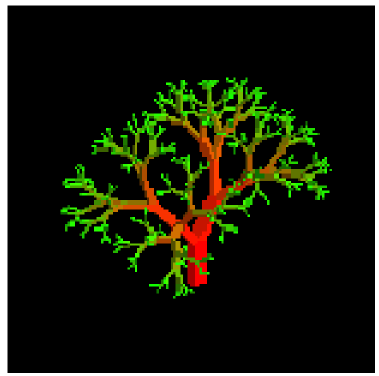
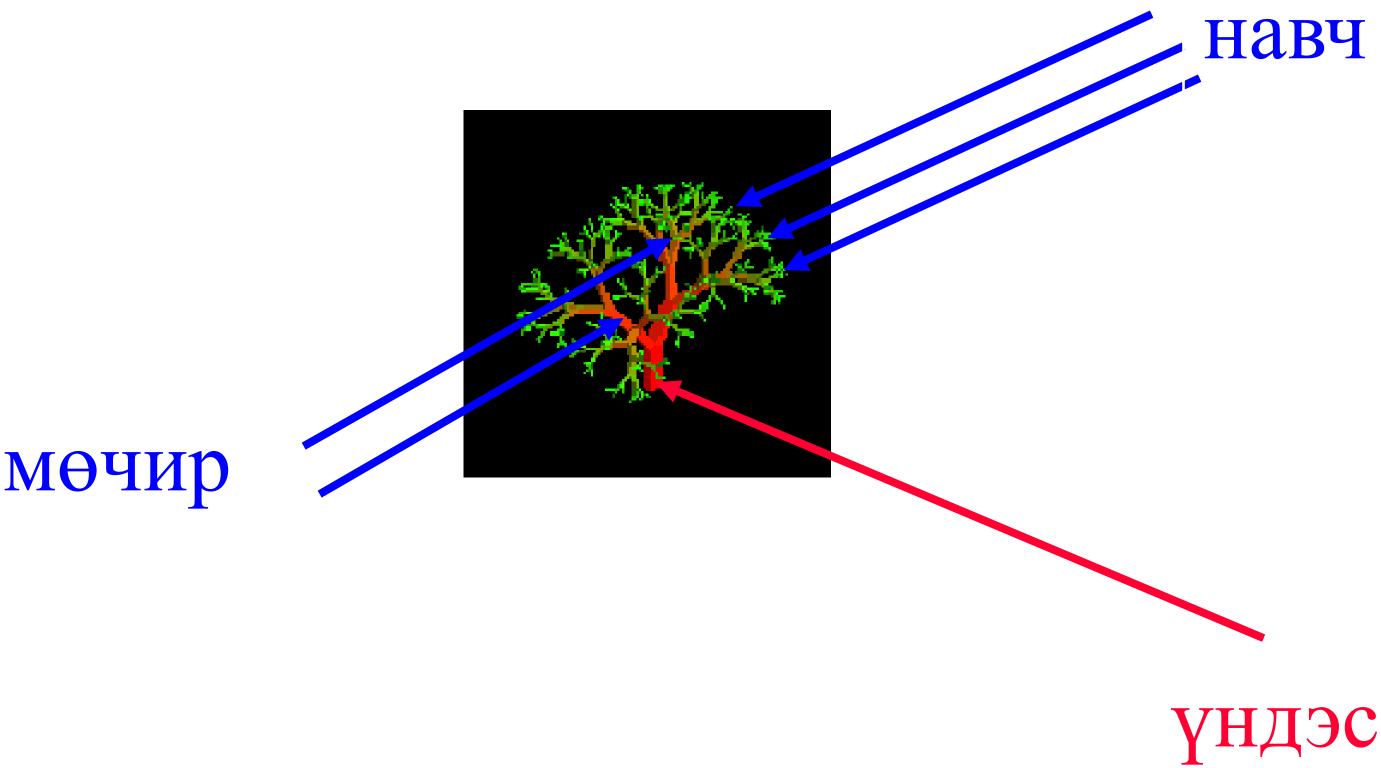
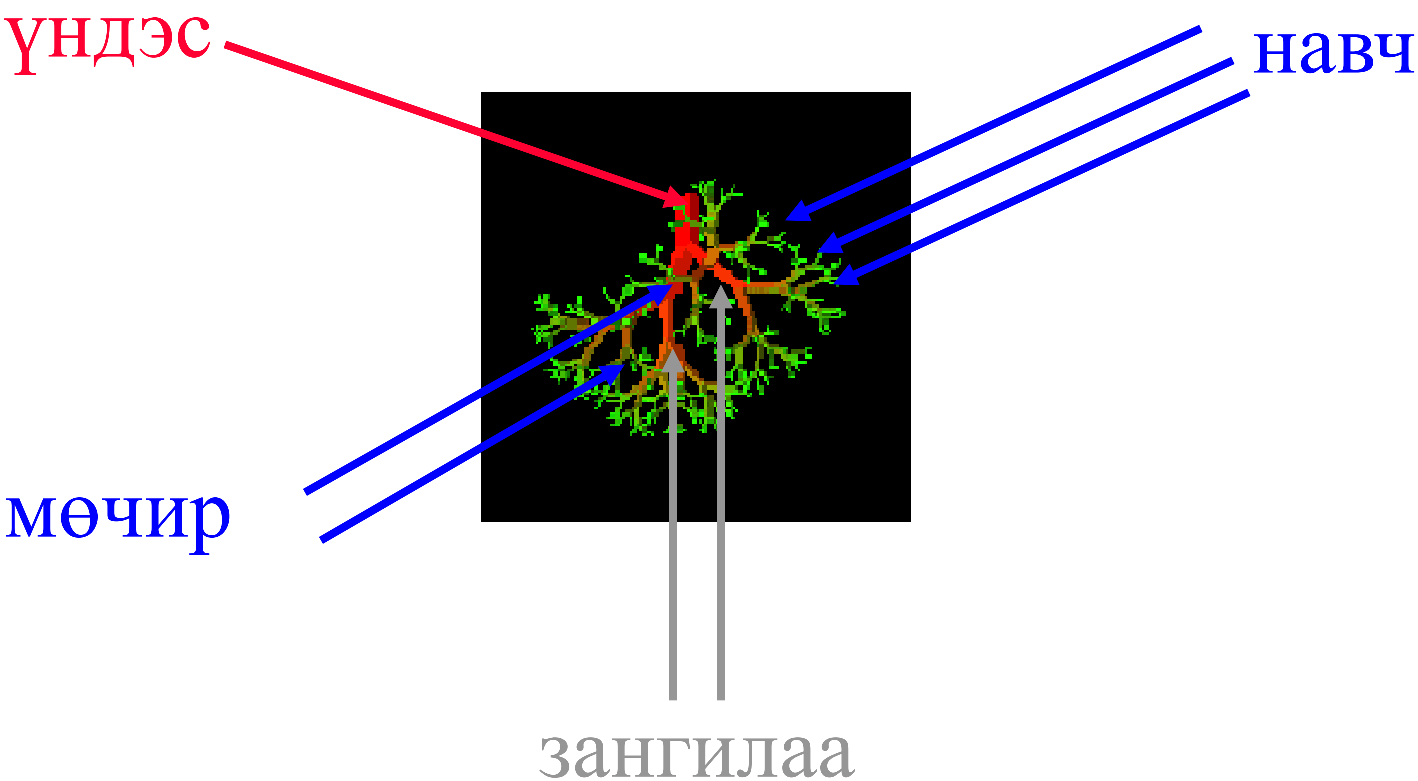
Мод



Байгальд хайртай хүний нүдээр



Компьютерийн хүний нүдээр



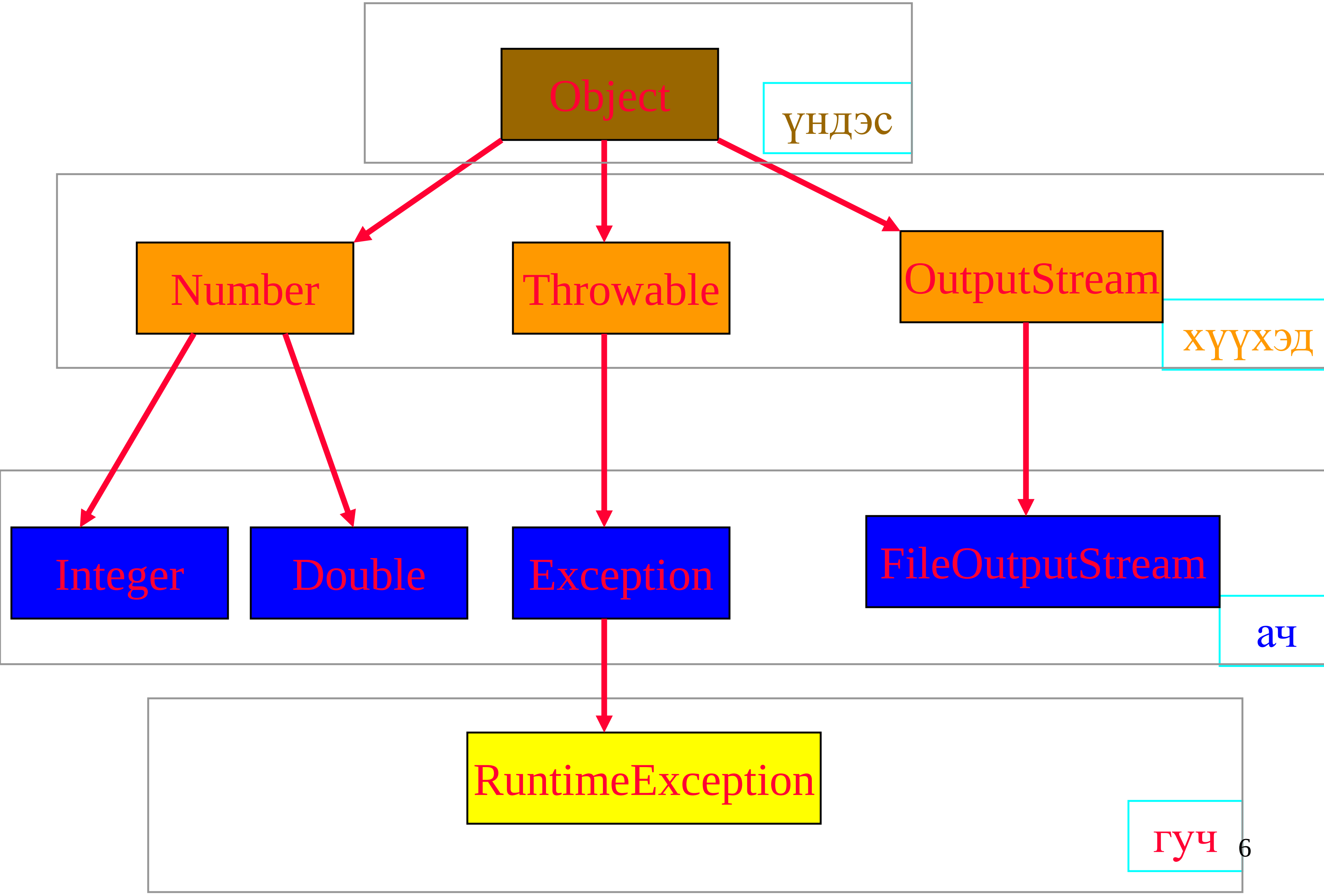
Шугаман жагсаалт ба Мод

- Шугаман жагсаалт цуварч эрэмбэлэгдсэн өгөгдөлд тохиромжтой.
  - $(e_0, e_1, e_2, \ldots, e_{n-1})$
  - 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.

