




Ү.СS311 Програм хангамжийн бүтээлт



Объект хандалтат програмчлалын хэлд байх шалгуурууд

Объект хандалтат хэл болон орчин нь хамтдаа объект хандалтат аргыг хэрэгжүүлэх ба бүхэл амьдралын циклын туршид арга механизмаар хангаснаар амжилттай хэрэгжих болно.



Класс

- Объект хандалтат програмчлалын хэлний үндсэн зарчим.



Даатгал (Assertion)

Хийсвэр өгөгдлийн төрлүүдтэй
классаар хангана. Тэдгээр нь
урьдчилсан ба дараах гэсэн
нөхцөлүүдтэй байх ба олон
өгөгдлийн төрөл авах чадвартай
байдаг. Мөн тэдгээрийг хянах
механизмтай байдаг.



Класс бол Модул

Класс нь модул байх ёстой.





Класс бол төрөл

Өгөгдлийн төрөл бүр класс дээр
тулгуурлана.



ТӨЛӨВТ ҮНДЭСЛЭСЭН ТООЦООЛОЛ

Feature-based computation

- Объект хандалтат тооцоололд ганцхан үндсэн тооцооллын механизм байдаг бөгөөд тэр нь ямар нэгэн классаас үүссэн тодорхой объект нь тухайн объект дахь тухайн классын төлвийг дуудах юм.
- Жишээ нь : Цонх гэсэн классын display “харуулах” гэсэн төлвийг дуудах



Төлөвт үндэслэсэн тооцоолол

Тодорхойлолт:

Төлвийг дуудах нь тооцооллын үндсэн механизм байна.




Мэдээллийг нууцлах

Тодорхойлолт:

Классын зохиогч нь тодорхой төлвийг бүх хэрэглэгч дуудах боломжтой, ямар ч хэрэглэгч дуудах боломжгүй эсвэл тодорхой заасан хэрэглэгч дуудах боломжтой байх ёстой.

Алдаа барих Exception handling

Пх-ын системд хэвийн бус үйл явдлууд явагдах боломжтой. Эдгээр нь техник хангамжийн буруу ажиллах, тоо хязгаар хэтрэх, програмын алдааны үр дүнд явагддаг. Найдвартай програм хангамж бий болгохын тулд эдгээр тохиолдлуудыг урьдчилан тооцоолох шаардлагатай.



Алдаа барих

Тодорхойлолт:

Програмчлалын хэл нь эдгээр тооцоологдоогүй хэвийн бус тохиолдлуудыг илрүүлэх механизмаар хангагдсан байх ёстой.

Өгөгдлийн төрлийн ойлголт

Static typing

Пх-ийн систем ажиллаж байх үед тодорхой объектын тодорхой төлөвийг дуудах үед тухай объект дуудлагыг хэрхэн удирдах вэ ? Үүнийг зөв ажиллуулах баталгаа нь програмчлалын хэлэнд тохирох дүрэм сахиулах явдал юм. Үүнд:


- Програмчлалын оролцож байгаа объект бүр нь классаас үүснэ эсвэл тодорхой зарлагдсан өгөгдлийн төрлөөс үүснэ.
- Тодорхой програм хангамжийн объектын тодорхой төлвийг дуудахад харгалзах классын төлвийг ашиглана гэсэн үг.
- Утга олголт, аргумент дамжуулалт нь удамшилд үндэслэсэн, дамжуулж байгаа утга, утга хүлээж авах төрөл 2 ижил байх ёстой гэсэн дүрэмд захирагдана.



Өгөгдлийн төрлийн ойлголт

Тодорхойлолт:

Сайн тодорхойлогдсон өгөгдлийн төрөлтэй систем нь тодорхой тооны өгөгдлийн төрөл зарлалт болон тохирох дүрмэнд захирагддаг. Ингэснээр системийн ажиллах үеийн өгөгдлийн төрөл аюулгүй байдлын баталгаа болно.



