(Лаборатори №8)

Э.Багабанди

ХШУИС - ийн ПХ хөтөлбөрийн

3-р түвшний оюутан, [19B1NUM0700@num.edu.mn](mailto:19B1NUM0700@num.edu.mn)

1. **ОРШИЛ/УДИРТГАЛ**

Уг лабораторийн хүрээнд өгөгдсөн дасгалуудыг гүйцэтгэж, жижиг хэмжээний програм хөгжүүлэхэд ашиглагдсан C++ хэлний онолын ойлголт /удамшилын харьцаа, бүрдэл харьцаа, хориглолт/ - уудыг судалж, эзэмшсэн болно.

2. **ЗОРИЛГО**

С++ хэлний удамшилын харьцаа, бүрдэл харьцаа, тэдгээрийн хэрэгжүүлэлт, бүрдэл харьцааны constraint буюу хориглолтын талаар судалж, үүнийг ашиглан лабораторт өгөгдсөн дасгалуудыг гүйцэтгэн жижиг хэмжээний хэрэгжүүлэлт хийх.

3. **ОНОЛЫН СУДАЛГАА**

3.1. Удамшил ямар харьцаа үүсгэдэг вэ?

* Удамшил гэдэг нь нэгэнт бий болсон зүйлийг дахин хэрэглэхтэй холбоотой ойлголт бөгөөд туршиж бэлэн болгосон классын кодыг өөр системд дахин хэрэглэх буюу эргэн хэрэглэгдэх чадвар, чадамжтай болгодог. Өөрөөр хэлбэл бэлэн байгаа классаас шинэ класс үүсгэх арга технологи гэж хэлж болох бөгөөд энэ үед шинээр үүсч буй класс нь өмнөх классынхаа шинжийг өвлөн авна.
* Өөрөөс нь класс үүсч байгаа классыг үндсэн(дээд/эх) класс, шинээр үүсч байгааг нь удамших, удамшсан(дэд/охин/хүүхэд) класс гэнэ.
* Удамшиж үүсэх класс нь эх классынхаа шинж, аргаас өвлөж авахаас гадна өөрийн гэсэн шинжтэй, өөрөөр хэлбэл эх класст байхгүй гишүүдтэй байж болно. Эх классын гишүүн функцийг удамших класс дотор дахин тодорхойлж болох ба ингэхэд эх классын эх код заавал байх албагүй юм.
* Удамшил нь класс хооронд “**is a**” буюу “**- төрлийнх, бол**” гэсэн харьцаа үүсгэдэг. Өөрөөр хэлбэл, суудлын машин нь машин төрлийнх гэвэл суудлын машин нь машинаас удамшсан гэж үзэх юм.

3.2. Объект хандлагат программчлалд хэрхэн хэрэгжүүлдэг вэ?

* **Классыг public горимоор удамшуулж үүсгэх** үеийн удамших классын тодорхойлолт:



* Эх класс public горимоор удамших тохиолдолд түүний public гишүүд удамшиж үүсэх классын public гишүүд болно. Иймээс удамших классын функц ба объект эх классынхаа public гишүүд рүү хандана. Харин эх класс public горимоор удамших үед түүний private гишүүд удамших хэдий ч удамших классын гишүүд эх классынхаа private гишүүд рүү хандаж чадахгүй. Мөн эх класс public горимоор удамших үед түүний protected гишүүд удамших классынхаа protected гишүүд болох тул уг гишүүд рүү удамших классын функцээс хандаж болно.
* **Классыг private горимоор удамшуулж үүсгэх** үеийн удамших классын тодорхойлолт:



* Эх класс private горимоор удамших тохиолдолд түүний public болон protected гишүүд удамшиж үүсэх классын private гишүүд болно. Иймд удамших классын объект эх классынхаа гишүүд рүү хандаж чадахгүй. Харин функц нь классынхаа private гишүүд рүү хандаж болдог тул эх классынхаа public, protected гишүүд рүү удамших классын функцээс хандана.
* **Классыг protected горимоор удамшуулж үүсгэх** үеийн удамших классын тодорхойлолт:



* Эх класс protected горимоор удамших тохиолдолд түүний public болон protected гишүүд удамшиж үүсэх классын protected гишүүд болно. Иймд уг гишүүд рүү удамших классын функцээс хандаж болно. Харин эх класс protected горимоор удамших үед түүний private гишүүд удамших хэдий ч удамших классын гишүүд эх классынхаа private гишүүд рүү хандаж чадахгүй.
* **Удамших горимын хүснэгт:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Эх классын гишүүд** | **Удамших горим** | | |
| Public | private | protected |
| public | Public | private | protected |
| private | X | X | X |
| protected | Protected | private | protected |

3.3. Бүрдэл харьцаа гэж юу вэ?