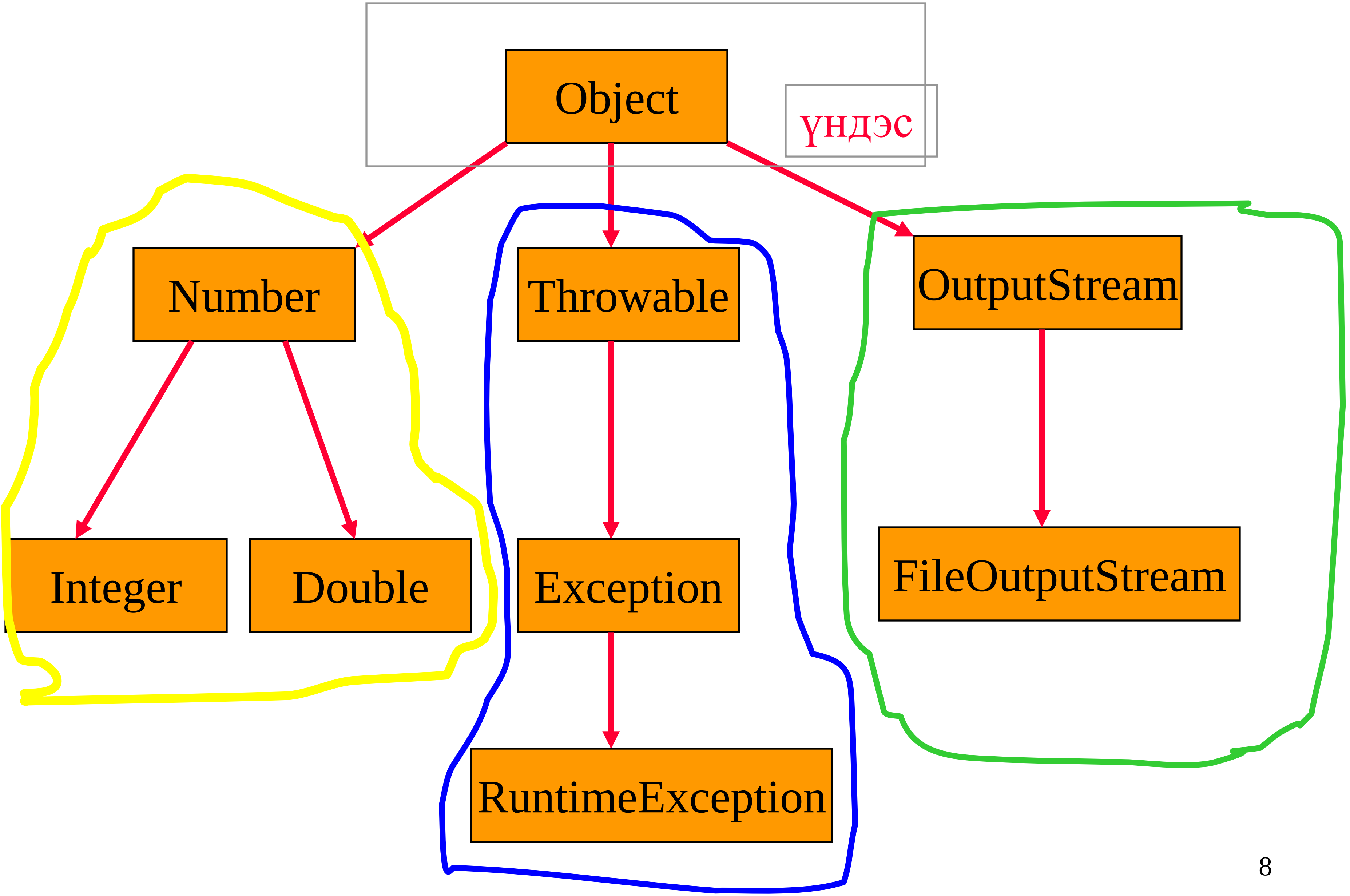
Java-ийн классууд



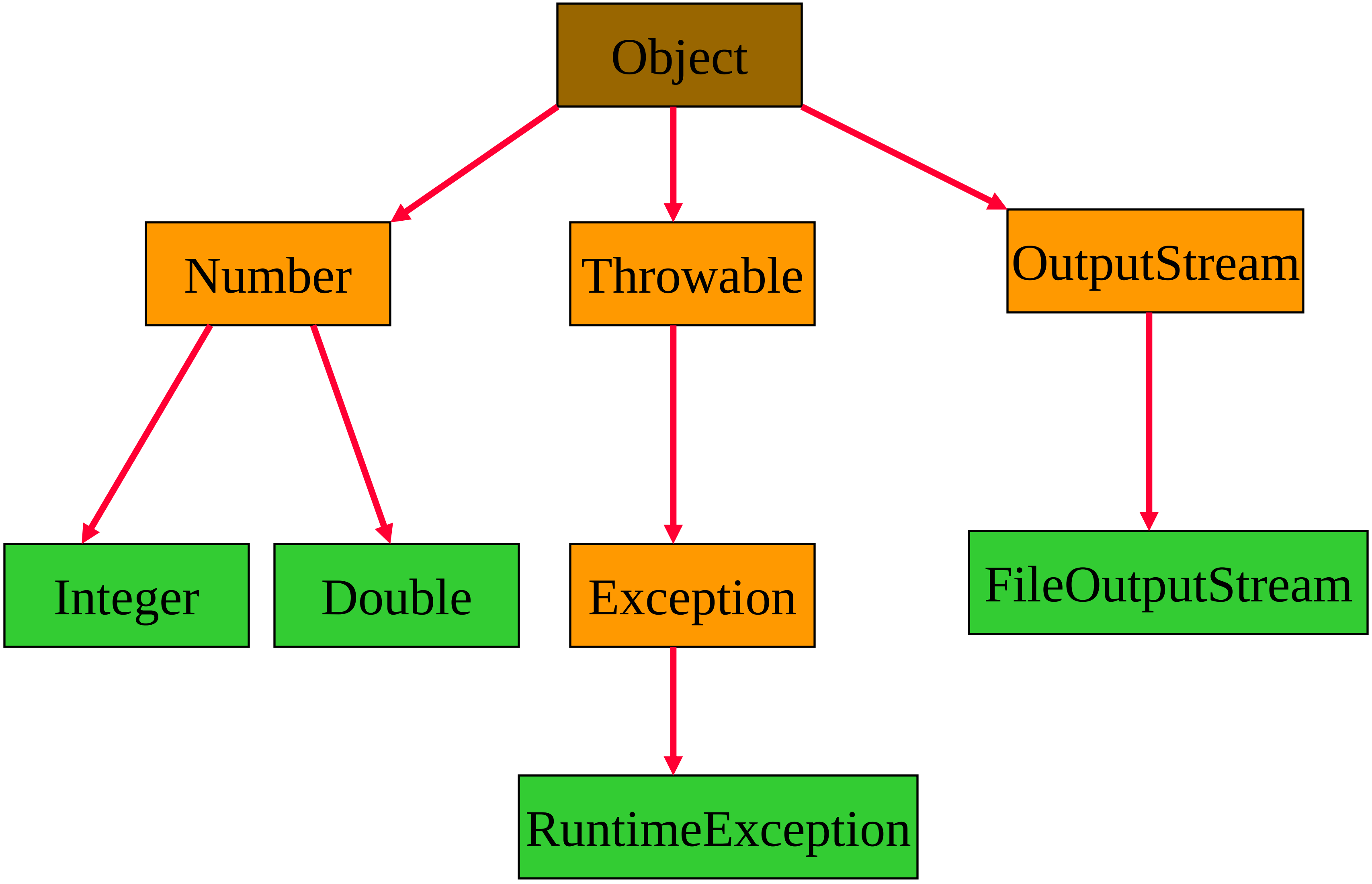
Тодорхойлолт

- 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

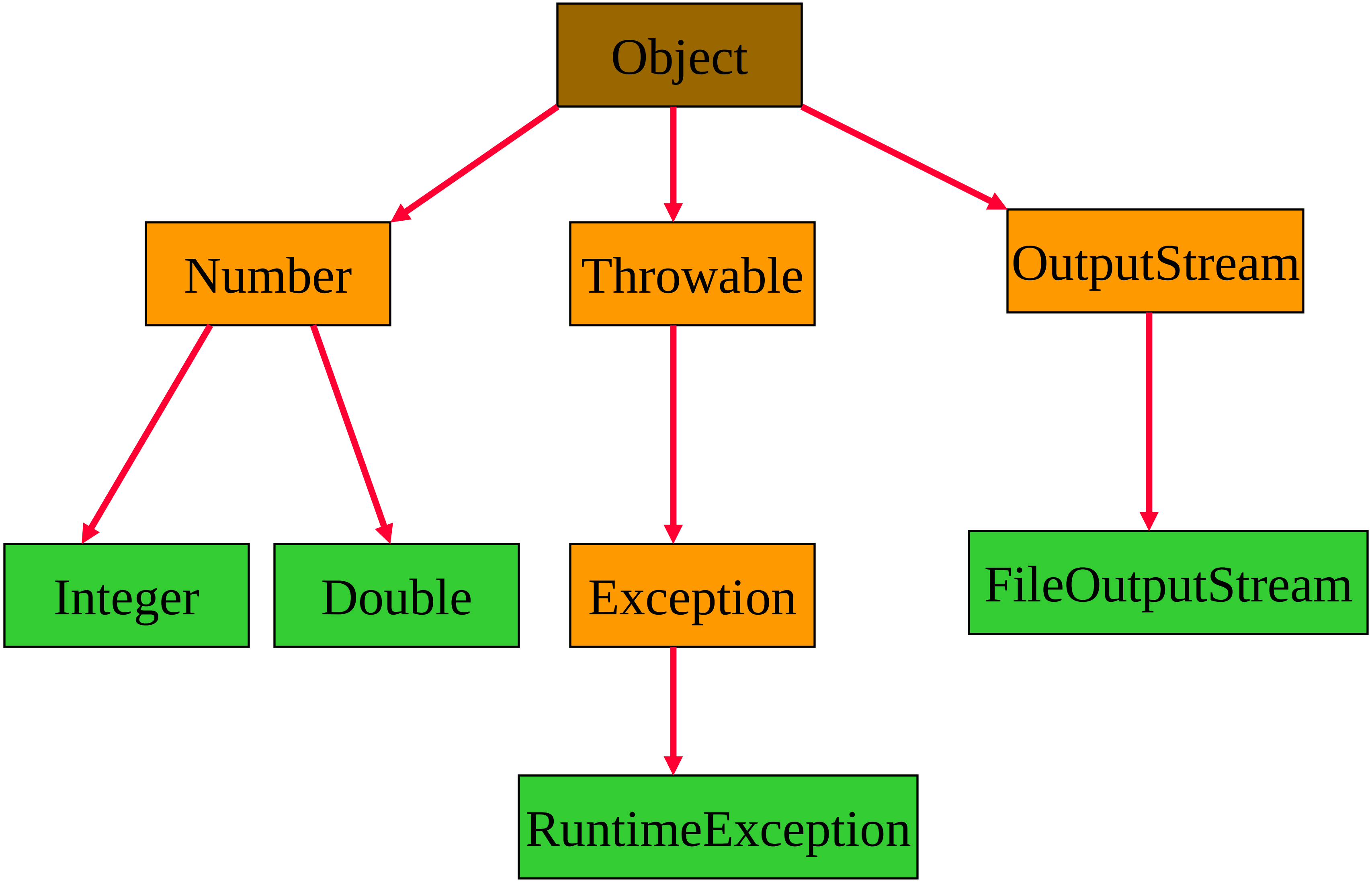
Дэд мод


Object
үндэс
Number
Throwable
OutputStream
Integer
Double
Exception
FileOutputStream
RuntimeException