Өгөгдлийн төрлийн ерөнхий байдал

Тодорхойлолт:

Дурын өгөгдлийн төрлийг параметр хэлбэрээр дамжуулж тухайн классыг ашиглах боломжтойгоор классыг тодорхойлох ёстой.

Жишээ нь: `LIST[INTEGER]`, `LIST[CHAR]`



Нэг классаас удамших

Тодорхойлолт:

Класс нь нөгөө классаас удамших
боломжтой байх шаардлагатай.



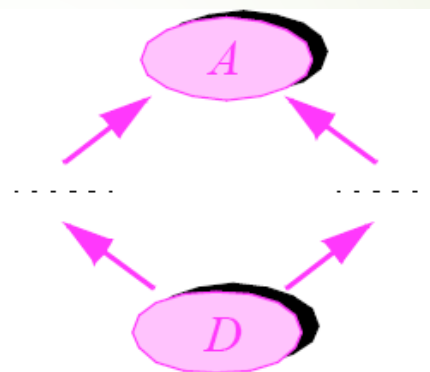
Олон классаас удамших

Тодорхойлолт:

Класс нь шаардлагатай бол олон
классаас удамших боломжтой байх
шаардлагатай.

Давтаж удамших Repeated inheritance

D классаас удамшсан 2
классаас А класс
удамшиж гарвал
дамтаж удамших
тохиолдол гарна. Энэ
тохиолдолд D-ийн
зөвхөн нэг төлөв
(sharing), D-ийн
хуулбарласан 2
төлөвтэй (replication)
байх боломжтой.



↑ *Inherits from*



Давтаж удамших

Тодорхойлолт:

Давтаж удамших үед төлвийн хуулбарыг удирдах нарийвчилсан дүрэм байх ёстой ба програм зохиогчид нэг төлөвт, олон төлөвтийн аль нь болохыг сонгох боломжтой байх ёстой.



Өгөгдлийн төрлийн ерөнхий байдлын хязгаарлалт Constrained genericity

Тодорхойлолт:

Өгөгдлийн төрлийн ерөнхий байдал болон удамшил гэсэн 2 ойлголтоос Өгөгдлийн төрлийн ерөнхий байдлыг хязгаарлах гэсэн ойлголт гарч ирнэ. Өөрөөр хэлбэл хэрэв тухайн заасан төрлөөс удамшсан төрөл байх тохиолдолд хүндрэл учирна.



Дахин тодорхойлох Redefinition

SESSION гэсэн класс байхад terminate гэсэн төлөв байлаа. SESSION классаас удамшсан REMOTE_SESSION гэсэн классд terminate гэсэн төлвийг дахин тодорхойлох шаардлага гардаг.

Тодорхойлолт:

Удамшиж авсан классын төлвийг хэрэглэхийн тулд түүнийг дахин тодорхойлох шаардлагатай.



Полиморфизм

Полиморфизм нь програмын элемент янз бүрийн боломжит төрлүүдээс үүссэн объект руу хандах боломж гэсэн ойлголт юм. Өөрөөр хэлбэл BOAT гэсэн классаас удамшсан MERCHANT_BOAT эсвэл SPORTS_BOAT классуудаас үүссэн объектууд руу хандах боломж гэсэн үг.



Полиморфизм

Тодорхойлолт:

Програм хангамжийн элемент янз бүрийн боломжит төрлүүдээс үүссэн объектууд руу удамшилд суурилсан арга ашиглан ижил төлвөөр хандах боломжтой байх ёстой.



Динамик холболт

Dynamic binding


Дахин тодорхойлох болон полиморфизм механизмуудыг хослуулан хэрэглэж ижил үйлдлээр ялгаатай үр дүн гарах үед аль классын үйлдэл буюу төлөв ажиллаж байгааг мэдэх шаардлага гардаг. Жишээ нь: `SPORT_BOAT` классын объектын төлөв ажилласан байна гэх мэт. Энэ шинж чанарыг динамик холболт гэнэ.