* Бүрдэл харьцаа гэдэг нь өөр классын объект тухайн классын гишүүн өгөгдөл болж, ашиглагдахын хэлнэ.
* Бүрдэл харьцаа нь класс хооронд “has a” буюу “-тай3” харьцаа үүсгэдэг. Өөрөөр хэлбэл машин нь мотортой гэвэл машин болон мотор нь хоорондоо бүрдэл харьцаатай бөгөөд мотор нь машин гэх классын гишүүн өгөгдөл болох юм.

3.4. Объект хандлагат программчлалд хэрхэн хэрэгжүүлдэг вэ?

* Бүрдэл харьцааг объект хандлагат програмчлалд дараах байдлаар хэрэгжүүлнэ.

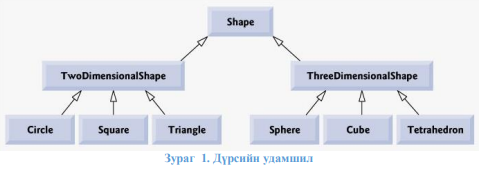


3.4. Бүрдэл харьцааны хориглолт (constraint) гэж юу вэ?

* Бүрдэл харьцааны хориглолт буюу constraint гэдэг нь өөр нэг классын объект тухайн классын гишүүн өгөгдөл болж, ашиглахад үед тухайн класст уг гишүүн өгөгдөл объект хэдэн ширхэг хамгийн багадаа, хамгийн ихдээ хэд байхыг заасан хязгаарлалты хэлнэ. Тухайлбал, машин мотор хоёр нь хоорондоо бүрдэл харьцаатай буюу машин нь мотортой байг. Энэ үед тухайн 2 – ийн бүрдэл харьцааны хориглолт 1..n гэж үзвэл машин хамгийн багадаа нэг мотортой, хамгийн ихдээ хэдэн ч мотортой байна гэдгийг хэлэх юм.

4. **ХЭРЭГЖҮҮЛЭЛТ**

Зураг 1-д дүрсэлсэн удамшлын модны дагуу хоёр хэмжээст геометрийн дүрсүүдийн классыг байгуул.

**Тойрог:**

* Гишүүн өгөгдөл-тойргийн төвийн координат, радиус, нэр;
* Гишүүн функц-талбай олох, тойргийн урт тооцох
* Тойргийн төвийн цэг, радиусд утга онооно.

**Квадрат:**

* Гишүүн өгөгдөл-оройнуудын координат, талын урт, нэр;
* Гишүүн функц-талбай олох, приметр тооцох
* Квадратын талын урт, зүүн дээд оройн координатад утга оноогоод бусад оройг тооцоолж классын харгалзах гишүүн өгөгдөлд хадгална.

**Зөв гурвалжин:**

* Гишүүн өгөгдөл-оройнуудын координат, талуудын урт, нэр;
* Гишүүн функц-талбай, приметр тооцох
* Зөв гурвалжингийн талын урт, дээд оройн координатыг утга оноогоод бусад оройг тооцоолно

Классуудад байгаа нийтлэг шинжийг эх классад оруулаад түүнээсээ удамшуулж классыг үүсгэнэ.

5. **ДҮГНЭЛТ**

Классыг удамшуулж үүсгэснээр өмнө нь бичсэн код болон програмыг дахин хэрэглэх, бичиглэл багасах, гишүүн өгөгдөл болон функцийг дундаа хэрэглэх гэх зэрэг давуу тал боломжууд бүрддэг. Мөн класс хооронд бүрдэл харьцаа үүсгэснээр өөр нэг классын объектыг өөр класс ашиглах боломж бүрддэг бөгөөд энэ нь гишүүн өгөгдлийн төвөгтэй зохион байгуулалтыг багасгаж өгдөг. Классууд бүрдэл харьцаагаар холбогдсон үед бүрдэл харьцааны хориглолтын тавих буюу тухайн класст өгөгдлөөр орсон классын объектуудын байж болох хэмжээг зааж өгөх шаардлагатай. Ингэснээр тухайн класс заавал нэг байх эсвэл олон байх гишүүн өгөгдлийг санамсаргүйгээр устгаж, хоосон болгох зэрэг гэнэтийн үйлдлүүдээс сэргийлэх боломж бүрддэг.

6. **АШИГЛАСАН МАТЕРИАЛ**

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

7. **ХАВСРАЛ**