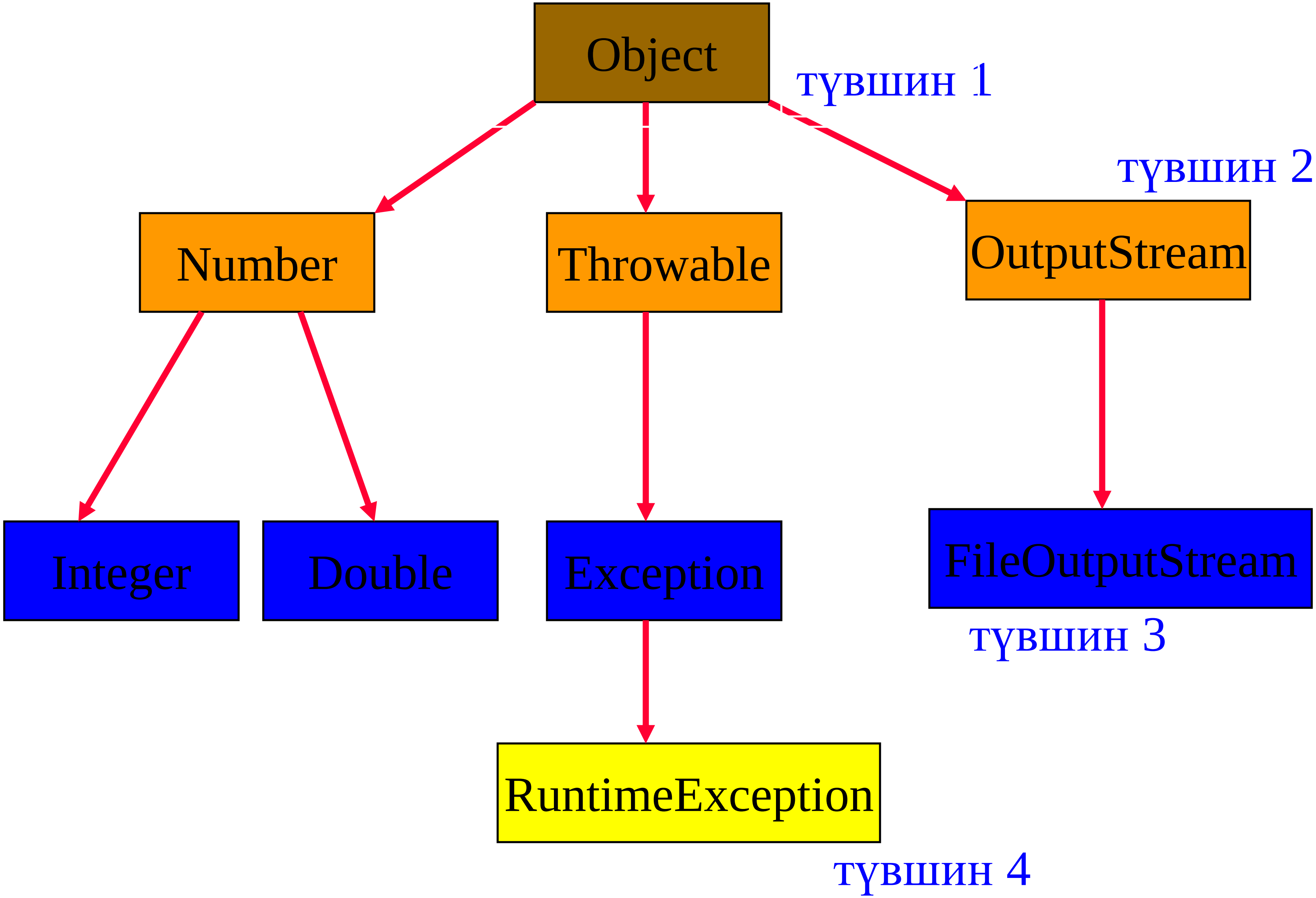
Навчис


Object
Number
Throwable
OutputStream
Integer
Double
Exception
FileOutputStream
RuntimeException

