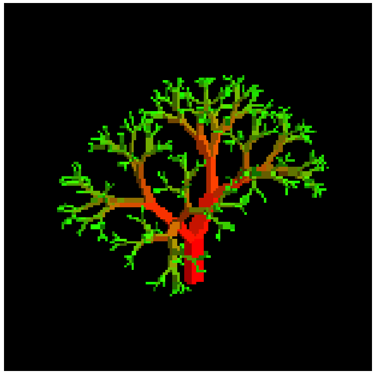
F.CS203 - Өгөгдлийн бүтэц ба алгоритм 2022-2023

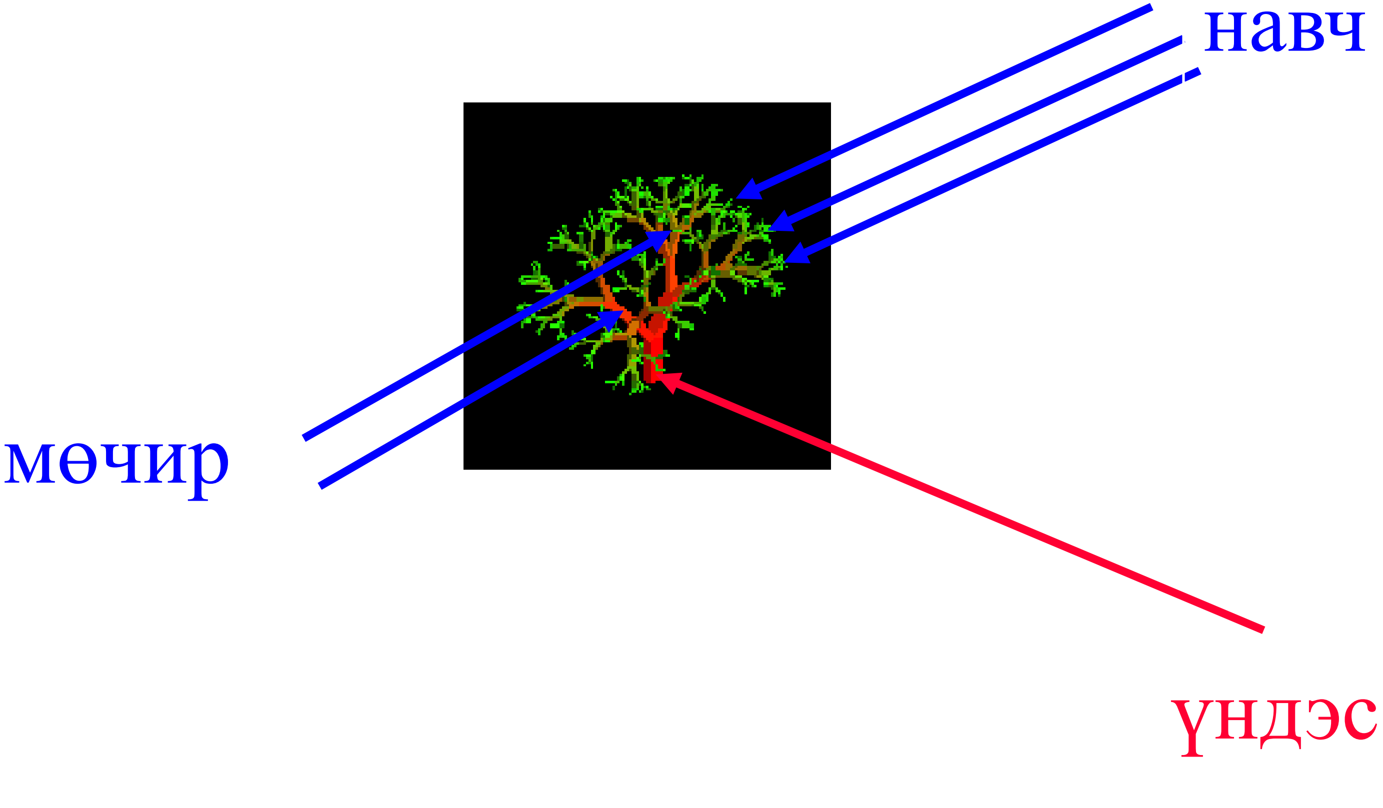
Д.Золбоо

МТ-н салбарын багш

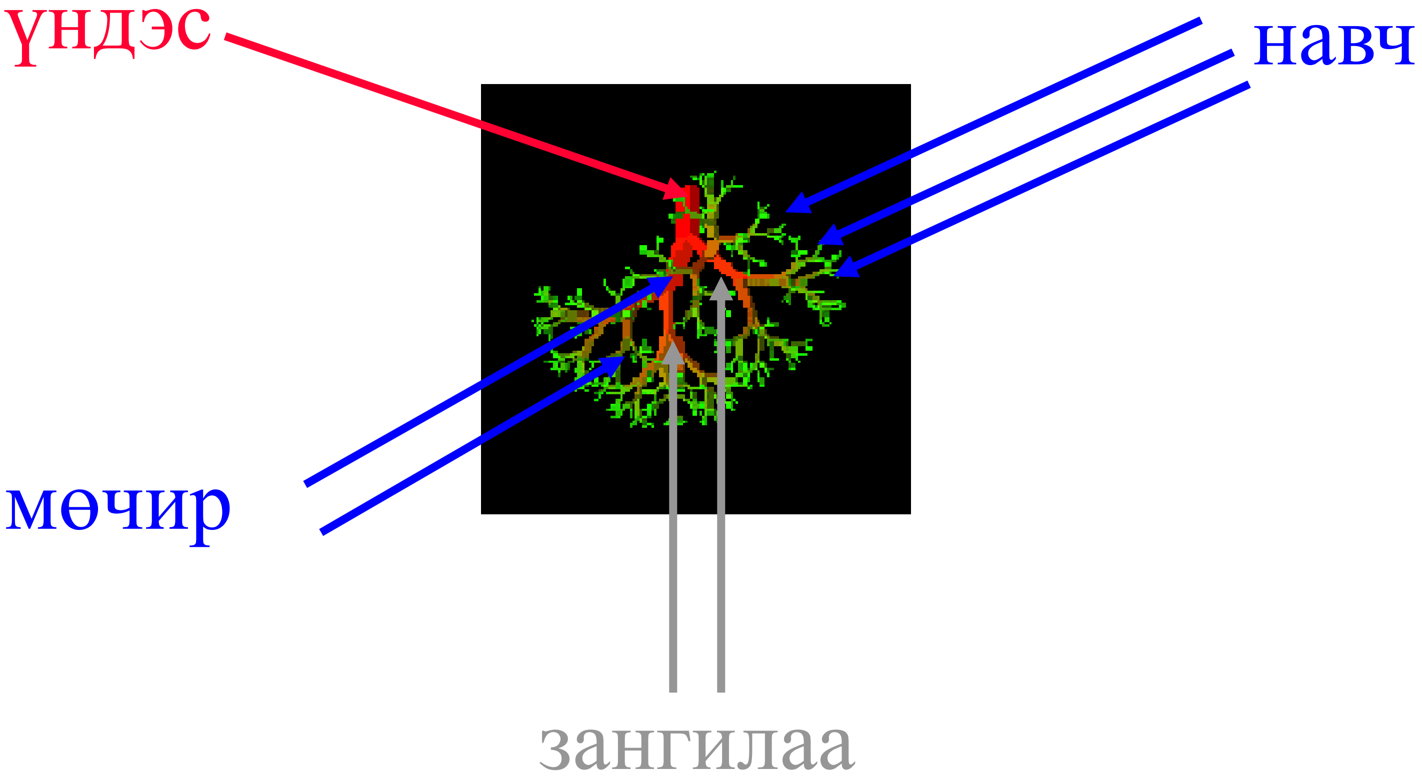
Мод



Байгальд хайртай хүний нүдээр



Компьютерийн хүний нүдээр



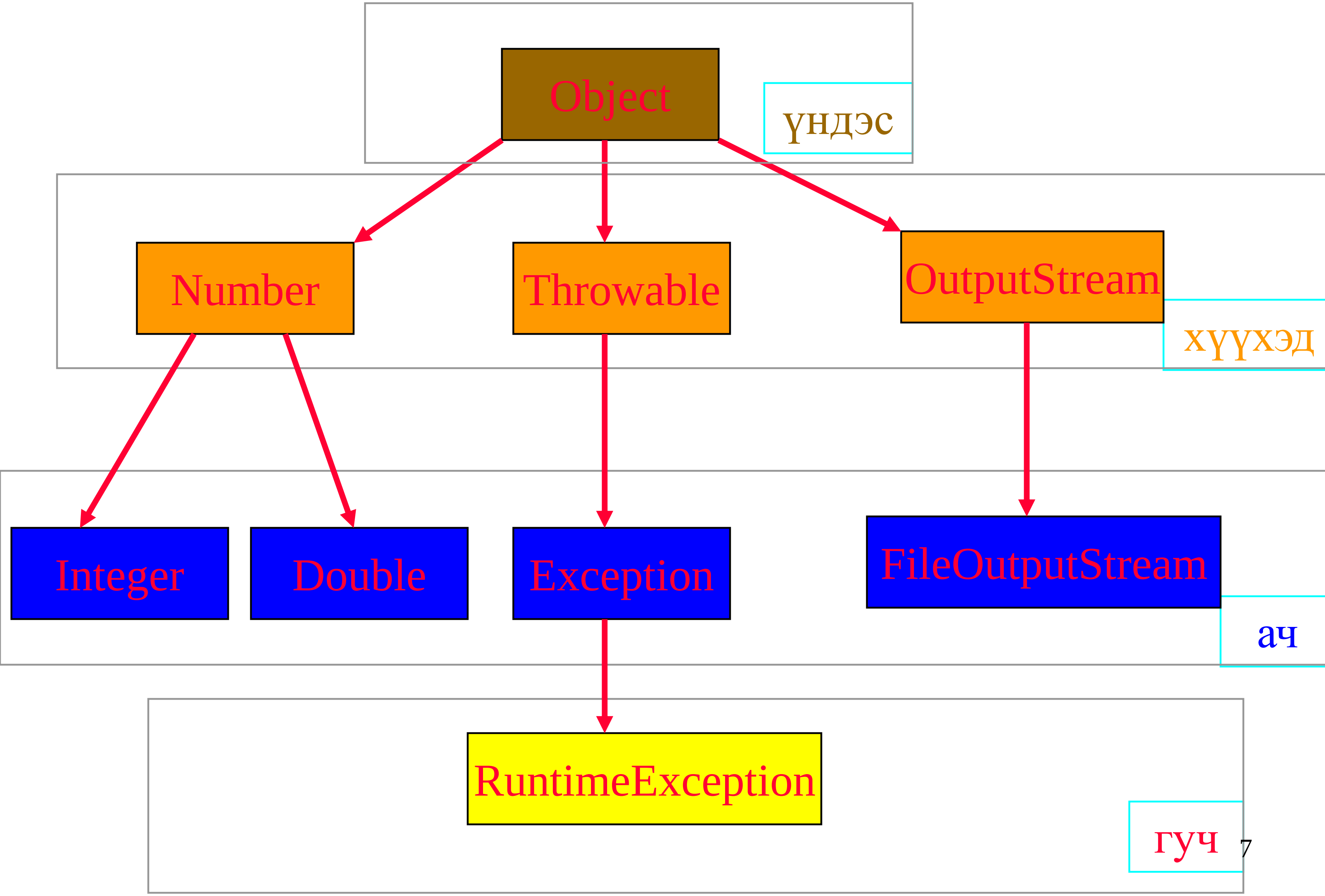
Шугаман жагсаалт ба Мод

- Шугаман жагсаалт цуварч эрэмбэлэгдсэн өгөгдөлд тохиромжтой.
  - (e_0, e_1, e_2, …, e_{n-1})
  - Долоо хоногийн өдрүүд.
  - Жилийн сарууд.
  - Ангийн оюутнууд.
- Мод үелэж эрэмбэлэгдсэн өгөгдөлд тохиромжтой.
  - Байгуулгын ажиллагсад.
    - Захирал, дэд захирал, менежер, гэх мэт.
  - Java-ийн классууд.
