламжтайгаар бичих мөн ямар нэгэн эх классаас хамаарсан алдаа харьцангуй багасна.  
 3. Илүү бага зардлаар буюу энэхүү классаа ашиглан илүү бүтээмжийг ихэсгэх боломжтой.

## 3.4 Удамшлын хэдэн төрөл байдаг вэ? Тус бүрийг тайлбарлан бичих

Удамшлын 5 төрөл байдаг.  
1. Нэг – нэг удамшил (Энгийн)  
Ямар нэгэн хүүхэд класс зөвхөн 1 эх классаас удамших хэлбэр  
  
2. Нэг – Олон удамшил (Нийлмэл)  
Хүүхэд класс нь олон эх классаас удамшиж буй хэлбэр  
  
3. Олон түвшинт удамшил  
Ямар нэгэн хүүхэд класс удамшин шинээр хүүхэд класс үүсгэх буюу давхар удамших хэлбэр  
  
4. Шаталсан удамшил  
Нэг эх классаас олон хүүхэд класс үүсэх хэлбэр  
  
5. Холимог удамшил  
Холимог удамшил бол олон-нэг буюу нийлмэл удамшил, шаталсан удамшил хоёрын нийлэмж удамшил юм.  


# 4. ХЭРЭГЖҮҮЛЭЛТ Shape классаас TwoDShape, цаашлан шаталсан хэлбэрээр Circle, Square, Triangle зэрэг хүүхэд классуудыг үүсгэв. Shape эх класст эдгээр хүүхэд класс бүрд байх name өгөгдлийг тодорхойлсон бөгөөд үүнийг динамик санах ойн new operator ашиглан тохирсон хэмжээ бүхий санах ой нөөцлөх байдлаар загварчлав. TwoDShape класст оройн цэгийн координатууд болон радиусыг гишүүн өгөгдөл болгон авч түүнээс удамшуулан дүрс бүрд харгалзах бусад оройн координатуудыг олов.

# Тойрог: Гишүүн өгөгдөл - тойргийн төвийн координат, радиус, нэр Гишүүн функц - талбай олох, тойргийн урт тооцох Квадрат: Гишүүн өгөгдөл - оройнуудын координат, талын урт, нэр Гишүүн функц - талбай олох, приметр тооцох Зөв гурвалжин: Гишүүн өгөгдөл - оройнуудын координат, талуудын урт, нэр Гишүүн функц - талбай олох, приметр тооцох Үр дүн:

5. ДҮГНЭЛТ

Удамшлыг ашигласнаар программын найдвартай үйл ажиллагаа, хөгжүүлэлтийн хурдыг нэмэгдүүлэх, өртөг бууруулах зэрэг олон давуу талууд үүсдэг. Хүүхэд буюу үүсэж буй класст байгуулагч тодорхойлон main функц дотроо объект бий болгон ажиллуулж үзэхэд тухайн хүүхэд классын эх класст буй байгуулагч түрүүлж ажиллах дууссаны дараа хүүхэд классын байгуулагч ажилладаг юм байна.

# 6. АШИГЛАСАН МАТЕРИАЛ

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

# 7. ХАВСРАЛТ

Github линк, [Lab 6](https://github.com/mendeeislife/object/blob/main/lab5/lab6.cpp)