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



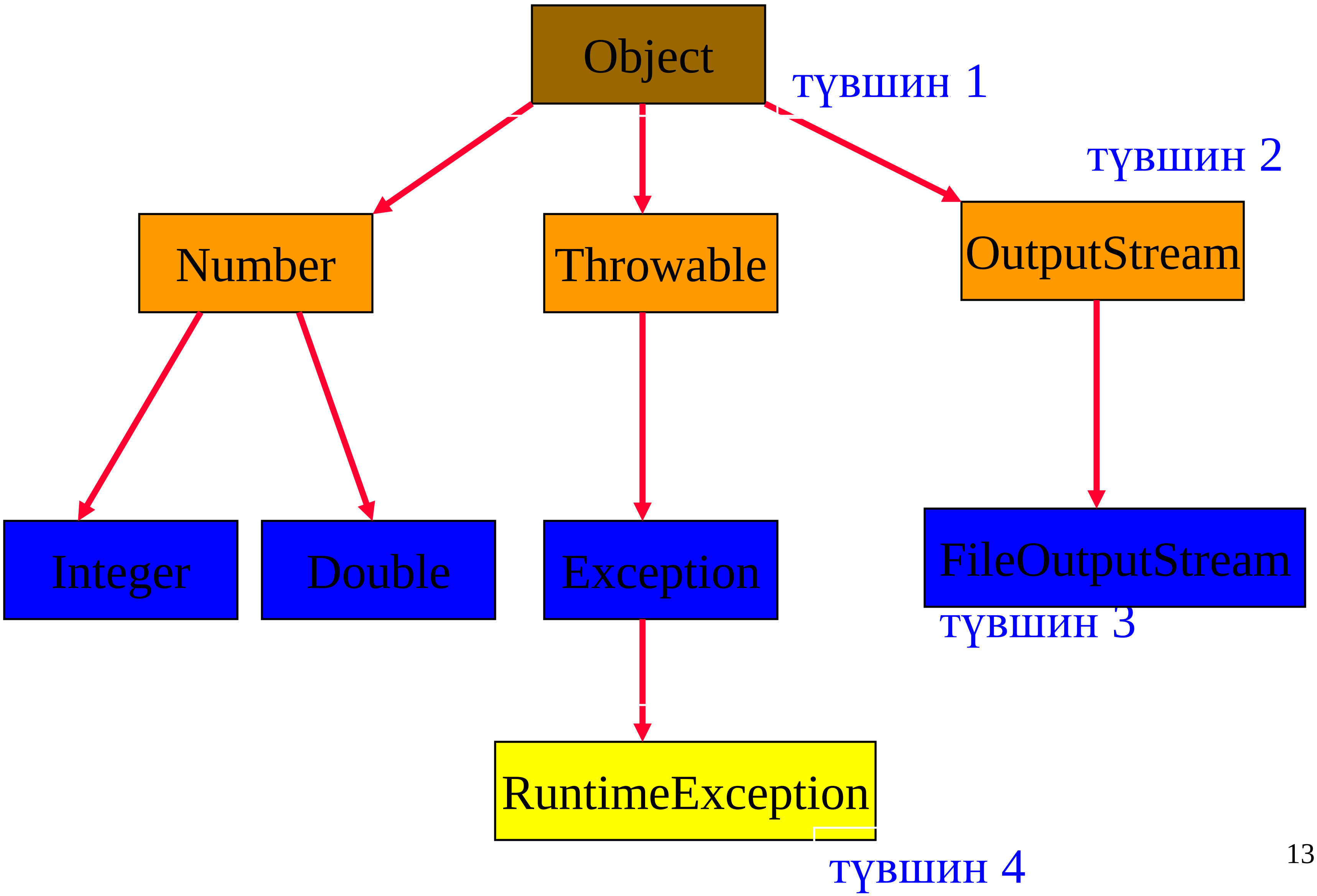
Түвшин-үе


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

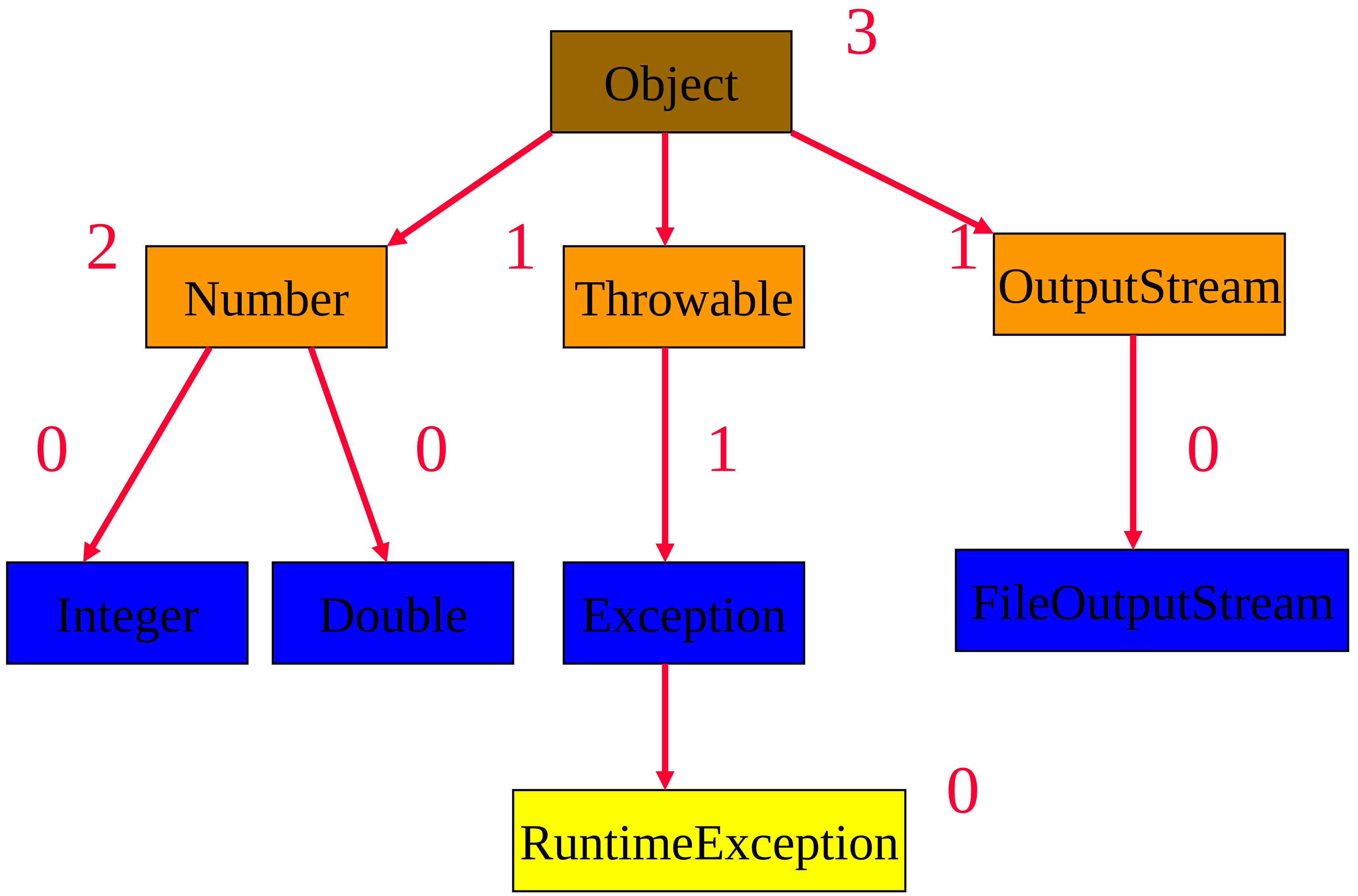
Анхааруулга

- 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

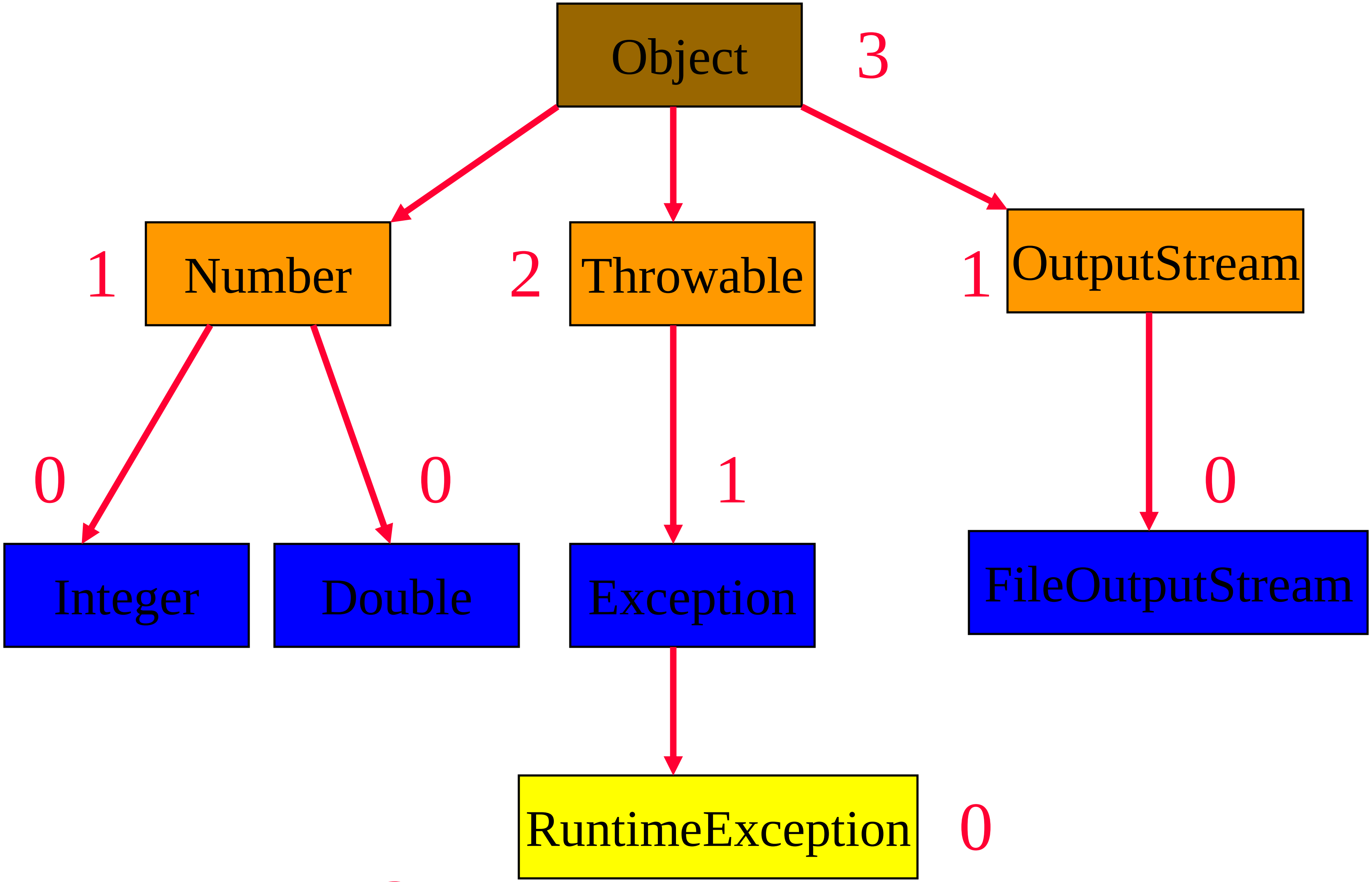
height = depth = түвшний тоо



Зангилааны зэрэг = Хүүхдийн тоо



Модны зэрэг = Max(Зангилааны зэрэг)



Модны зэрэг = 3.

Хоёртын мод

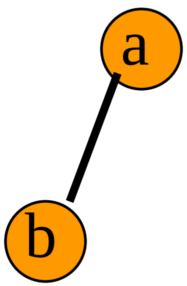
- Элементүүдийн төгсгөлтэй (хоосон байж болно) цуглуулга.
- Хоосон бус хоёртын мод root-үндэс элементтэй.
- Бусад элементүүд (байгаа бол) хоёр хоёртын модонд хуваагдана.
- Тэднийг хоёртын модны left-зүүн ба right-баруун дэд моднууд гэнэ.

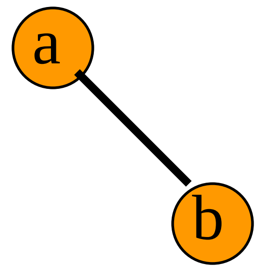
Мод, Хоёртын модны ялгаа

- 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), etc.).
  - 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 хооронд 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(-)
- Гишүүн 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 компьютерын 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

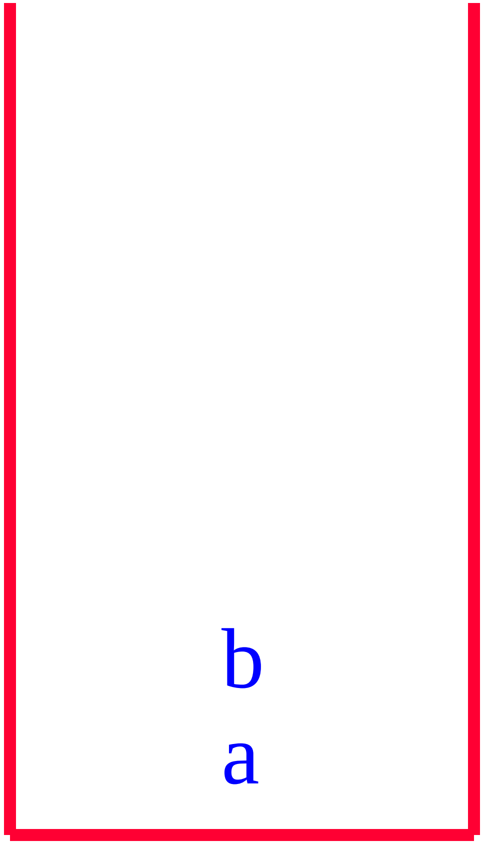
- Шинэ тэмдгээр орлуулна.
  - + a => a @
  - + a + b => a @ b +
  - - a => a ?
  - - a-b => a ? b -

Postfix бодолт

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

Postfix бодолт

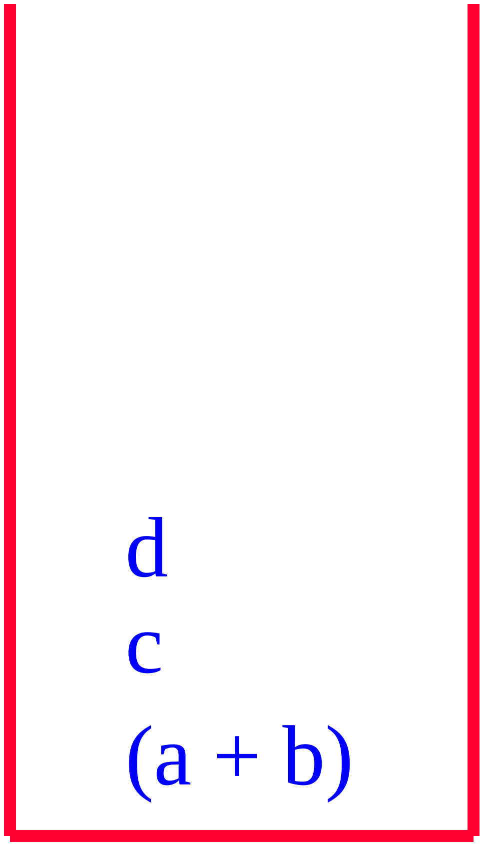
- (a + b) * (c – d) / (e + f)
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /
- a b + c d - * e f + /



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 + /



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)

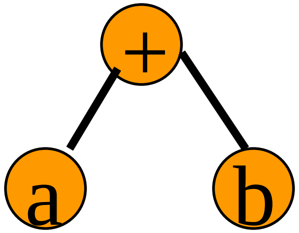
стек

Prefix хэлбэр

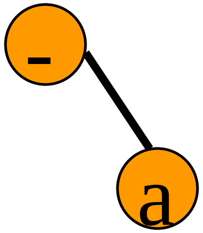
- Хувьсагч, тогтмолууд адилхан бичигдэнэ.
  - a, b, 3.25
- Үйлдлийн гишүүдийн дараалал infix, prefix хэлбэрүүдэд адилхан
- Үйлдлүүд postfix хэлбэрийн гишүүдийнхээ өмнө шууд бичигддэг.
  - Infix = a + b
  - Postfix = ab+
  - Prefix = +ab

Хоёртын модны хэлбэр

- a + b

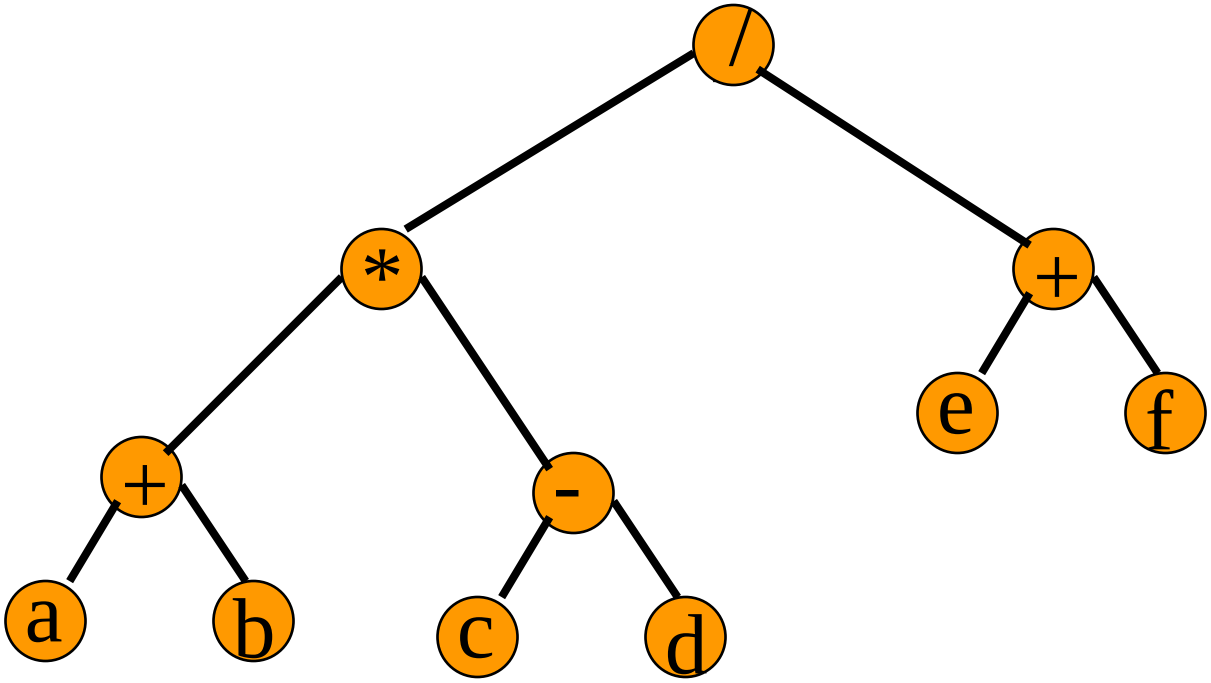


- **-** a



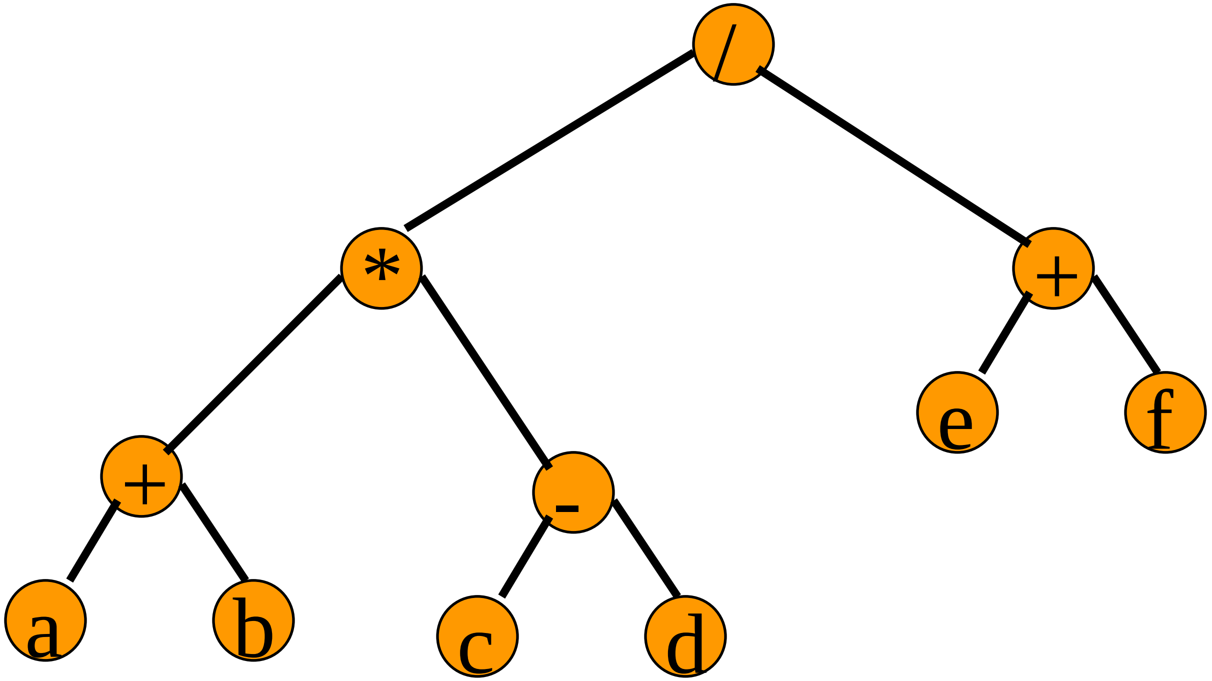
Хоёртын модны хэлбэр

- (a + b) * (c – d) / (e + f)