    - Object үечлэлийн оройд байдаг.
    - Object –ийн дэд классууд дараа нь, гэх мэт .

Үелсэн өгөгдөл ба Мод

- Дээд/оройн үеийн элемент бол root-үндэс.
- Хоёрдахь үеийн элементүүд бол үндсээс гарсан children-хүүхдүүд.
- Гуравдахь үеийн элементүүд бол үндсээс гарсан grandchildren-ач нар, гэх мэт.
- Хүүхэдгүй элемент бол leaves-навчис.

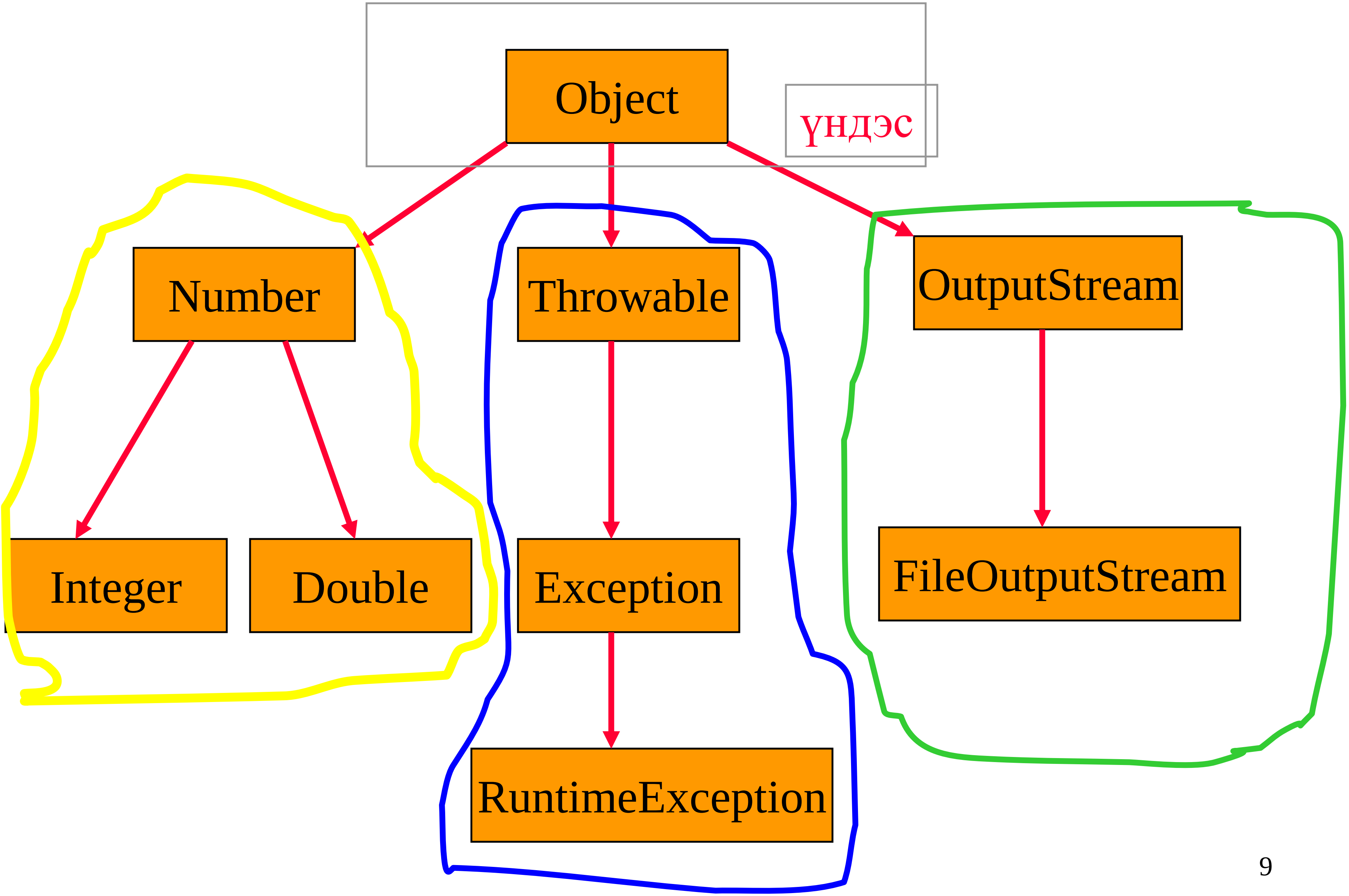
Java-ийн классууд



Тодорхойлолт

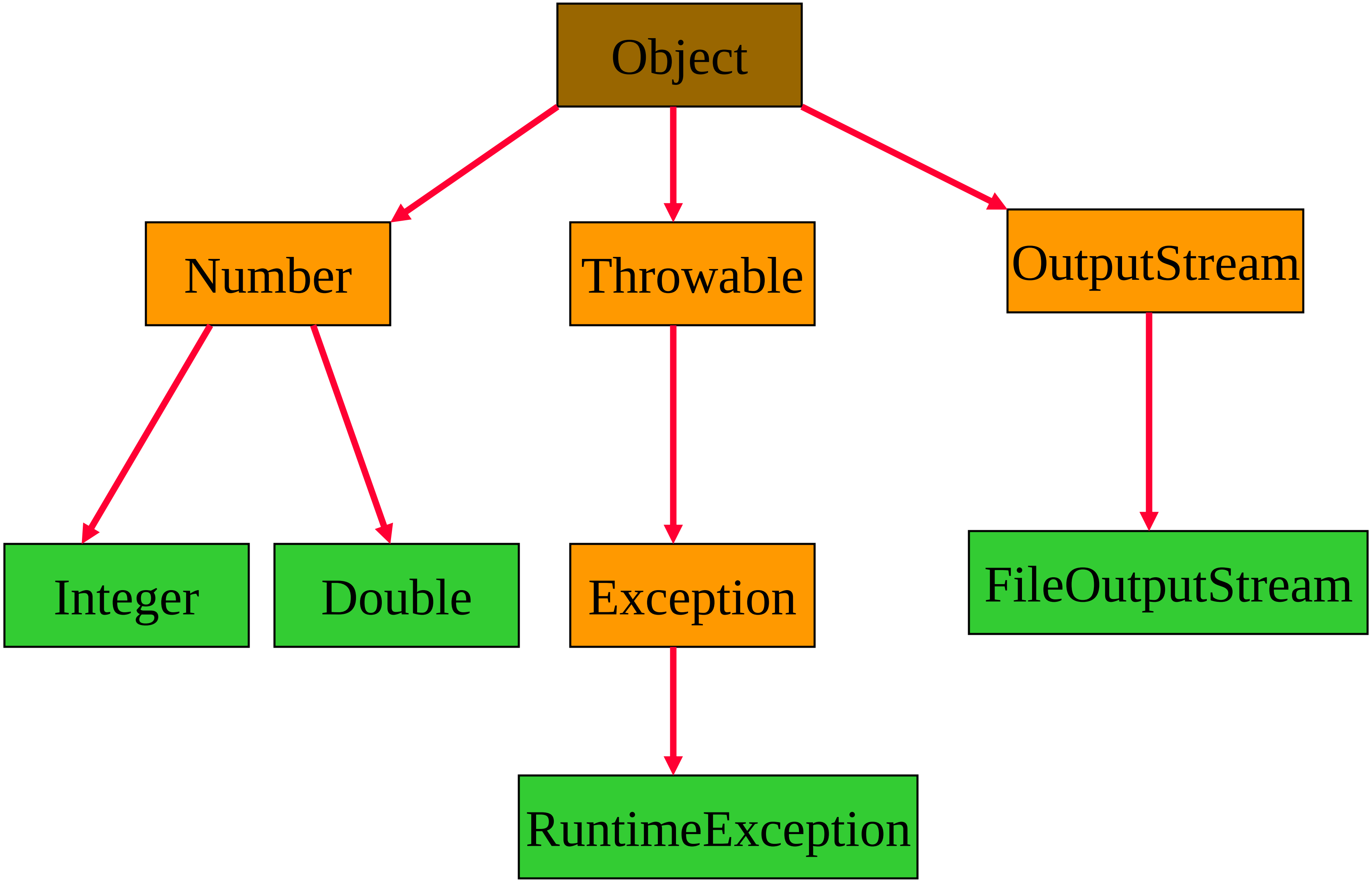
- Мод t элементүүдийн хоосон бус төгсгөлтэй олонлог.
- Элементүүдийн нэгийг нь үндэс гэнэ.
- Бусад элементүүд(хэрвээ байгаа бол) мод болж хуваагдана, түүнийг t -ийн дэд мод гэнэ

