**Лекц 15 Event Driven Programming**

# Java үйл явдлын загвар

График програмууд нь ихэвчлэн хулгана гэх мэт оролтын төхөөрөмж ашиглан програмын үйлдлийг хянах боломжийг хэрэглэгчдэд олгодог. Энэ төрлийн хэрэглэгчийн хяналтыг дэмждэг программуудыг интерактив программ гэж нэрлэдэг.

Хулгана дарах зэрэг хэрэглэгчийн үйлдлийг үйл явдал гэж нэрлэдэг. Үйл явдалд хариу үйлдэл үзүүлэх хөтөлбөрүүдийг үйл явдалд тулгуурласан гэж нэрлэдэг.

Орчин үеийн интерактив программуудад хэрэглэгчийн оролт нь урьдчилан таамаглах боломжтой цагт хийгддэггүй. Ажиллаж буй програм нь хулганыг хэзээ товшихыг хэрэглэгчдэд заадаггүй. Хэрэглэгч хулганаа хэзээ товшихыг шийддэг бөгөөд програм нь хариу үйлдэл үзүүлдэг. Үйл явдал хөтөлбөрөөр хянагддаггүй учир асинхрон гэж хэлдэг.

Та Java програм бичихдээ ямар нэг объектыг тухайн үйл явдлын сонсогчоор томилж хариу өгөхийг хүсч буй үйл явдлуудаа зааж өгнө. Үйл явдал тохиолдоход сонсогчдод мессеж илгээгддэг бөгөөд энэ нь зохих хариуг өдөөдөг.

# Event Төрөл

Java үйл явдлууд нь маш олон төрөлтэй байдаг. Энэ номонд ашигласан үйл явдлын төрлүүдэд дараахь зүйлс орно.

* Хэрэглэгч хулганыг хөдөлгөх эсвэл дарахад тохиолддог хулганы үйл явдлууд
* Хэрэглэгч гар дээр бичих үед гардаг гарын үйл явдлууд
* Хэрэглэгчийн интерфэйсийн үйлдлийн хариуд тохиолддог үйлдлийн үйл явдлууд

Үйл явдлын төрөл бүр сонсогчдод хэрхэн хариу үйлдэл үзүүлэхийг тодорхойлсон аргуудтай холбоотой байдаг. Эдгээр аргуудыг үйл явдлын төрөл тус бүрээр сонсогч интерфейст тодорхойлсон.

Жишээлбэл, хулганы сонсогч интерфейсийн аргуудын нэг нь mouseClicked юм. Таны бодож байсанчлан Java хулганыг дарахад энэ аргыг дууддаг.

MouseClicked гэх мэт сонсогч аргууд нь үйл явдлын талаар нэмэлт мэдээлэл агуулсан параметрийг авдаг. MouseClicked-ийн хувьд аргумент нь MouseEvent бөгөөд товшилт хийсэн байршлыг заана.

# MOUSElistener acm дотор

Илүү нарийвчилсан түвшинд Java-ийн хулгана сонсогчдод хандах хандлага нь өмнөх слайдаас харахад тийм ч хялбар биш юм. Үр ашгийг нэмэгдүүлэхийн тулд Java нь хоёр өөр сонсогч интерфейсийг тодорхойлдог:

* MouseListener интерфейс нь хулганы товчийг дарах гэх мэт харьцангуй ховор тохиолддог хулганы үйл явдлуудад хариу үйлдэл үзүүлдэг.
* MouseMotionListener интерфэйс нь хулганыг зөөх эсвэл чирэхэд илүү хурдан гал гарах үйл явдлуудад хариу үйлдэл үзүүлдэг.

ACM Java номын сангийн багцууд нь хулгана сонсогчдыг ашиглахад хялбар болгох үүднээс дараах стратегийг хэрэгжүүлдэг.

* GraphicsProgram анги нь MouseListener болон MouseMotionListener интерфэйс дэх аргууд бүрийн хоосон тодорхойлолтуудыг агуулдаг. Ингэснээр та эдгээр бүх аргуудыг тодорхойлох шаардлагагүй, харин шаардлагатай аргуудыг зүгээр л дарж болно гэсэн үг юм.
* GraphicsProgram анги нь мөн addMouseListeners аргыг тодорхойлдог бөгөөд энэ нь програмыг хоёр төрлийн үйл явдлын сонсогч болгон нэмдэг.