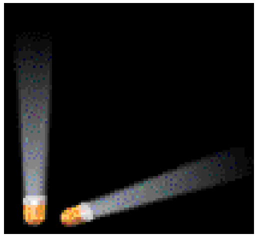


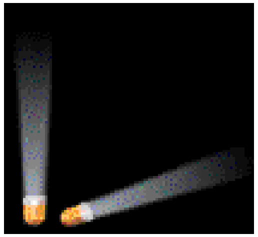
Хоёртын модны хэлбэрийн давуу тал

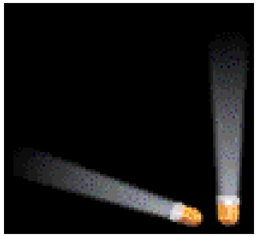
- Үйлдлийн зүүн, баруун гишүүд ил харагддаг.
- Илэрхийллийн хоёртын модны хэлбэр дээр кодыг оновчлох алгоритм сайн ажилладаг.
- Илэрхийллийг рекурсив аргаар бодоход хялбар.

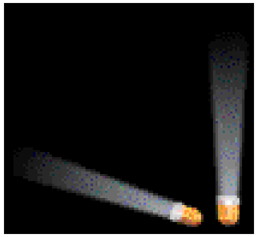


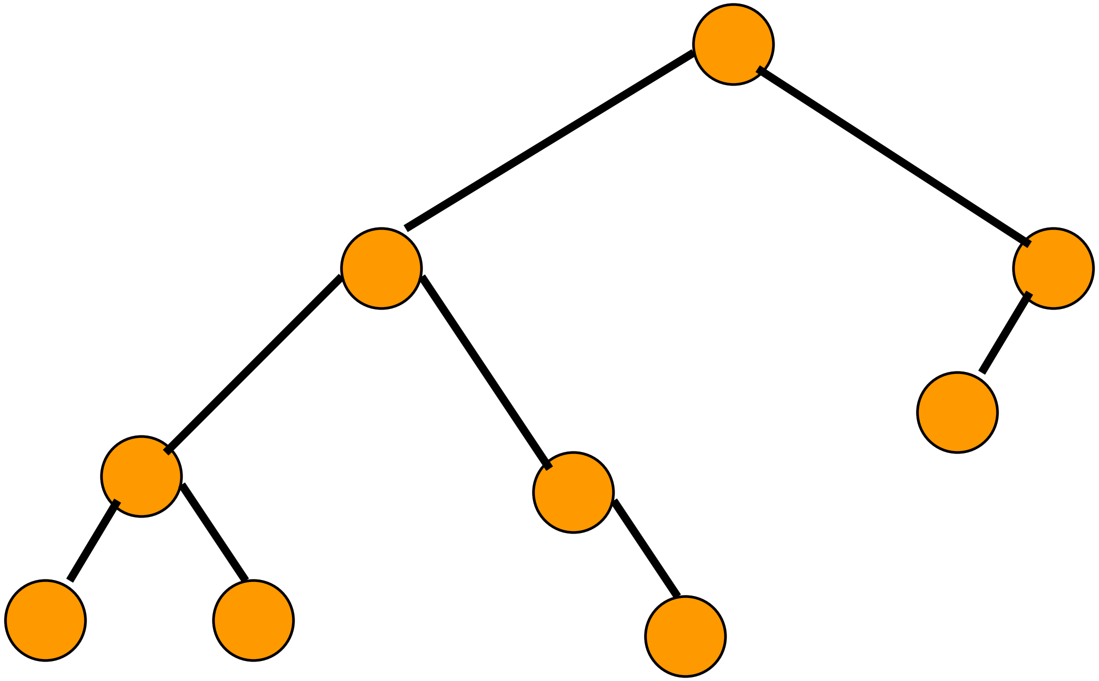
Хоёртын модны шинж ба дүрслэл





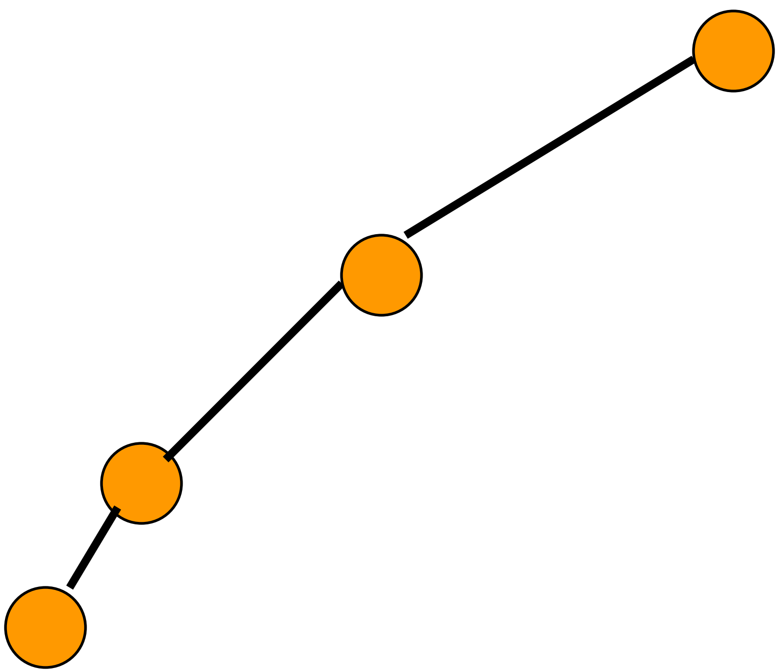






Минимум зангилааны тоо

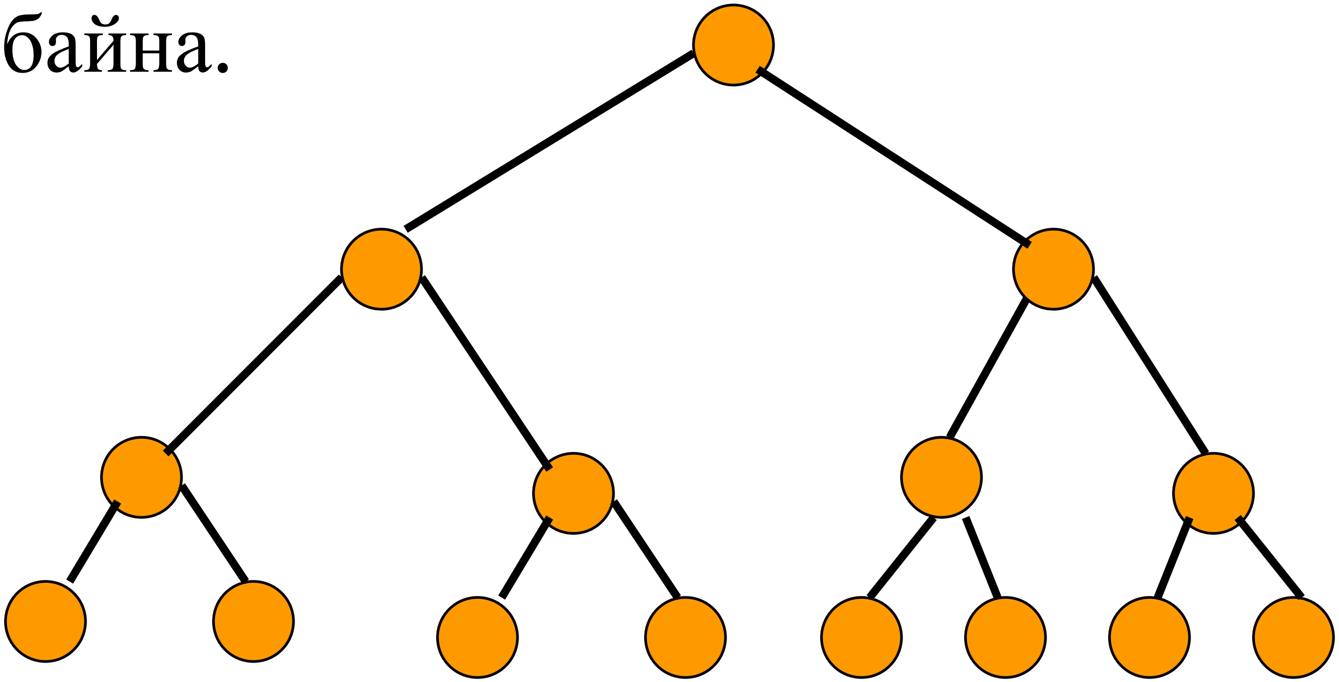
- h өндөртэй хоёртын модны минимум зангилааны тоо
- Эхний h түвшин тус бүрд ядаж нэг зангилаа байна



Минимум зангилааны тоо h

Максимум зангилааны тоо

- Эхний h түвшин бүрт боломжит бүх зангилаа байна.