Дэд мод

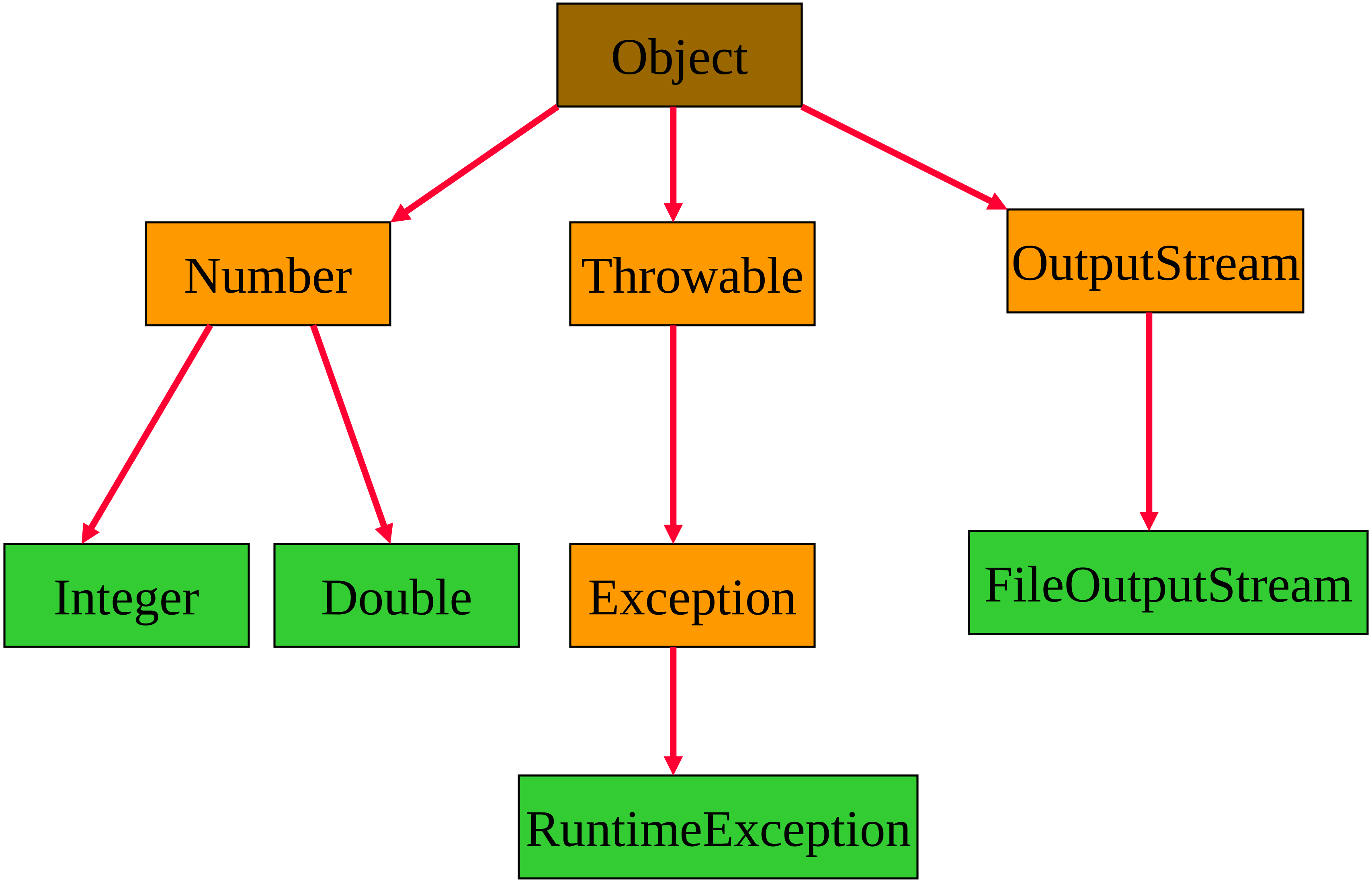


Object
үндэс
Number
Throwable
OutputStream
Integer
Double
Exception
FileOutputStream
RuntimeException

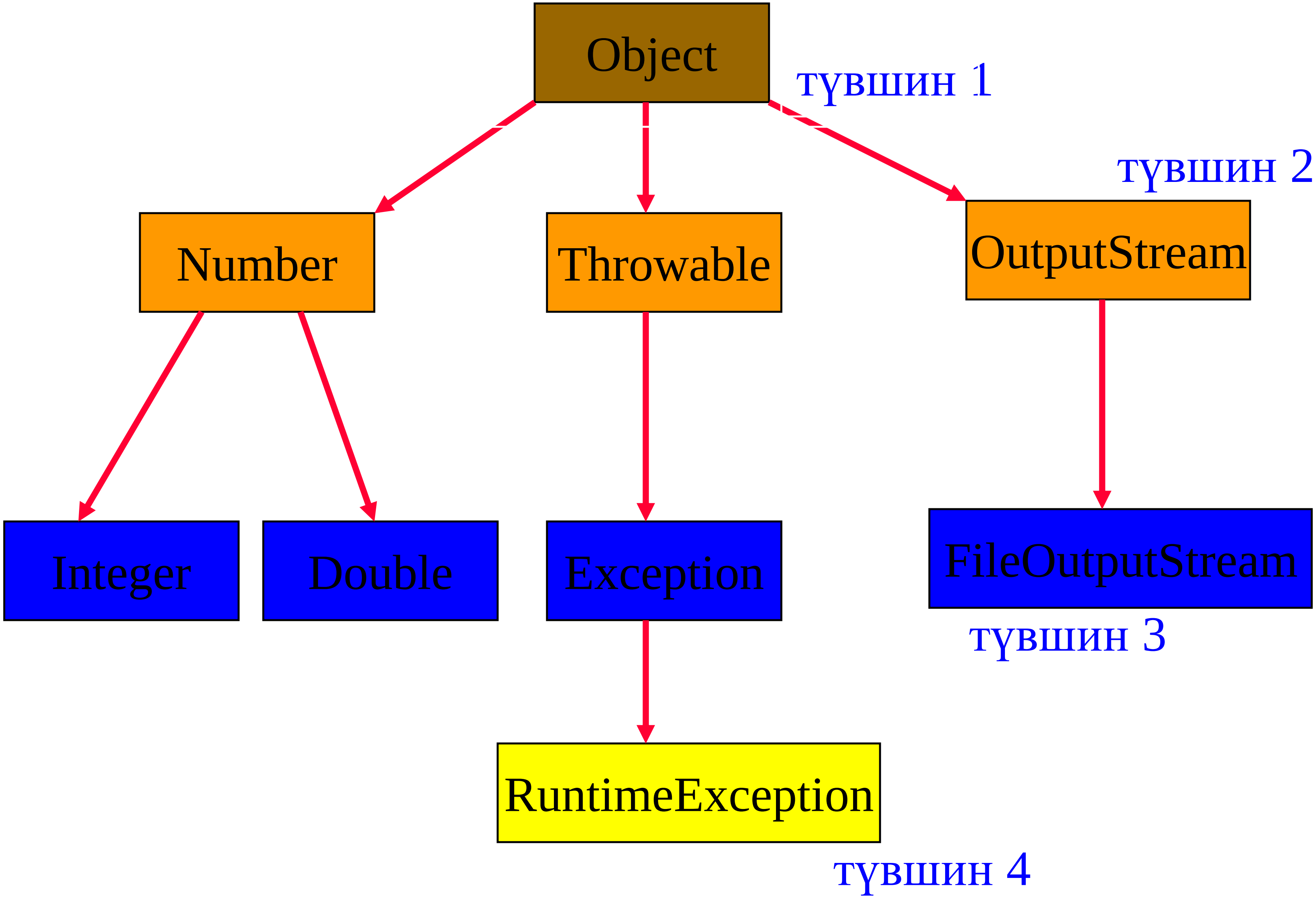
Навчис


Object
Number
Throwable
OutputStream
Integer
Double
Exception
FileOutputStream
RuntimeException

Parent, Grandparent, Siblings, Ancestors, Descendants – Эцэг, Өвөг эцэг, Ах дүүс, Удам, Хойч



