**Лекц 14 Char & String processing**

# Тооцооллын зарчим

Компьютерууд тоон өгөгдөлтэй ажиллахдаа сайн байдаг. Жишээлбэл, та int төрлийн хувьсагчийг зарлах үед Java виртуал машин нь бүхэл тоог тодорхойлсон мужид хадгалах зориулалттай санах ойд байршлыг хадгалдаг.

Гэхдээ бүхэл тоон утгыг илэрхийлэх чадвар нь бүхэл тоогоор эдгээр төрлүүдийг төлөөлөх боломжтой бол бусад өгөгдлийн төрлүүдтэй ажиллахад хялбар болгодог. Хязгаарлагдмал утгуудаас бүрдэх төрлүүдийн хувьд хамгийн хялбар арга бол цуглуулгын элементүүдийг дугаарлах явдал юм.

Элементүүдийг тоолж тодорхойлсон төрлүүдийг тоологдсон төрлүүд гэнэ.

# Тооллогын байгууламж дахь солилцоо

Тооллогын хэрэгсэл нь бүхэл тоо ашиглахаас хэд хэдэн давуу талтай:

* Хөрвүүлэгч нь бүхэл тооны кодыг автоматаар сонгоно.
* Хувьсагчийн мэдэгдлүүд нь илүү утга учиртай төрлийн нэрийг ашигладаг.
* Хөрвүүлэгч нь утгууд нь төрөлтэй тохирч байгаа эсэхийг шалгадаг.
* Төрөл бүрийн утгууд нь хэвлэх үед нэрээр гарч ирдэг.

Гэсэн хэдий ч тооллогын байгууламж нь сул талуудтай:

* Тогтмолууд нь Direction.EAST дээрх шиг төрлийг агуулсан байх ёстой.
* Хачирхалтай нь, тохиолдол заалтуудад хэрэглэгддэг тогтмолууд энэ төрлийг орхих ёстой.
* Enum ашиглан бичсэн код нь зөвхөн Java-ийн шинэ хувилбаруудад ажилладаг.
* Тооллогын байгууламж нь үнэндээ энд тайлбарласнаас хамаагүй илүү төвөгтэй юм.

Одоо байгаа Java код нь хуучин хэв маягийг ашигладаг тул та ямар ч байсан үүнийг сурах хэрэгтэй.

Номыг бичихдээ би сул тал нь давуу талаасаа ялимгүй давж, програмчлалын жишээн дэх enum байгууламжийг ашиглаагүй гэж шийдсэн.

# Тэмдэгтүүд

Компьютерууд машины санах ойн доторх тэмдэгтийн өгөгдлийг төлөөлөхийн тулд тоолох зарчмыг ашигладаг. Эцсийн эцэст гар дээр хязгаарлагдмал тооны тэмдэгтүүд байдаг. Хэрэв та тэмдэгт бүрт бүхэл тоо оноох юм бол тухайн бүхэл тоог түүний төлөөлж буй тэмдэгтийн код болгон ашиглаж болно.

Гэхдээ тэмдэгтийн кодууд нь стандартчилагдаагүй л бол тийм ч ашигтай биш юм. Хэрэв өөр өөр компьютер үйлдвэрлэгчид өөр өөр кодчиллын дарааллыг ашигладаг бол (үнэхээр эхний жилүүдэд байсан шиг) ийм өгөгдлийг машин хооронд хуваалцах нь илүү хэцүү болно.

Анхны өргөн хэрэглэгддэг тэмдэгтийн кодчилол нь ASCII байв.

# Тэмдэгтийн дүрслэлийн талаархи тэмдэглэл

Java программ дахь тэмдэгтийг зааж өгөхийн тулд та нэг хашилтанд орсон хүссэн тэмдэгтээс бүрдэх тэмдэгтийн тогтмолыг ашиглах хэрэгтэй. Тиймээс, програмын тогтмол 'A' нь том үсгийн А-г Юникод хэлбэрээр илэрхийлдэг. Энэ нь 1018 утгатай байна гэдэг нь хамааралгүй мэдээлэл юм.