Максимум зангилааны тоо

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

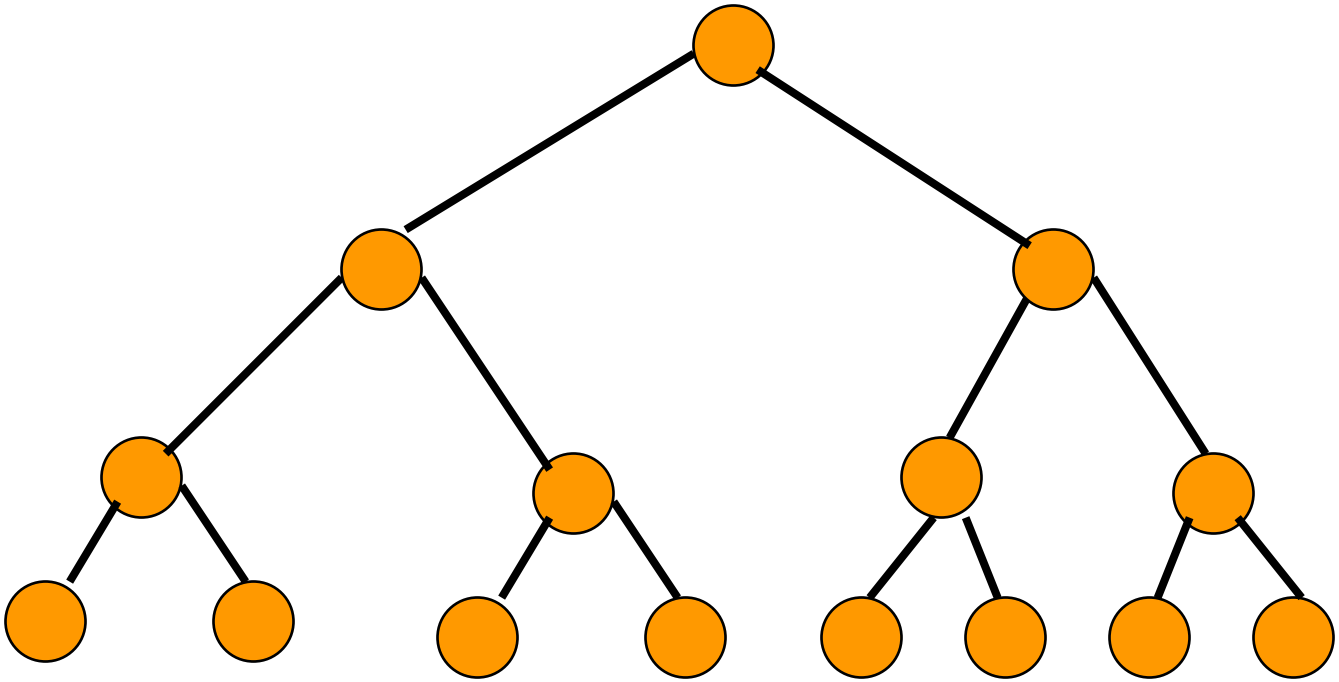
$= 2^h - 1$

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

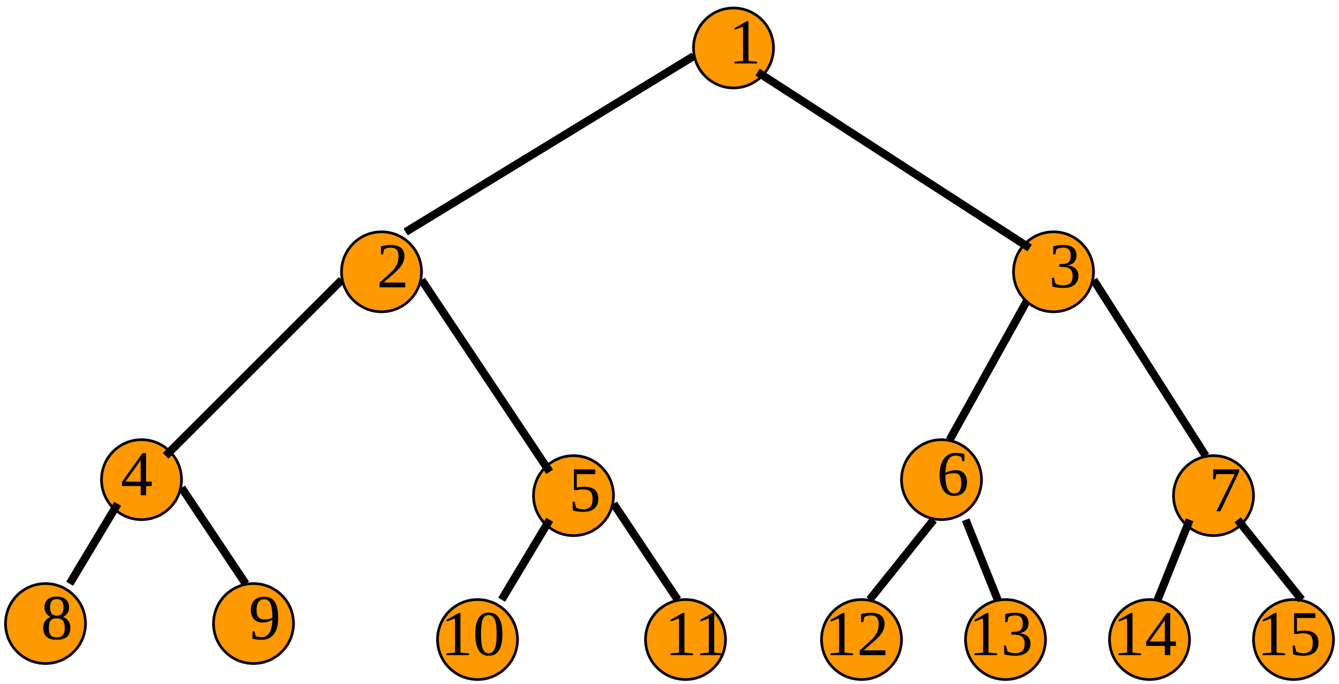
- h өндөртэй хоёртын бүтэн модонд $2^h - 1$ зангилаа байна.



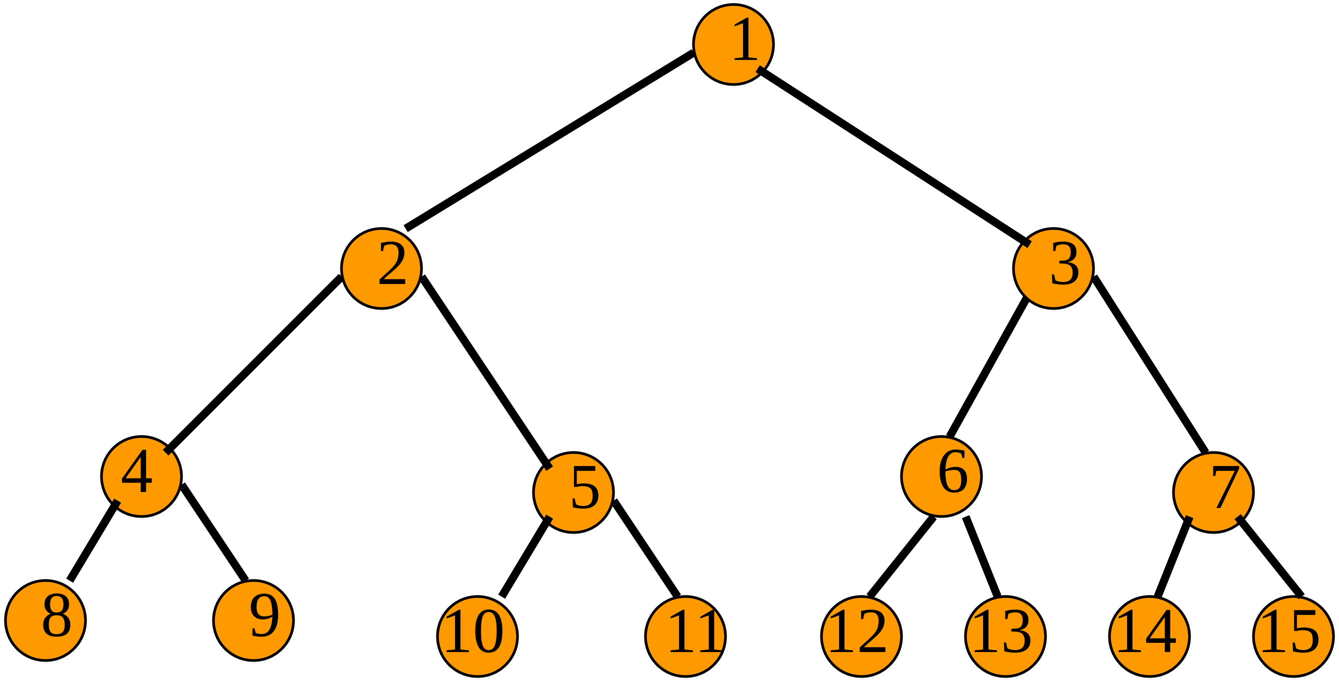
4 –н өндөртэй бүтэн хоёртын мод.

Хоёртын бүтэн модны зангилааг дугаарлах

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

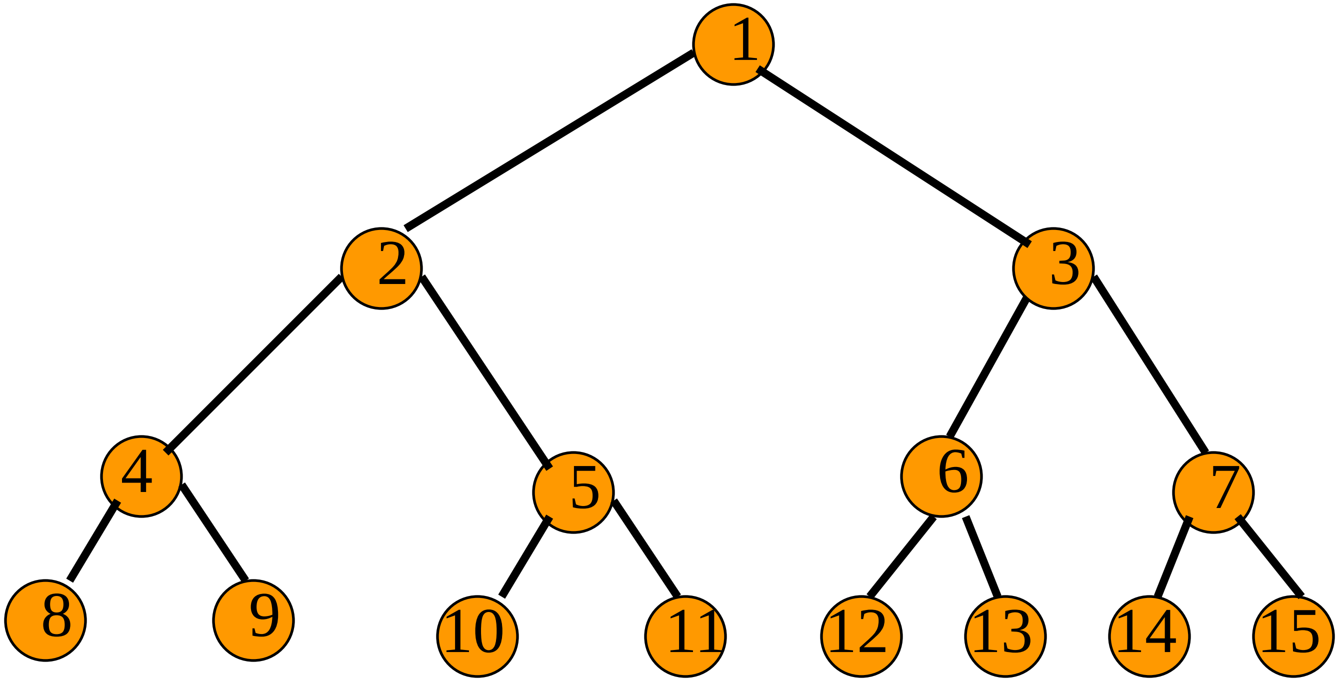


Зангилааны дугаарын шинж



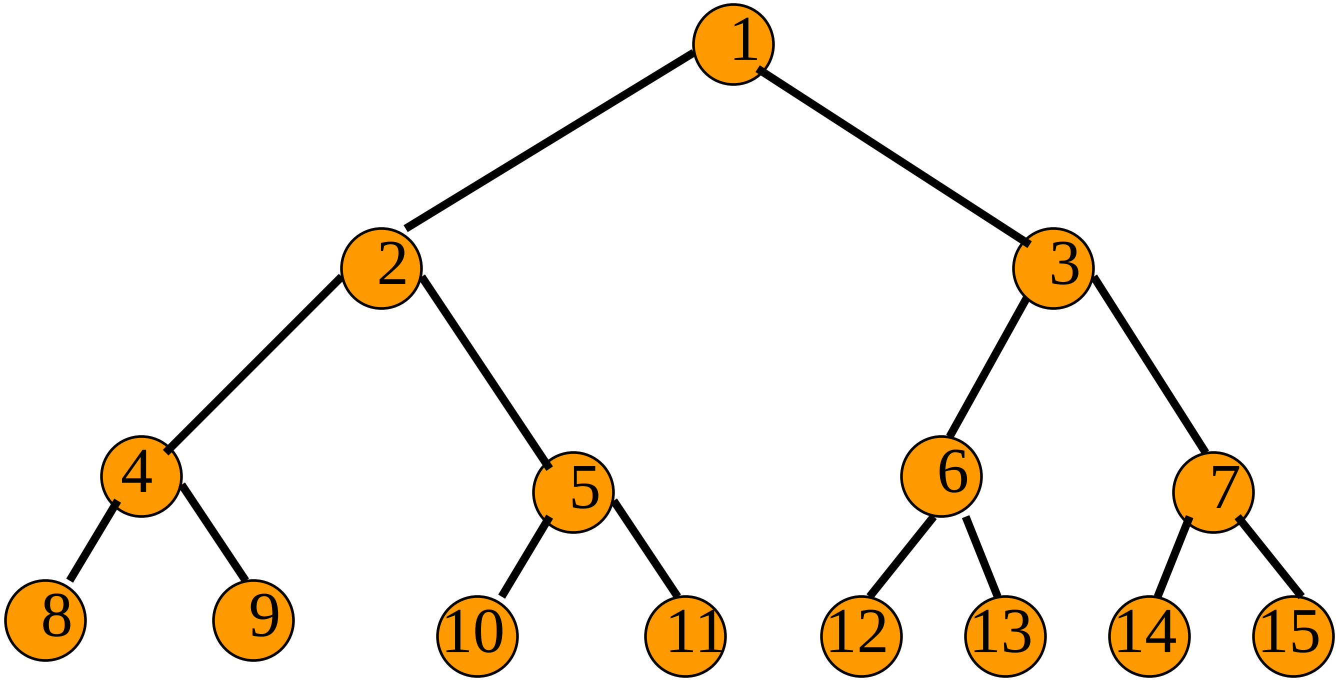
- Зангилаа i –ийн эцэг i / 2, (i ≠ 1)
- Зангилаа 1 бол үндэс, эцэггүй.