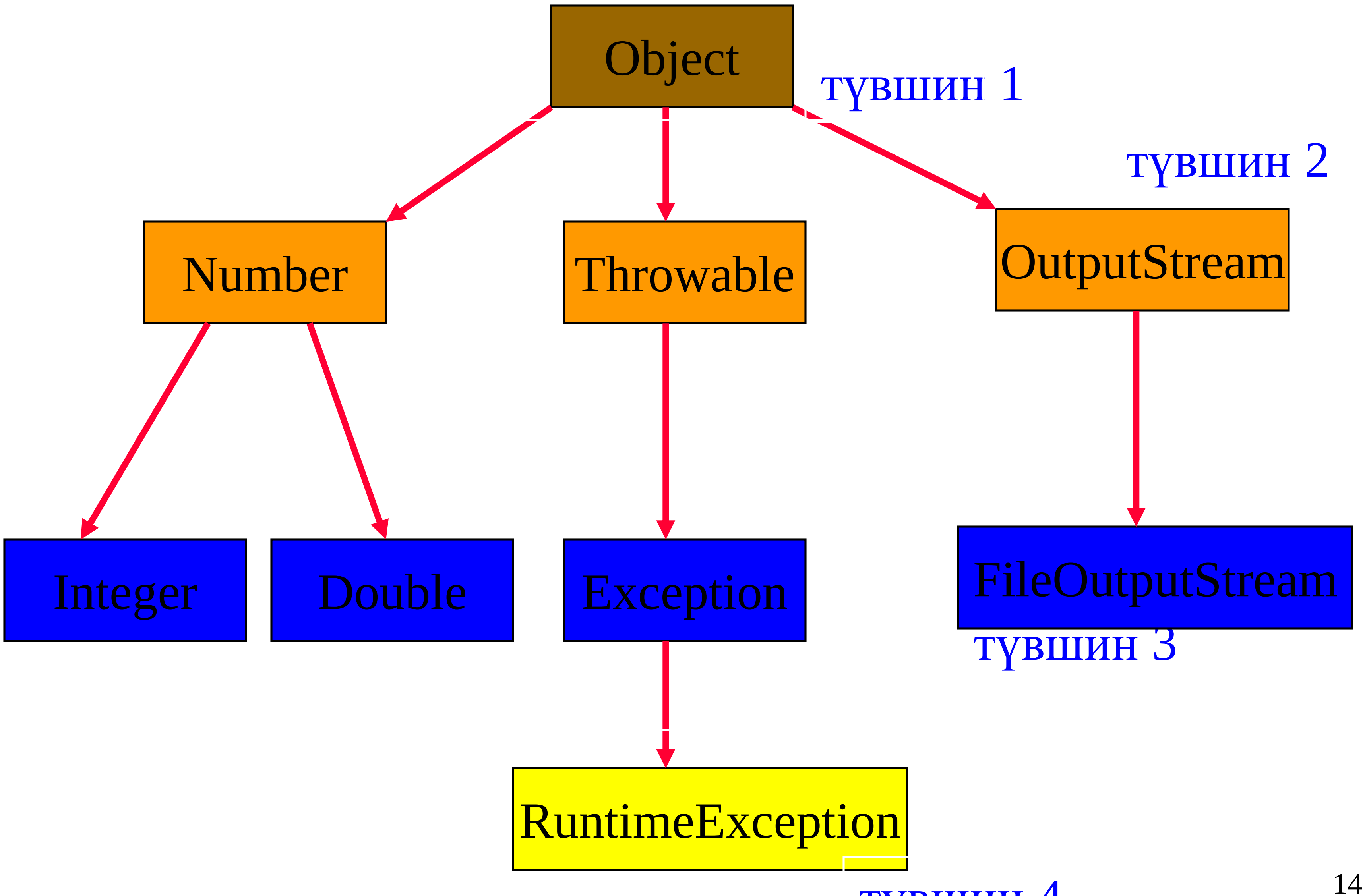
Түвшин-үе


Object
түвшин 1
Number
Throwable
OutputStream
түвшин 2
Integer
Double
Exception
FileOutputStream
түвшин 3
RuntimeException
түвшин 4

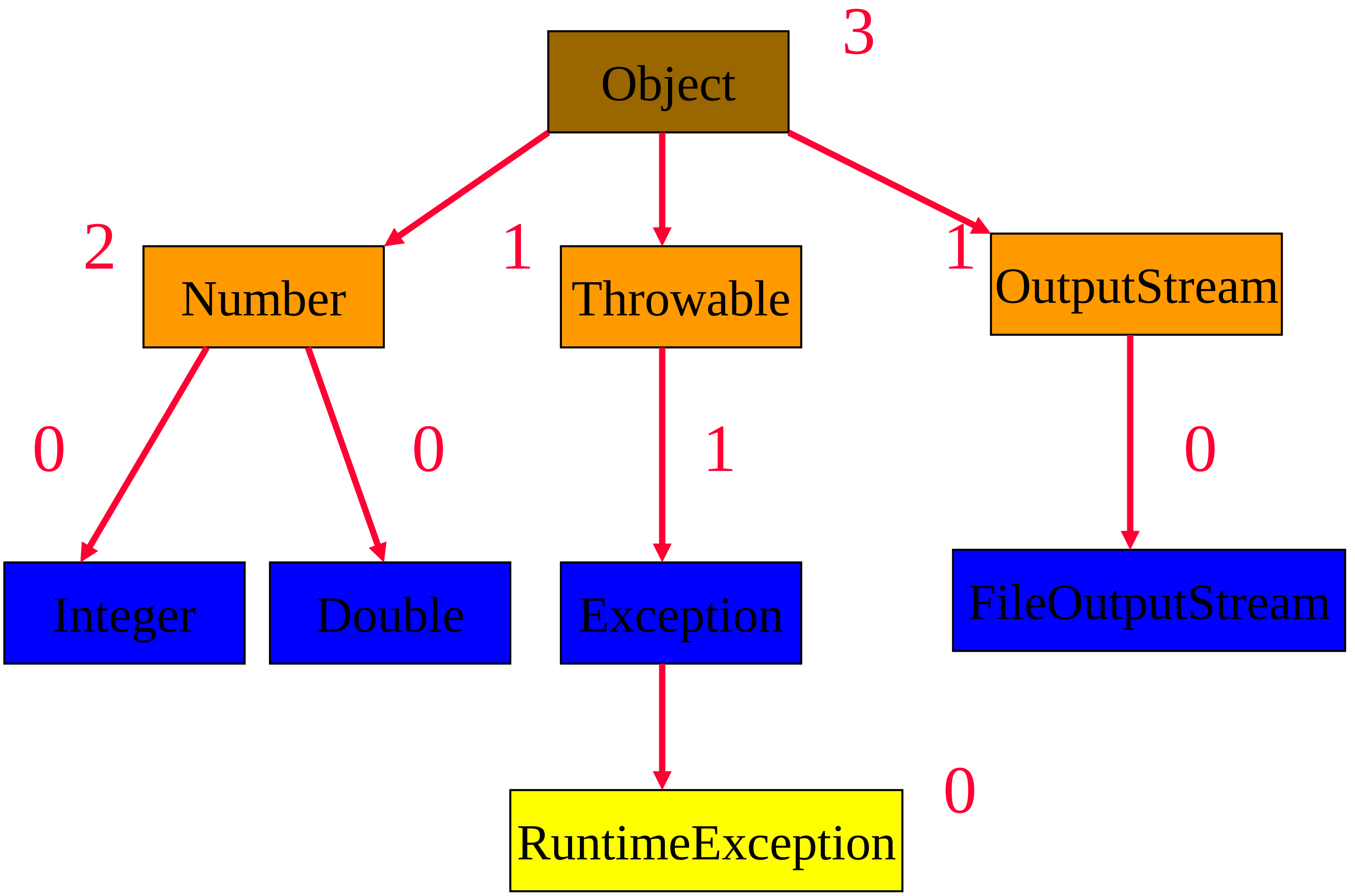
Анхааруулга

- Зарим номонд түвшинг 1 -ээс биш 0 –ээс эхэлж дугаарладаг.
- Үндэсний түвшин 0.
- Түүний хүүхдийн түвшин 1.
- Ач/зээгийн түвшин 2.
- Гэх мэт.
- Бид түвшинг 1 -ээс эхэлж дугаарлана

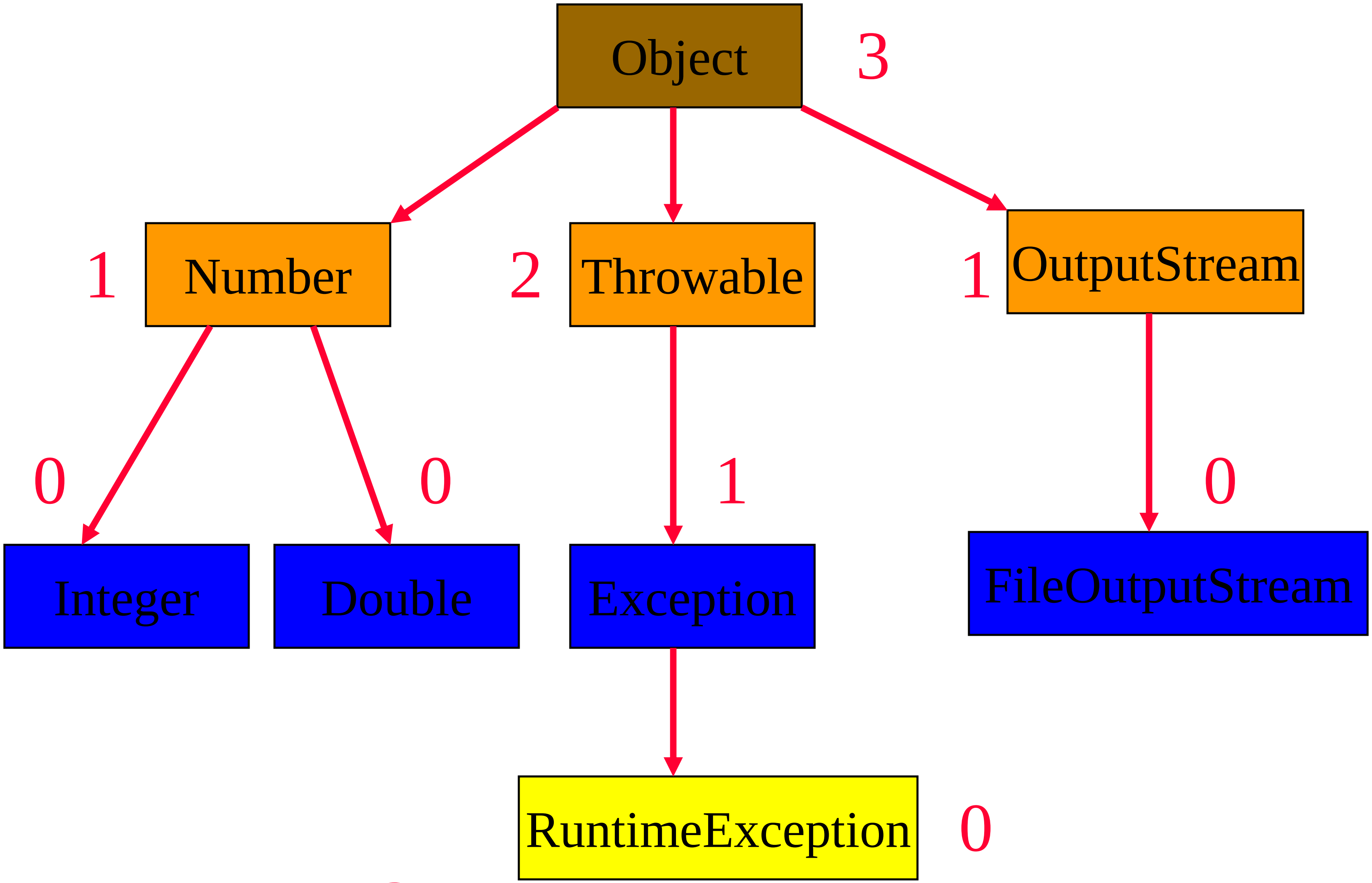
height = depth = түвшний тоо



Зангилааны зэрэг = Хүүхдийн тоо



Модны зэрэг = Max(Зангилааны зэрэг)



Модны зэрэг = 3.

Хоёртын мод

- Элементүүдийн төгсгөлтэй (хоосон байж болно) цуглуулга.
- Хоосон бус хоёртын мод root-үндэс элементтэй.
- Бусад элементүүд (байгаа бол) хоёр хоёртын модонд хуваагдана.
- Тэднийг хоёртын модны left-зүүн ба right-баруун дэд моднууд гэнэ.