Юникод хүснэгтийн хоёр шинж чанарыг онцгой анхаарах нь зүйтэй:

Цифрүүдийн тэмдэгтийн кодууд нь дараалсан байна.

* Цагаан толгойн үсгүүд нь том, жижиг үсэг гэсэн хоёр мужид хуваагддаг.
* Муж бүрийн дотор Юникод утгууд нь дараалсан байна.

# String ангид аргуудыг ашиглах

Java нь String анги дээр ажилладаг олон ашигтай аргуудыг тодорхойлдог. Эдгээр аргуудыг тусад нь ашиглахыг оролдохын өмнө эдгээр аргууд илүү ерөнхий түвшинд хэрхэн ажилладагийг ойлгох нь чухал юм.

String анги нь таныг стринг дээрх аргыг дуудах үед хүлээн авагчийн синтаксийг ашигладаг. Статик аргыг дуудахын оронд (жишээ нь Character класстай адил) Java-ийн загвар нь та мөрөнд мессеж илгээх явдал юм.

Java-ийн String анги дахь аргуудын аль нь ч хүлээн авагч болгон ашигласан мөрийн утгыг өөрчилдөггүй. Үүний оронд юу болох вэ гэвэл эдгээр аргууд нь хүссэн өөрчлөлтүүдийг хийсэн шинэ мөрийг буцаана.

Үйлчлүүлэгчид объектын төлөвийг өөрчлөхийг хориглодог ангиудыг хувиршгүй гэж нэрлэдэг. Үл өөрчлөгддөг ангиуд нь олон давуу талтай бөгөөд програмчлалд чухал үүрэг гүйцэтгэдэг.

# Холболт

Мөртүүдэд ашиглах боломжтой хамгийн ашигтай үйлдлүүдийн нэг бол хоёр мөрийг хооронд нь залгах тэмдэгтгүйгээр нэгтгэхээс бүрдэх холболт юм.

String анги нь холболтыг гүйцэтгэдэг concat хэмээх аргыг экспортлодог боловч энэ аргыг бараг ашигладаггүй. Холболтыг Java-д + оператор хэлбэрээр суулгасан.

Хэрэв та тоон операндуудтай +-г ашиглавал энэ нь нэмэхийг илэрхийлнэ. Хэрэв түүний операндуудын ядаж нэг нь тэмдэгт мөр бол Java нь +-г холболт гэж тайлбарладаг. Үүнийг ийм байдлаар ашиглах үед Java дараах алхмуудыг гүйцэтгэдэг.

Хэрэв операндуудын аль нэг нь мөр биш бол тухайн ангид toString аргыг хэрэглэж мөр болгон хөрвүүлнэ.

Утгыг нэгтгэхийн тулд concat аргыг хэрэглэнэ.

# TranslateLine дизайн

TranslateLine арга нь оруулах мөрийг үг болгон хувааж, үг бүрийг орчуулж, дараа нь тэдгээр үгсийг дахин нэгтгэх ёстой.

Хэдийгээр мөрийг үг болгон хуваах код бичих нь тийм ч хэцүү биш ч Java номын санд байгаа хэрэгслийг ашиглан энэ ажлыг гүйцэтгэхэд илүү хялбар байдаг. Нэг стратеги нь java.util багц дахь StringTokenizer классыг ашиглах явдал бөгөөд энэ нь стринг токен гэж нэрлэгддэг бие даасан нэгжүүдэд хуваадаг. Дараа нь үйлчлүүлэгч эдгээр жетонуудыг нэг нэгээр нь уншдаг. Токенизаторын илгээсэн жетонуудын багцыг токен урсгал гэж нэрлэдэг.

Токен гэж юу болох тухай нарийн тодорхойлолт нь програмаас хамаарна. Гахайн латин хэлний асуудлын хувьд токенууд нь үгс эсвэл үгсийг тусгаарлах тэмдэгтүүд бөгөөд тэдгээрийг тусгаарлагч гэж нэрлэдэг. Аппликейшн нь зөвхөн үгтэй ажиллах боломжгүй, учир нь гаралт дээр үгсийг хамт ажиллуулахгүй байхын тулд тусгаарлах тэмдэгтүүд шаардлагатай.