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

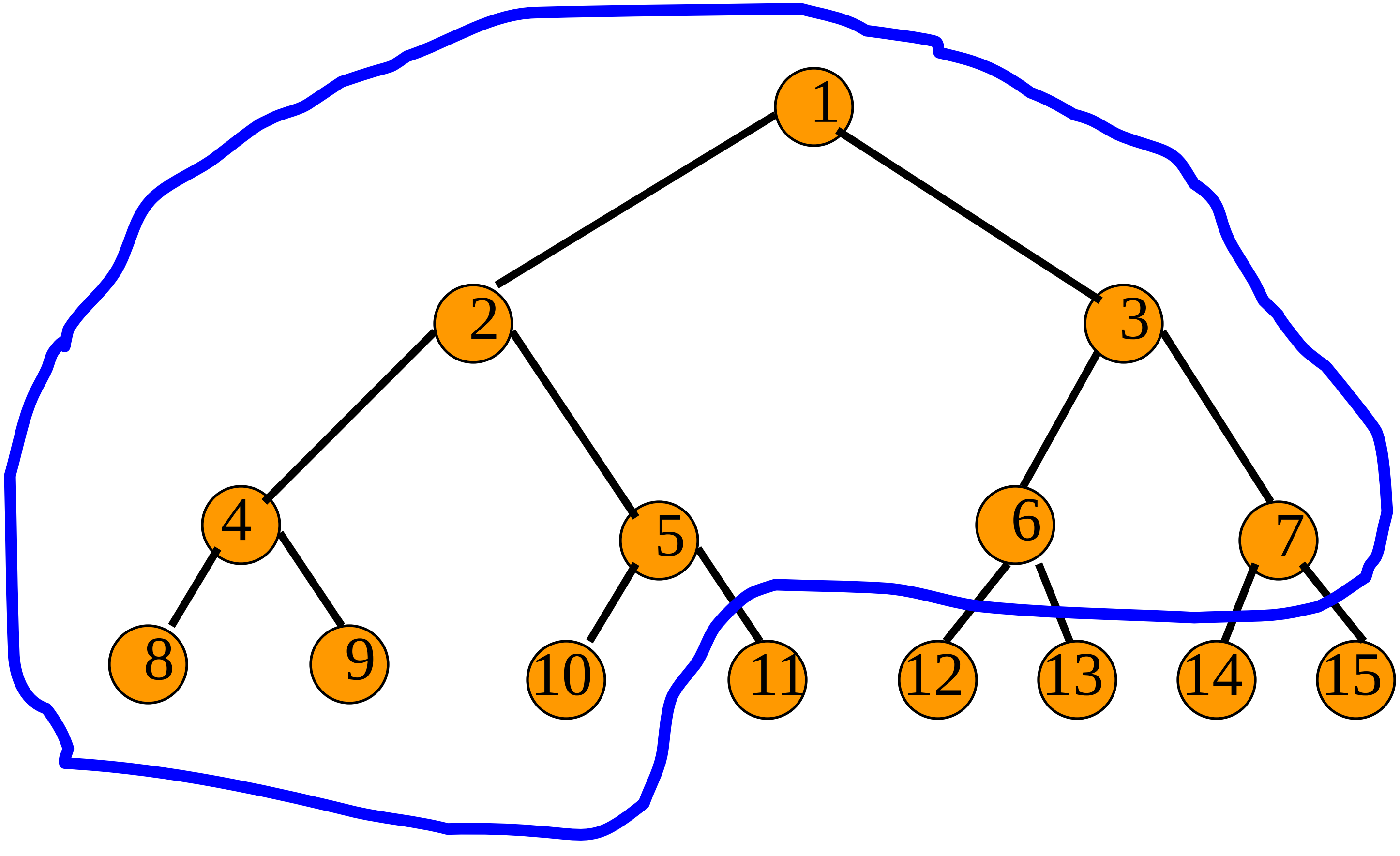


- $2i+1 > n$, биш бол зангилаа i –н баруун хүү, үүнд n зангилааны тоо.
- Хэрвээ $2i+1 > n$, зангилаа i баруун хүүгүй.

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



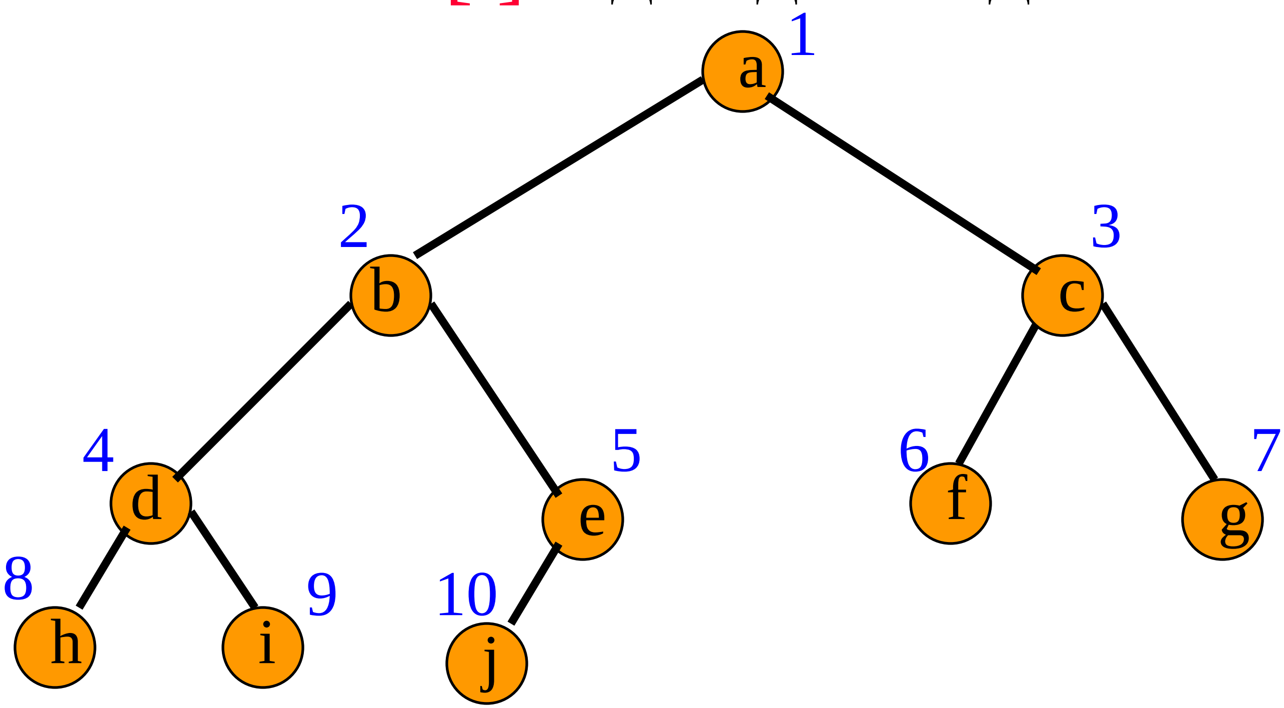
- 10 зангилаатай төгс хоёртын мод.

Хоёртын модыг дүрслэх

- Массив дүрслэл.
- Холбоост дүрслэл.

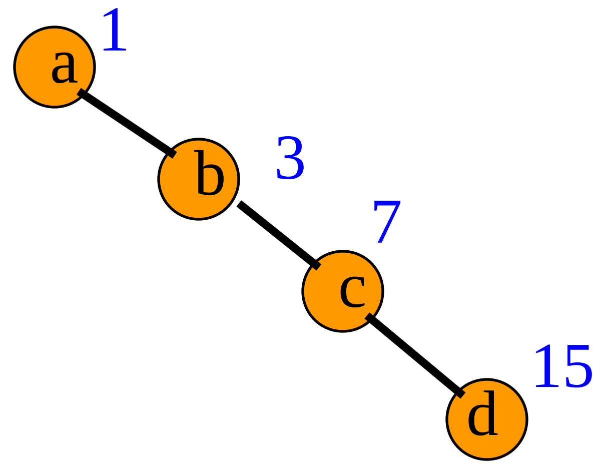
Массив дүрслэл

- Бүтэн хоёртын модыг дугаарлах схемээр зангилаануудыг дугаарла. i дугаартай зангилаа tree[i] –д хадгалагдана



tree[]		a	b	c	d	e	f	g	h	i	j
	0					5					10

Баруун-хазайлттай хоёртын мод



tree[]

	a	-	b	-	-	-	c	-	-	-	-	-	-	-	d
0					5					10					15

- n зангилаатай хоёртын модонд шаардлагатай массивын урт $n+1$ ба 2^n хооронд байна