Мод, Хоёртын модны ялгаа

- Хоёртын модны зангилааны зэрэг 2 -оос ихгүй байхад <> Хязгааргүй.
- Хоёртын мод хоосон байж болно <>Хоосон бус.

Мод, Хоёртын модны ялгаа

- Хоёртын модны дэд моднууд эрэмбэлэгдсэн <> Эрэмбэлэгдээгүй.

- Хоёртын мод гэж харвал ялгаатай.
- Мод гэж харвал адилхан.

Арифметик илэрхийлэл

- (a+b)*(c+d) + e – f/g*h + 3.25

- Илэрхийлэл 3 зүйлийг нэгтгэдэг:
  - Үйлдэл (+,-,/,*)
  - Гишүүд (a,b,c,d,e,f,g,h,3.25, (a+b),(c+d), гэх мэт)
  - Зааглагч ((,)).

Үйлдлийн зэрэг

- Үйлдэлд орох гишүүдийн тоо.
- Хоёр гишүүнтэй үйлдэл. Binary Operator
  - a + b
  - c / d
  - e – f
- Нэг гишүүнтэй үйлдэл. Unary Operator
  - + g
  - - h

Infix хэлбэр

- Илэрхийлэл бичих ердийн арга.
- Хоёр гишүүнтэй үйлдэл зүүн, баруун гишүүний хоорондн бичигдэнэ.
  - a * b
  - a + b * c
  - a * b / c
  - (a + b) * (c + d) + e – f/g * h + 3.25

Үйлдлийн ахлах чанар

- Үйлдлүүд яаж хийгдэх вэ?
  - a + b * c
  - a * b + c / d
- Priority-Үйлдлийн ахлах чанараар зохицуулагдана.
  - priority(*)=priority(/) > priority(+) = priority(-)

❿Гишүүн хоёр үйлдлийн хооронд байвал илүү ахлах чанартай үйлдэлд харъяалагдана.

Tie Breaker – Хайнцааг таслах

- Гишүүн ижил ахлах чанартай хоёр үйлдлийн хооронд байвал зүүн талын үйлдэлд харъяалагдана.
  - a + b – c
  - a * b / c / d

Зааглагч

- Зааглагчийн дотор бичигдсэн дэд илэрхийллийг нэг гишүүн гэж үзнэ.
  - (a + b) * (c - d) / (e – f)

Infix илэрхийлэл задлан хийхэд хэцүү

- Үйлдлийн ахлах чанар, хайнцааг таслах, зааглагч шаардлагатай.
- Энэ нь компьютерийн бодолтыг хүндрүүлдэг.
- Postfix болон prefix илэрхийллийн хэлбэрүүд үйлдлийн ахлах чанар, хайнцаа таслах, зааглагчаас хамаарахгүй.
- Иймд эдгээр хэлбэрийн илэрхийллийг компьютер амархан боддог.

Postfix хэлбэр

- Хувьсагч, тогтмолууд адилхан бичигдэнэ.
  - a, b, 3.25
- Үйлдлийн гишүүдийн дараалал Infix, Postfix хэлбэрүүдэд адилхан.
- Үйлдлүүд postfix хэлбэрийн гишүүдийнхээ ард шууд бичигддэг.
  - Infix = a + b
  - Postfix = ab+

Postfix жишээ

- Infix = a + b * c
  - Postfix = a b c * +
- Infix = a * b + c
  - Postfix = a b * c +
- Infix = (a + b) * (c - d) / (e + f)
  - Postfix = a b + c d – * e f + /

Нэг гишүүнтэй үйлдэл

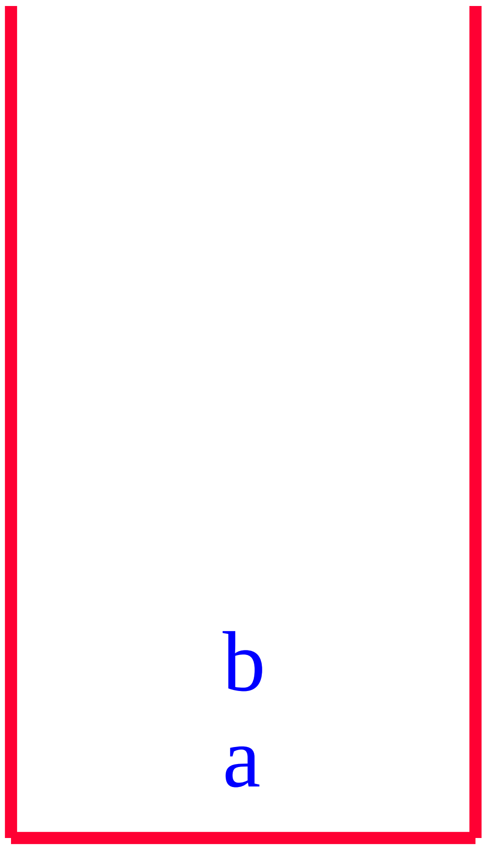
- Шинэ тэмдгээр орлуулна.
  - + a => a @
  - + a + b => a @ b +
  - - a => a?

Postfix бодолт

- Postfix илэрхийллийг зүүнээс баруун тийш шинжихдээ гишүүдийг стект хийнэ.
- Үйлдэл тааралдвал стекээс хэрэгтэй гишүүдээ аваад үйлдлийг гүйцэтгэж хариуг стект хийнэ.
- Postfix-д үйлдэл гишүүдийнхээ араас ордог болохоор энэ арга ажиллана.

Postfix бодолт

- (a + b) * (c – d) / (e + f)
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /



стек

Postfix бодолт

- (a + b) * (c – d) / (e + f)
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /

d
c
(a + b)

стек

Postfix бодолт

- (a + b) * (c – d) / (e + f)
- a b + c d - * e f + /
- a b + c d - * e f + /

(c – d)

(a + b)

стек

Postfix бодолт

- (a + b) * (c – d) / (e + f)
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /

f

e

(a + b)*(c – d)

стек

Postfix бодолт

- (a + b) * (c – d) / (e + f)
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /

(e + f)
(a + b)*(c – d)

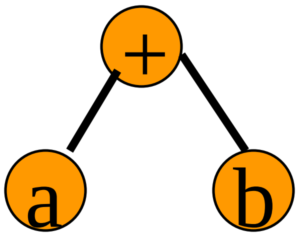
стек

Prefix хэлбэр

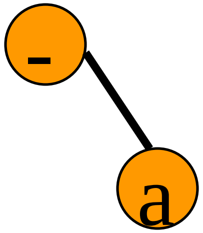
- Хувьсагч, тогтмолууд адилхан бичигдэнэ.
  - a, b, 3.25
- Үйлдлийн гишүүдийн дараалал infix, prefix хэлбэрүүдэд адилхан.
- Үйлдлүүд postfix хэлбэрийн гишүүдийнхээ өмнө шууд бичигдэнэ.
  - Infix = a + b
  - Postfix = ab+
  - Prefix = +ab