Динамик холболт

Тодорхойлолт:

П-ын элемент дотор төлөв дуудахад үед ажиллаж байгаа объектын харгалзах төрөлд мэдээлдэг байх ёстой.



Ажиллах үеийн шалгалт

Run-time type interrogation

Удамшлаас үүдэлтэй полиморфизм, динамик холболт гэсэн тодорхойлолт, аргууд байгаа ч гэсэн тухайн удамшлуудын үүссэн класс буюу төрөл нь бодит эсэх, түүнд баталгаатай хандаж болох эсэх нь програм хангамж алдаагүй ажиллах нэг үзүүлэлт болно.

Ажиллах үеийн шалгалт

- Програм хангамж нь объект руу ямар нэгэн асуудалгүй найдвартай аргаар хандах механизм хэрэгтэй. Энэ механизм нь полиморфизм, динамик холболт 2-той зохицож, хамтарч ажилладаг байх ёстой.

Ажиллах үеийн шалгалт

Тодорхойлолт:

Програмчлалын хэл нь програм ажиллаж байх үед объектын төрөл нь өгөгдсөн төрөлтэй ижил байна уу ? үгүй юу ? гэдгийг шалгах боломжтой байх ёстой.




Хийсвэр төлөв ба класс Deferred features and classes

Жишээ нь: BOAT гэсэн классаас
удамшсан MERCHANT_BOAT,
SPORTS_BOAT классуудаас үүссэн
объектууд. BOAT класс turn төлөвтэй.
Програм ажиллах үед хүү классаас
объект үүсэхэд эцэг класс болон
түүний төлвүүд хийсвэрээр үүссэн
байна.




Memory management and garbage collection

Объект хандалтат орчинд санах ойг цэвэрлэх үйл ажиллагаа garbage collector гэх ажиллах үеийн системийн бүрэлдэхүүн хэсгээр автоматаар хийгддэг. Энэ нь янз бүрийг заагчид, санах ойн мужууд, хэрэглэгдэхгүй болсон хувьсагчдыг чөлөөлөх, санах ойг дахин хуваарилах үйл ажиллагаа хийнэ.




Тэсвэртэй байдал Persistence

Ихэнх програмд нэг session-оос нөгөө session дуудагдах үед зарим объектыг хадгалах шаардлага гардаг. Иймээс програм нь үүнийг шийдэх энгийн механизмтай байх хэрэгтэй.



Програм хангамжийг автоматаар шинэчлэлт - Automatic update

Програмыг өөрчилж, сайжруулсаны
дараа програм зохиогчоос үл
хамааран програм хангамж нь
автоматаар өөрийгөө шинэчилдэг
байх ёстой.



Хурдан шинэчлэлт

Fast update

Зарим автоматаар хийж болохгүй шинэчлэлт, бас системийн бүхлээр нь биш зөвхөн зарим хэсгүүд, хурдан хийгддэг өөрчлөлтийг хийхэд ашиглана. Хурдан шинэчлэлд нарийн төвөгтэй үйлдлүүд хийгдэх ба магадгүй жижиг хэмжээний шинэчлэл хийдэг програм байж болно. Зарим хэрэглээний програм програмын кодыг ажиллуулдаг өөрийн гэсэн модультай байдаг.



Баримт бичиг

Програм хангамж нь системийн
классууд болон системийн тухай
бичиг баримтыг гаргадаг байх
шаардлагатай.



Програм хангамжийн бичиг баримт

Програм хангамжийг хөгжүүлэгчид
зориулсан классууд болон түүний
холбоо хамаарал, функцуудыг
тайлбарласан модультай байх ёстой.



Програм хангамжын сан – Library

- Үндсэн сан
 - Үндсэн өгөгдлийн бүтцүүд – олонлог, жагсаалт, мод, стейк
 - Үндсэн алгоритмууд – эрэмбэлэх, хайх, нэвтрэх
- График болон хэрэглэгчийн интерфeйсийн сан
- Сан хөгжүүлэх боломж