# Үйлдлийн үйл явдлыг илрүүлэх

Үйлдлийн үйл явдлыг илрүүлэхийн өмнө дэлгэцэн дээрх товчлууруудын үйлдэл сонсогчийг идэвхжүүлэх хэрэгтэй. Хамгийн хялбар стратеги бол init аргын төгсгөлд addActionListeners-ийг дуудах явдал юм. Энэ дуудлага нь дэлгэц дээрх бүх товчлуурын сонсогчоор програмыг нэмдэг.

Та товчлуур бүрийн зөв үйлдлийг хэрэгжүүлдэг шинэ хувилбараар Performed-ийн тодорхойлолтыг дарж товчлуур дээр дарахад хариу өгөхийг зааж өгнө.

# Тоон талбарууд

acm.gui багц нь график хэрэглэгчийн интерфэйс доторх тоон оролтыг унших үйл явцыг хялбаршуулдаг JTextField хоёр дэд ангийг агуулдаг. IntField анги нь текстийн мөрийг int гэж тайлбарладаг; DoubleField анги нь текстийн мөрийг давхар гэж тайлбарладаг.

JTextField дээрх ердийн үйлдлүүдээс гадна IntField болон DoubleField ангиуд нь талбарын тоон утгыг авч тохируулах getValue болон setValue аргуудыг экспортлодог.

Хэдийгээр энэ нь текстийн хамрах хүрээнээс хэтэрсэн ч IntField болон DoubleField ангиуд нь тоон форматыг дэмждэг тул дэлгэцэн дээрх цифрүүдийн тоог хянах боломжтой. Энэ чадварыг дэмжих аргуудыг эдгээр ангиудын javadoc баримт бичигт тайлбарласан болно.

# Бүрэлдэхүүн хэсгийн зохион байгуулалтыг удирдах

Хөтөлбөрийн ирмэгийн дагуух хяналтын туузыг ашиглах нь энгийн хэрэглээнд тустай ч илүү боловсронгуй хэрэглэгчийн интерфэйсийг бий болгоход та цонхны аль ч хэсэгт харилцан үйлчлэгч байрлуулах чадвартай байхыг шаарддаг.

Гоёмсог, хэрэглэхэд хялбар интерфэйс үүсгэхийн тулд интерактивуудыг зохион байгуулах нь дизайны хэцүү сорилт юм. Энэ төрлийн дизайныг төвөгтэй болгодог нэг хүчин зүйл бол програмын цонхны хэмжээ цаг хугацааны явцад өөрчлөгдөж болох явдал юм.

Java нь цонхны хэмжээ өөрчлөгдөхөд интерактив болон бусад бүрэлдэхүүн хэсгүүдийг зохион байгуулах үүрэгтэй зохион байгуулалтын менежерүүдийг ашиглан цонхны хэмжээг өөрчлөх асуудлыг шийдэхийг эрэлхийлдэг.

# Бүрэлдэхүүн хэсэг ба савнууд

Хэрэв та эхлээд Java-ийн цонхны системийн үндсэн хэсэг болох Бүрэлдэхүүн ба Контейнер гэсэн хоёр ангиллын хоорондын хамаарлыг ойлговол байршлын менежерүүд хэрхэн ажилладагийг ойлгоход илүү хялбар болно.

Component анги нь Java-ийн цонхны системийн шатлалын үндсийг бүрдүүлдэг, учир нь цонхонд гарч буй бүх зүйл нь Component-ийн дэд анги юм.

Container анги нь бусад бүрэлдэхүүн хэсгүүдийг агуулж болох бүрэлдэхүүн хэсгийн дэд ангилал бөгөөд ингэснээр дурын гүнтэй бүтэц дотор бүрэлдэхүүн хэсгүүдийг үүрлэх боломжтой болгодог.

Дараагийн слайд дээрх шаталсан диаграмаас харахад текст дээр үзсэн олон ангиуд нь Бүрэлдэхүүн хэсэг болон Контейнерийн дэд ангиуд юм. Ялангуяа, бүх Swing интеракторууд, GCanvas анги, Програмын ангиуд нь бүрэлдэхүүн хэсэг ба контейнер юм.