Хоёртын модны хэлбэр

- a + b

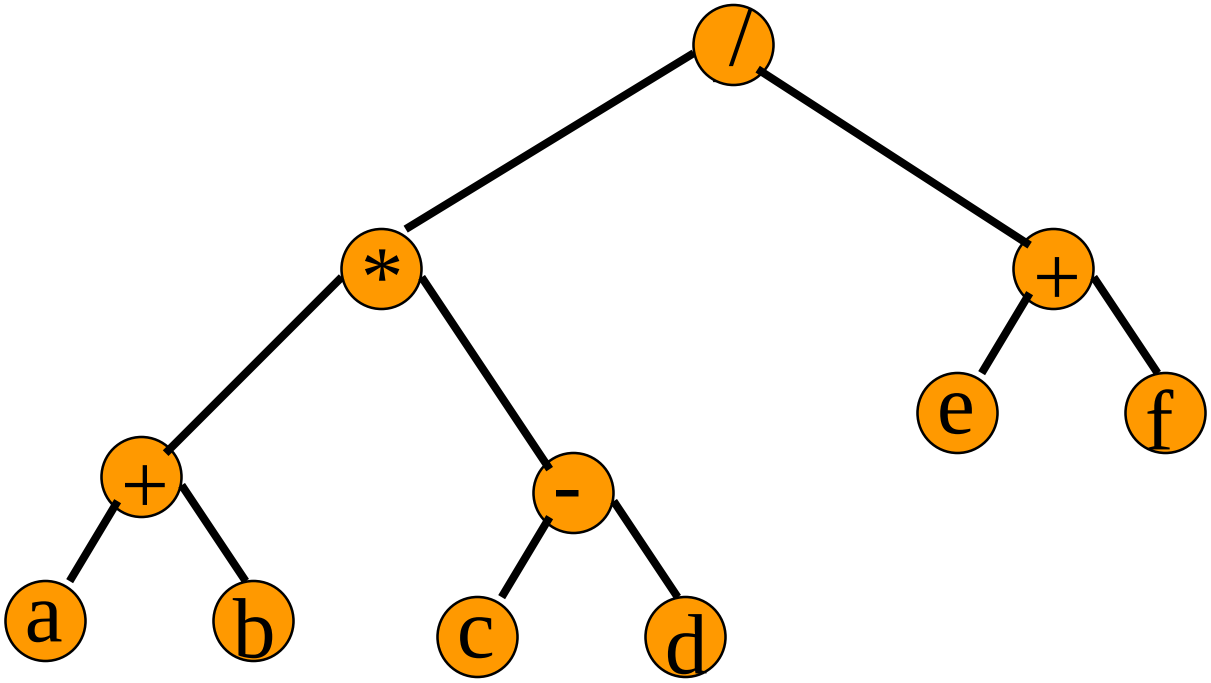


- - a



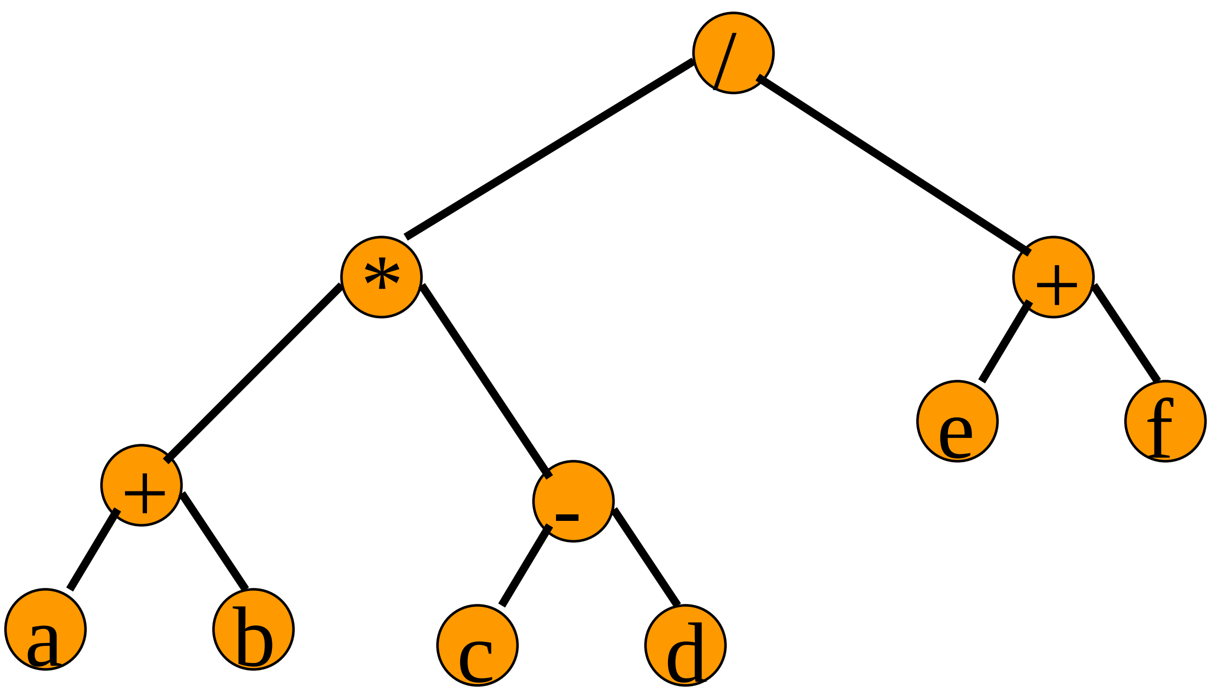
Хоёртын модны хэлбэр

- (a + b) * (c – d) / (e + f)

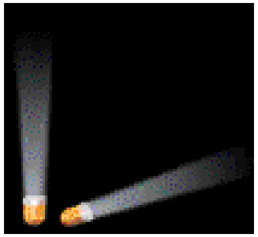
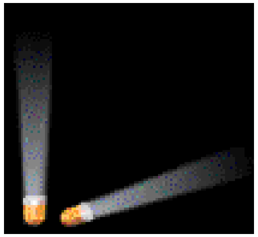
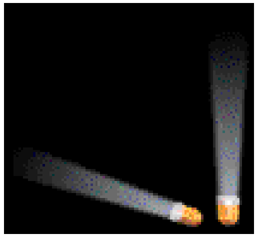
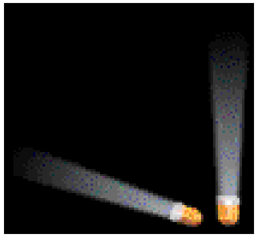


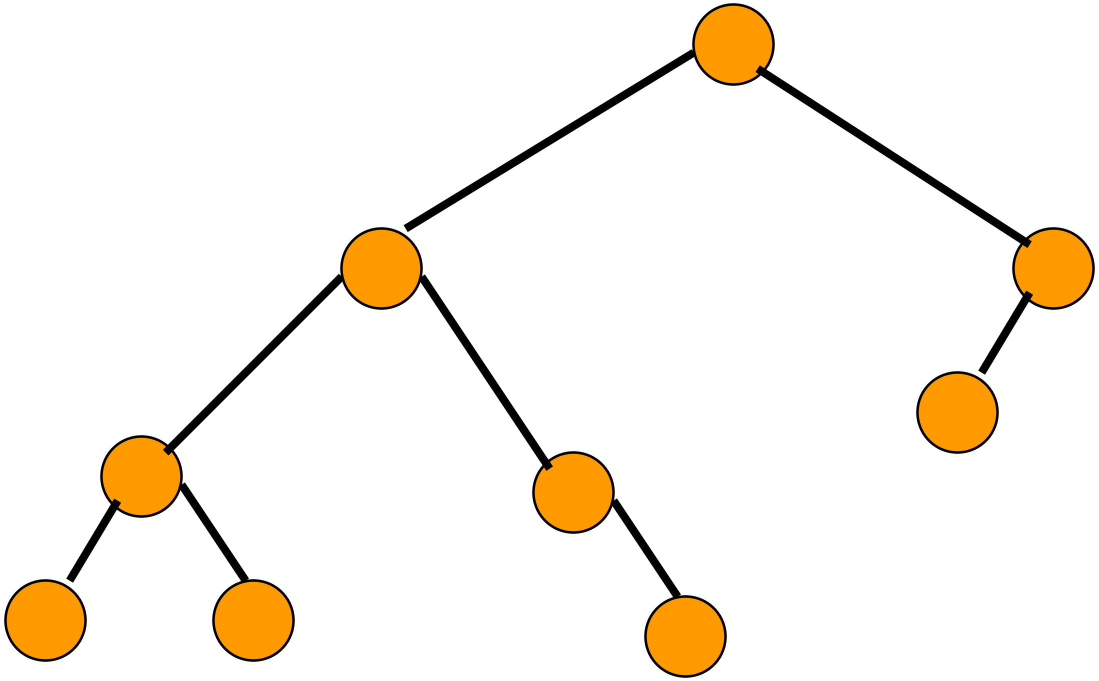
Хоёртын модны хэлбэрийн давуу тал

- Үйлдлийн зүүн, баруун гишүүд ил харагддаг.
- Илэрхийллийн хоёртын модны хэлбэр дээр кодыг оновчлох алгоритм сайн ажилладаг.
- Илэрхийллийг рекурсив аргаар бодоход хялбар.



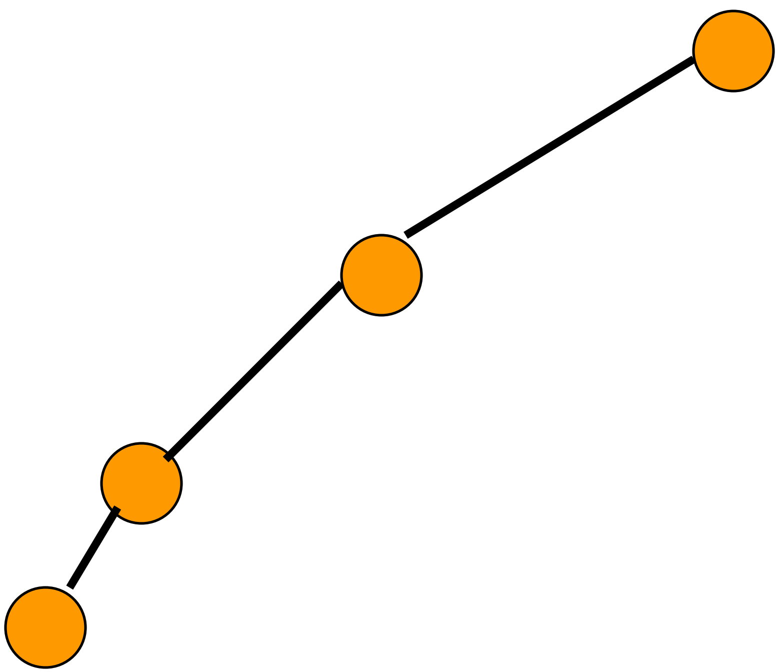
Хоёртын модны шинж ба дүрслэл



Минимум зангилааны тоо

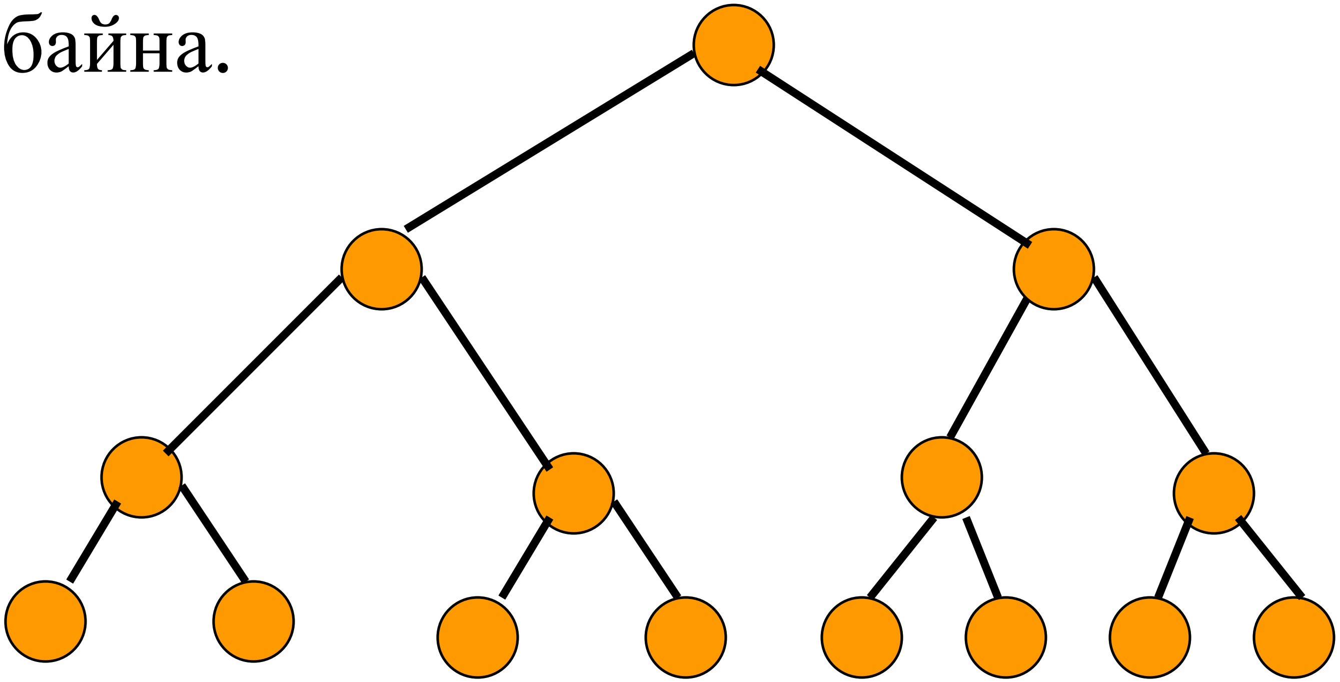
- h өндөртэй хоёртын модны минимум зангилааны тоо
- Эхний h түвшин тус бүрд ядаж нэг зангилаа байна



Минимум зангилааны тоо h

Максимум зангилааны тоо

- Эхний h түвшин бүрт боломжит бүх зангилаа байна.



Максимум зангилааны тоо

$= 1 + 2 + 4 + 8 + \ldots + 2^{h-1}$

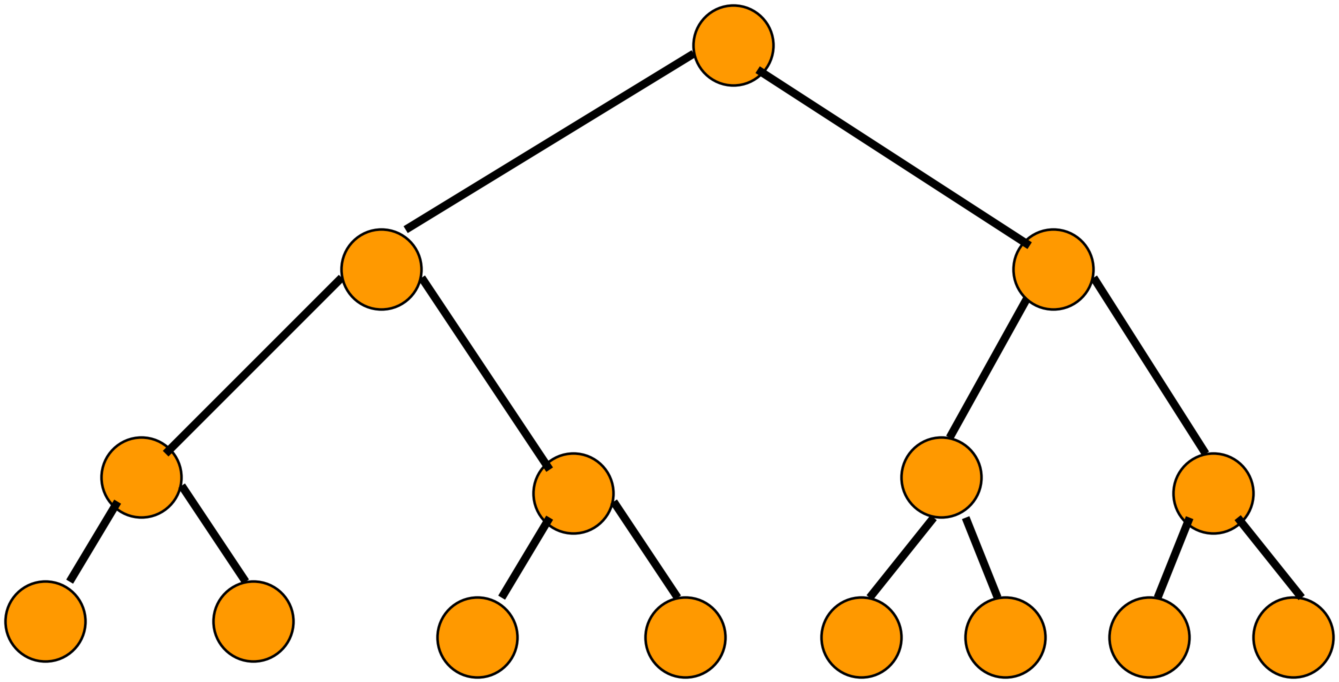
$= 2^h - 1$

Зангилааны тоо ба Өндөр

- Хэрвээ n нь h өндөртэй хоёртын модны зангилааны тоо бол:
- $h <= n <= 2^h - 1$
- $\log_2(n+1) <= h <= n$

Хоёртын бүтэн мод

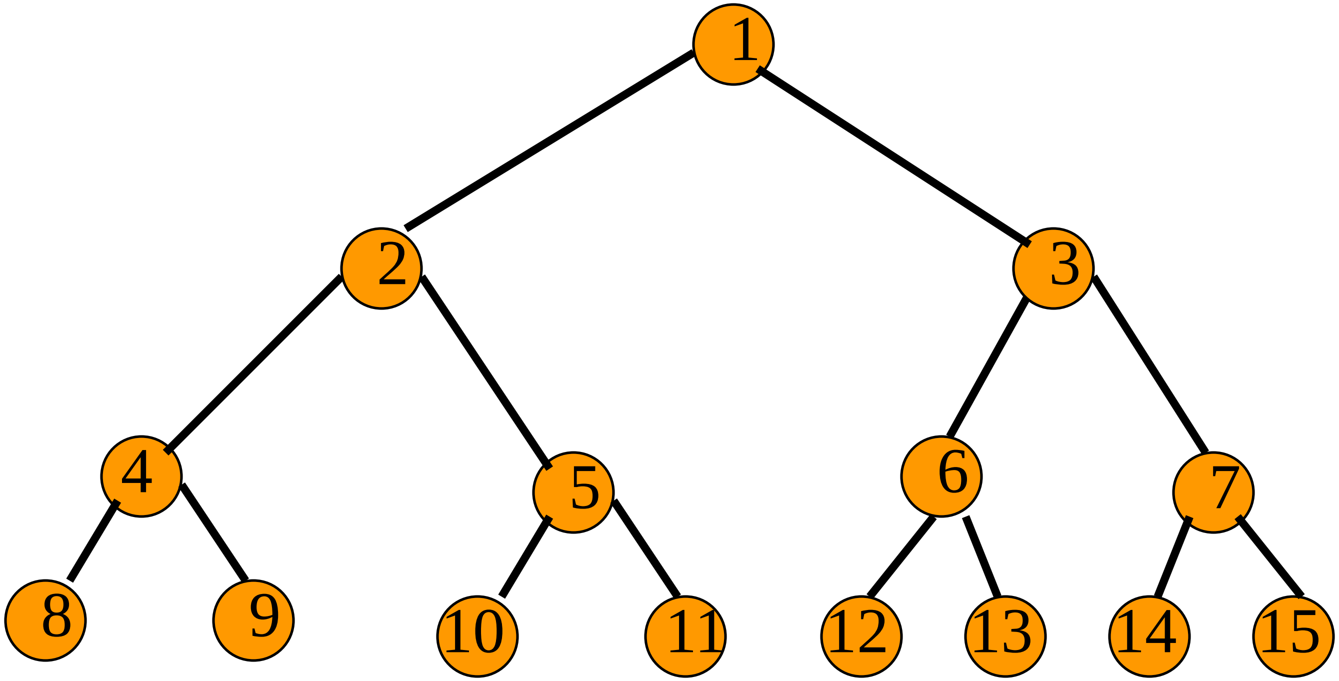
- h өндөртэй хоёртын бүтэн модонд $2^h - 1$ зангилаа байна.



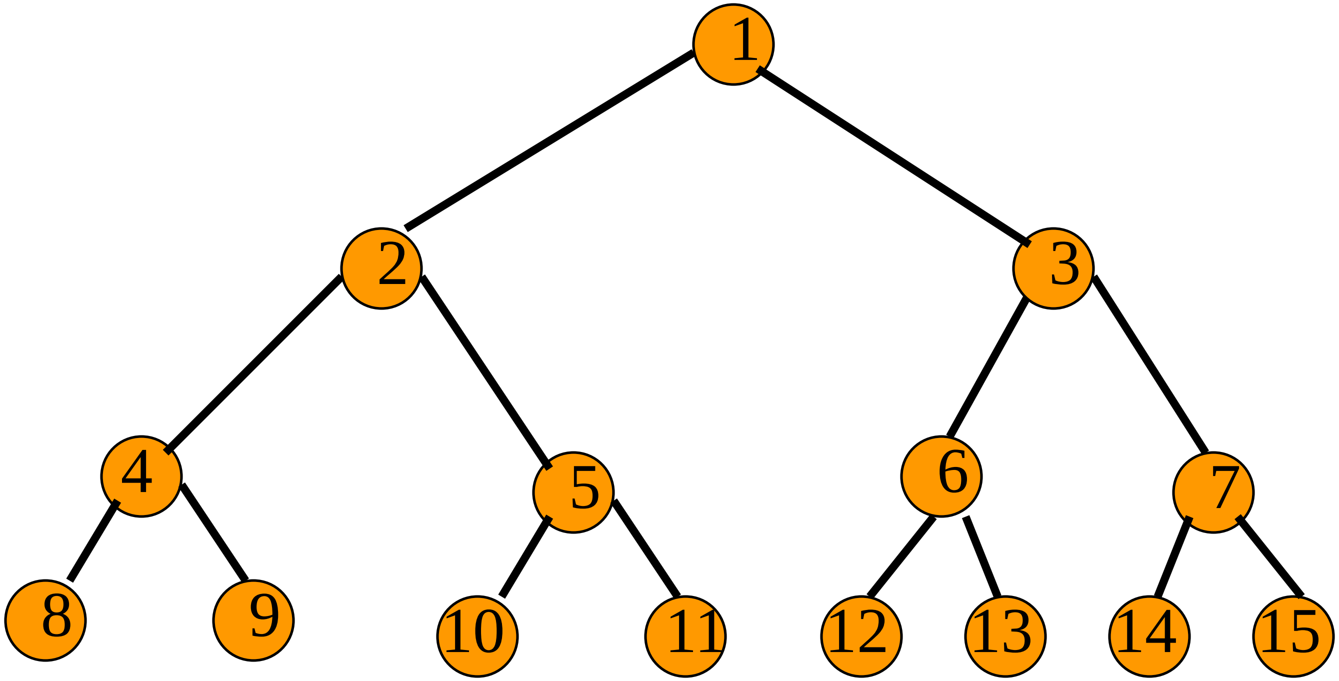
4 –н өндөртэй бүтэн хоёртын мод.

Хоёртын бүтэн модны зангилааг дугаарлах

- Зангилааны дугаар 1 - $2^h - 1$.
- Түвшин дээрээс доош дугаарлагдана
- Түвшин дотроо зүүнээс баруун тийш.

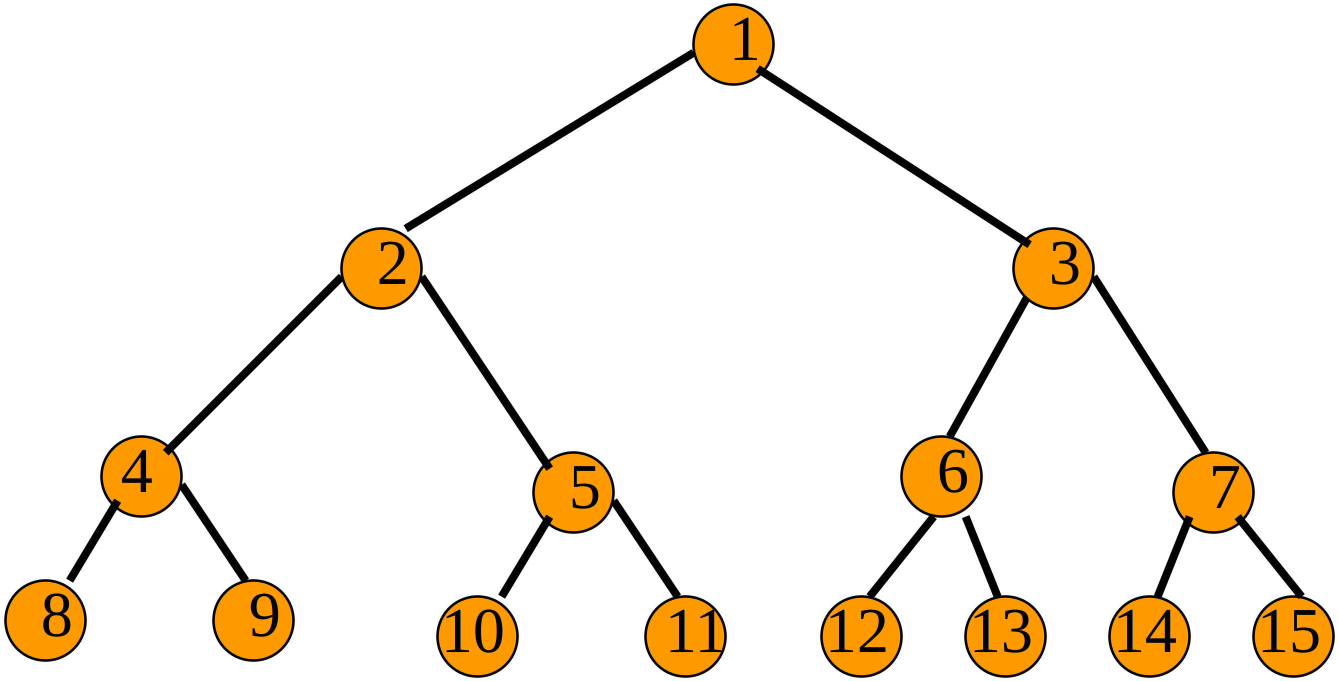


Зангилааны дугаарын шинж



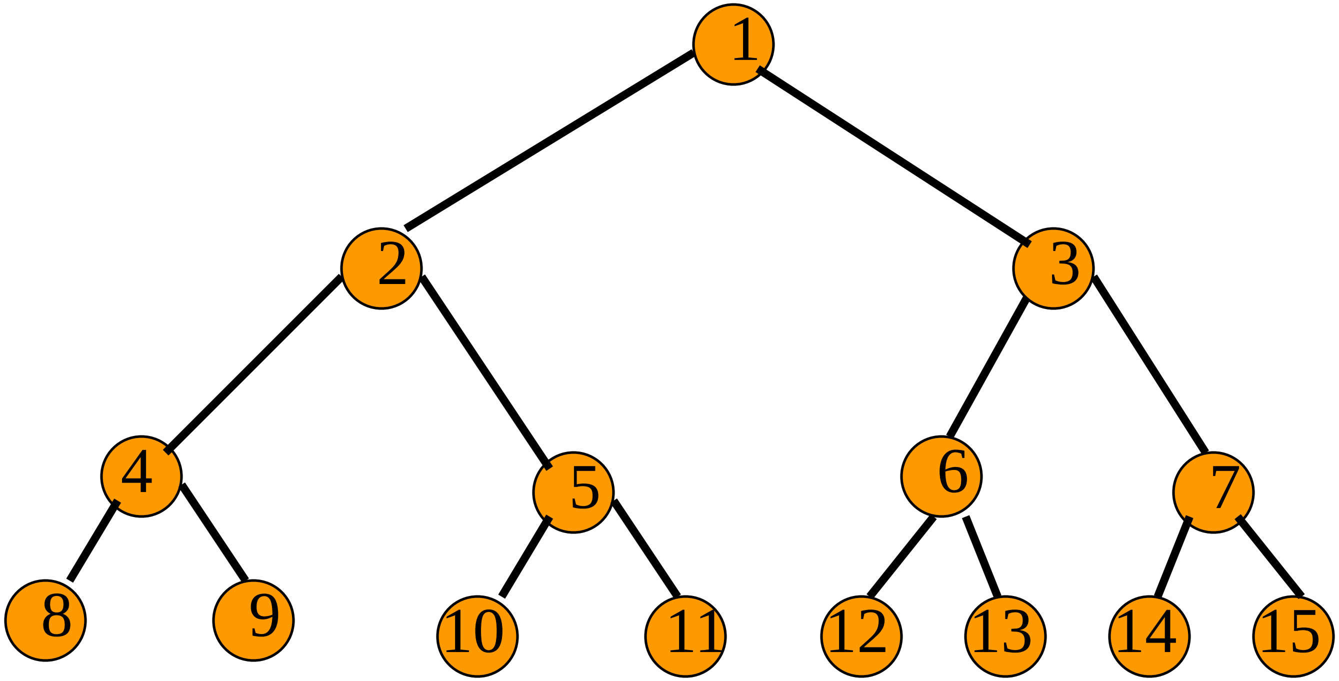
- Зангилаа i –ийн эцэг $i / 2$, ($i \neq 1$)
- Зангилаа 1 бол үндэс, эцэггүй.

Зангилааны дугаарын шинж



- $2i > n$ биш бол зангилаа i –н зүүн хүү нь $2i$, үүнд n зангилааны тоо.
- Хэрвээ $2i > n$, зангилаа i зүүн хүүгүй.

Зангилааны дугаарын шинж

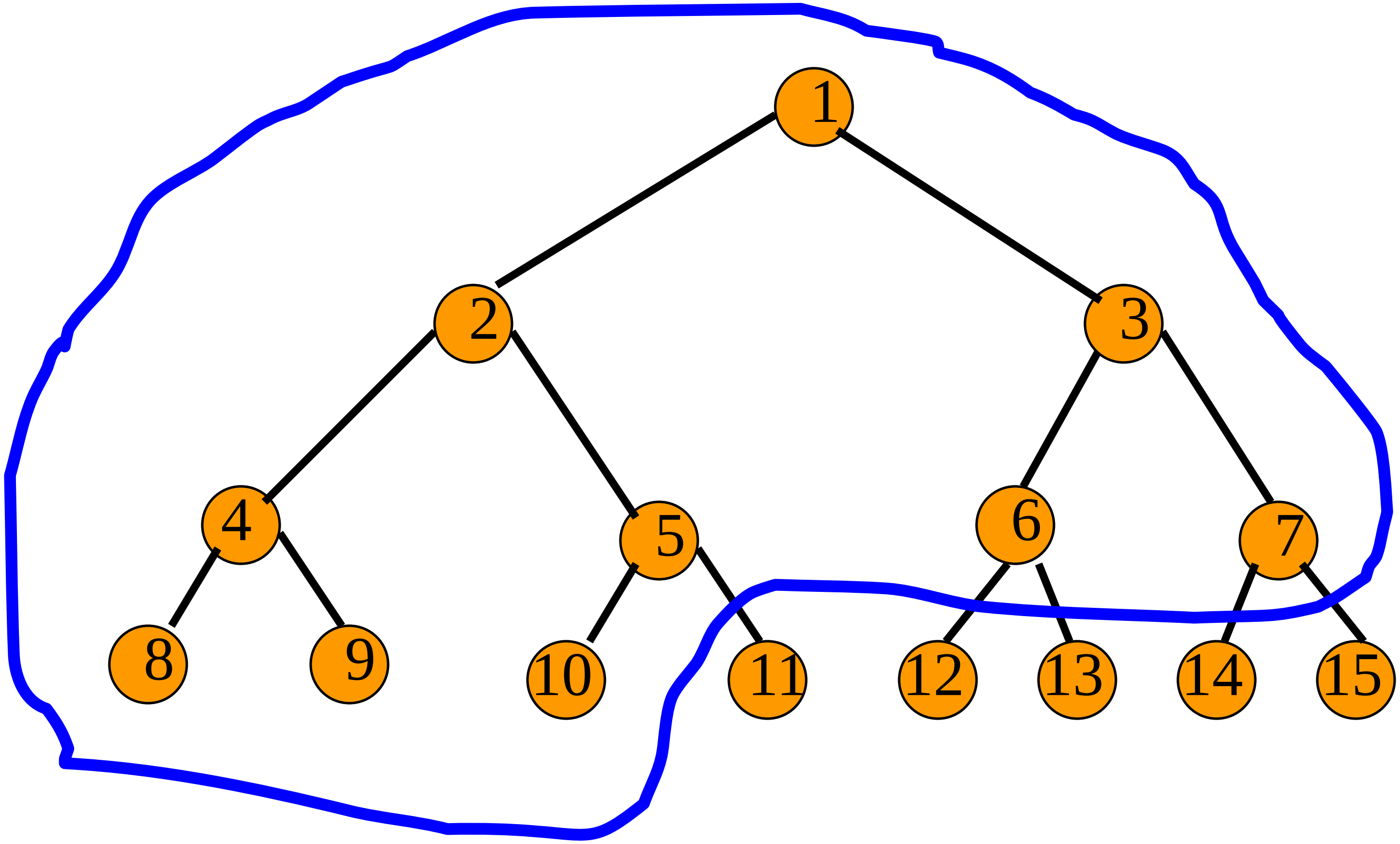


- $2i+1 > n$, биш бол зангилаа i –н баруун хүү, үүнд n зангилааны тоо.
- Хэрвээ $2i+1 > n$, зангилаа i баруун хүүгүй.

n зангилаатай төгс хоёртын мод

- Дор хаяж n зангилаатай бүтэн модноос эхэл.
- Өмнө үзсэнээр зангилааг дугаарла.
- 1 –ээс n хүртэл дугаарлагдсан зангилаатай хоёртын модыг орь ганц n зангилаатай төгс хоёртын мод гэнэ.

Жишээ



- 10 зангилаатай төгс хоёртын мод.

Хоёртын модыг дүрслэх

- Массив дүрслэл
- Холбоост дүрслэл