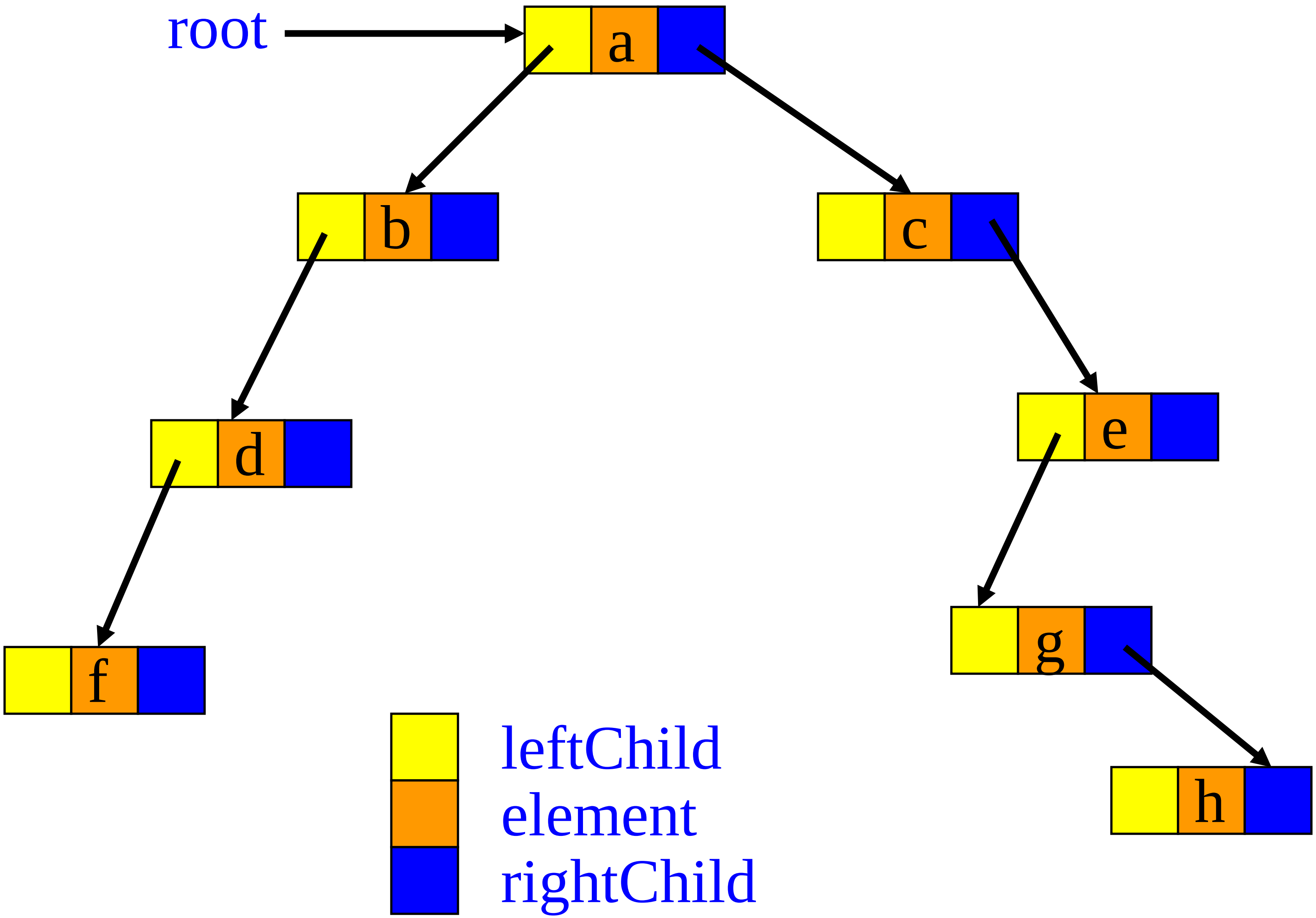
Холбоост дүрслэл

- Хоёртын модны зангилаа бүр BinaryTreeNode гэсэн өгөгдлийн төрлийн объект байна.
- n зангилаатай хоёртын модонд шаардлагатай орон зай n * (нэг зангилааны орон зай).

BinaryTreeNode класс

```
package dataStructures;
public class BinaryTreeNode
{
   Object element;
   BinaryTreeNode leftChild; // зүүн дэд мод
   BinaryTreeNode rightChild;// баруун дэд мод
   // байгуулагч болон бусад аргууд
   // энд бичигдэнэ
}
```

Холбоост дүрслэлийн жишээ



Хоёртын модны зарим үйлдлүүд

- Өндрийг олох.
- Зангилааны тоог олох.
- Хувилах.
- Хоёр хоёртын мод хувилагдсан эсэхийг тогтоох.
- Хоёртын модыг харуулах.
- Хоёртын модоор дүрслэгдсэн арифметик илэрхийллийг бодох.
- Илэрхийллийн infix хэлбэрийг гаргах.
- Илэрхийллийн prefix хэлбэрийг гаргах.
- Илэрхийллийн postfix хэлбэрийг гаргах.

Хоёртын модоор нэвтрэх

- Ихэнх хоёртын модны үйлдлүүд нь хоёртын модоор нэвтрэх замаар хийгддэг.
- Аялахдаа хоёртын модны элемент бүрээр зөвхөн нэг удаа зочилдог.
- Элементээр зочлохдоо тэр элементтэй холбоотой үйлдэл (хувилах, харуулах, үйлдлийг бодох, гэх мэт.) хийгддэг.

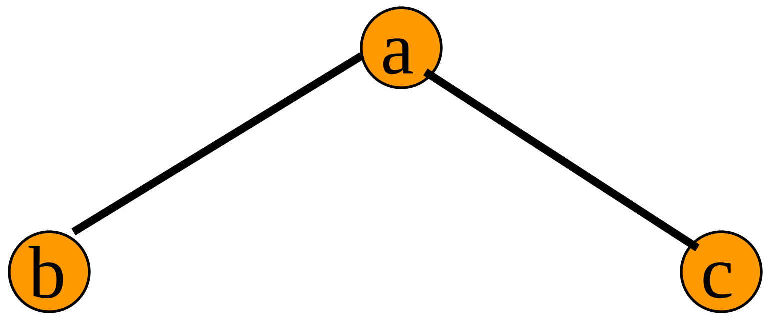
Хоёртын модоор нэвтрэх аргууд

- Preorder
- Inorder
- Postorder
- Level order

Preorder нэвтрэлт

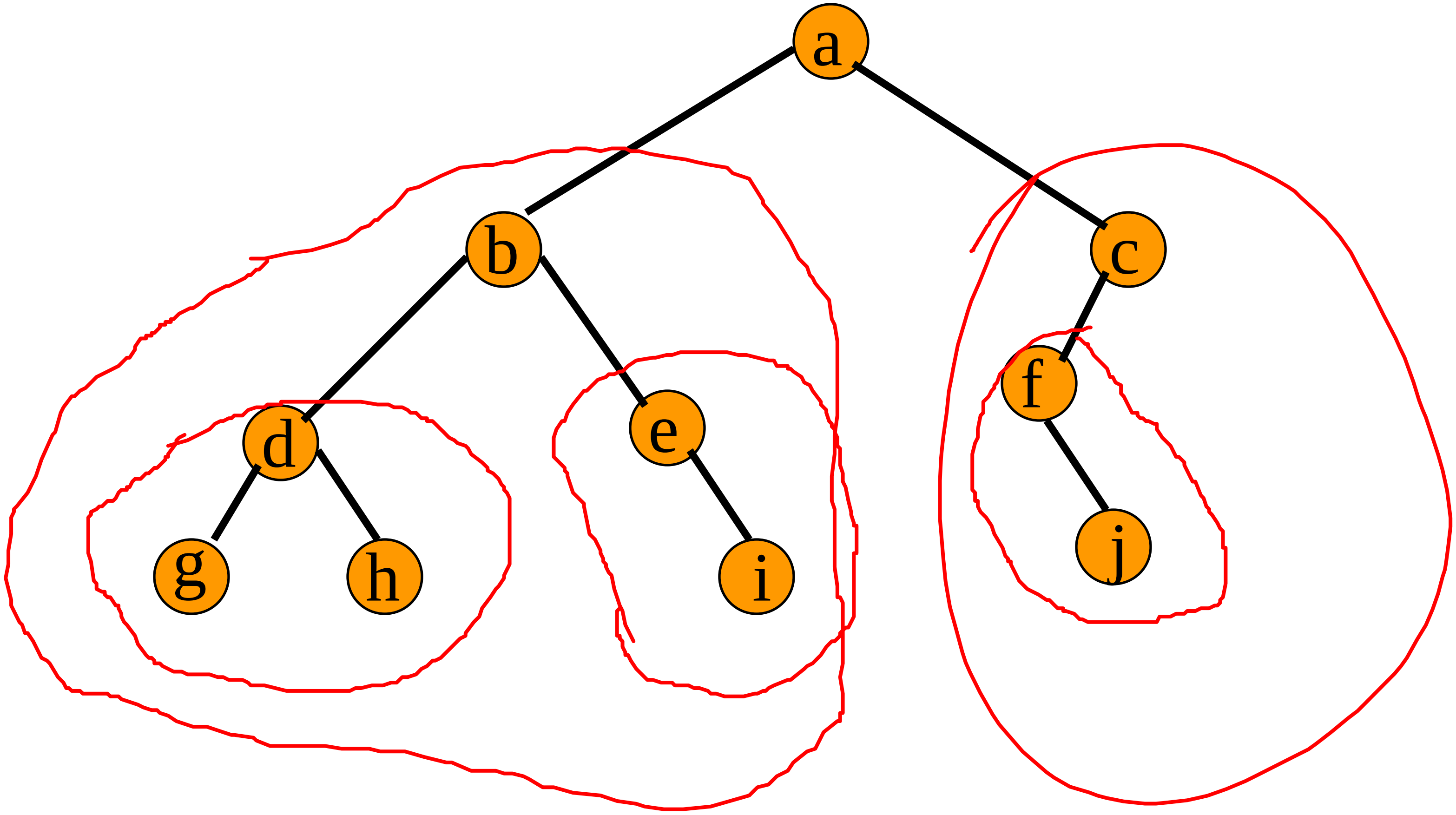
```
public static void preOrder(BinaryTreeNode t)
{
    if (t != null)
    {
        visit(t);
        preOrder(t.leftChild);
        preOrder(t.rightChild);
    }
}
```

Preorder жишээ (visit = print)



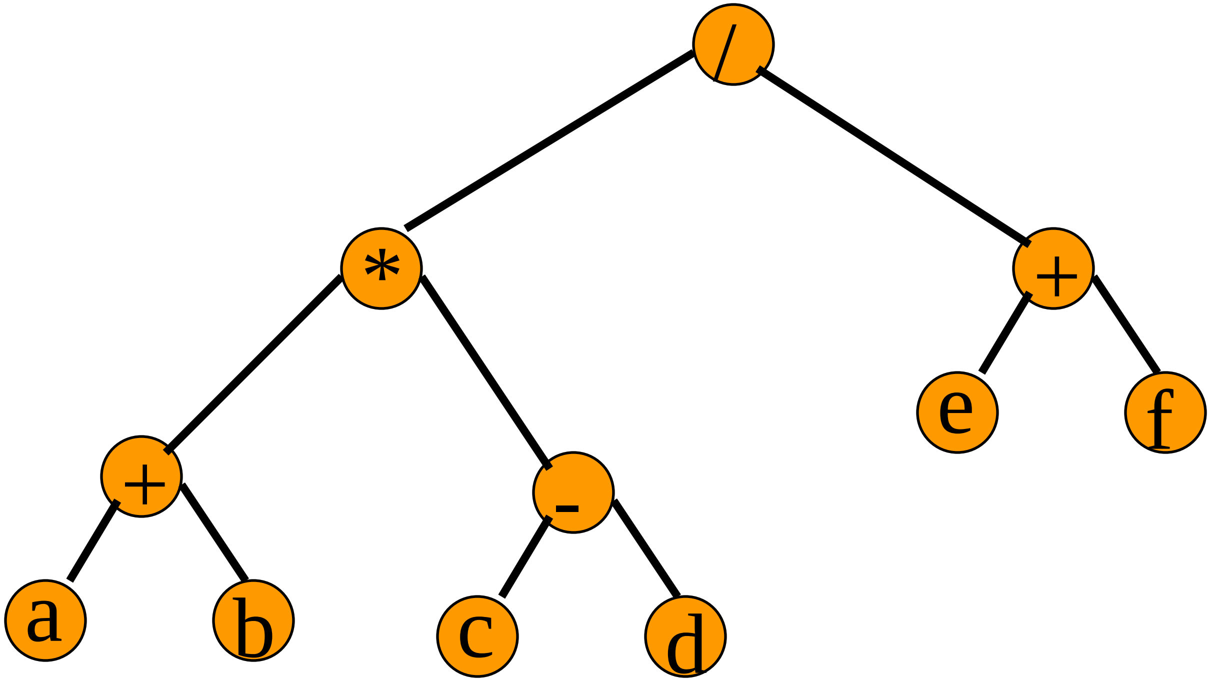
a b c

Preorder жишээ (visit = print)



a b d g h e i c f j

Preorder илэрхийллийн мод



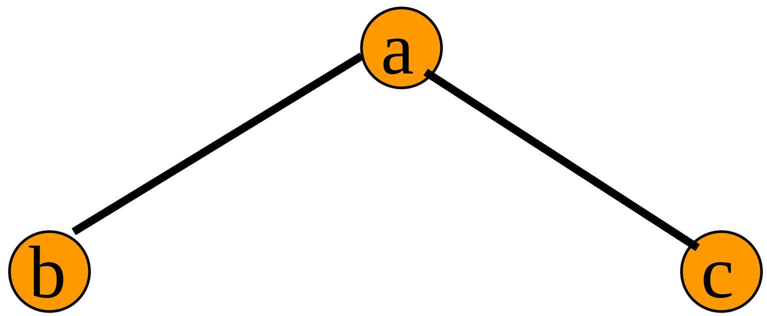
/ * + a b - c d + e f

Энэ мод илэрхийллийн prefix хэлбэрийг өгнө!

Inorder нэвтрэлт

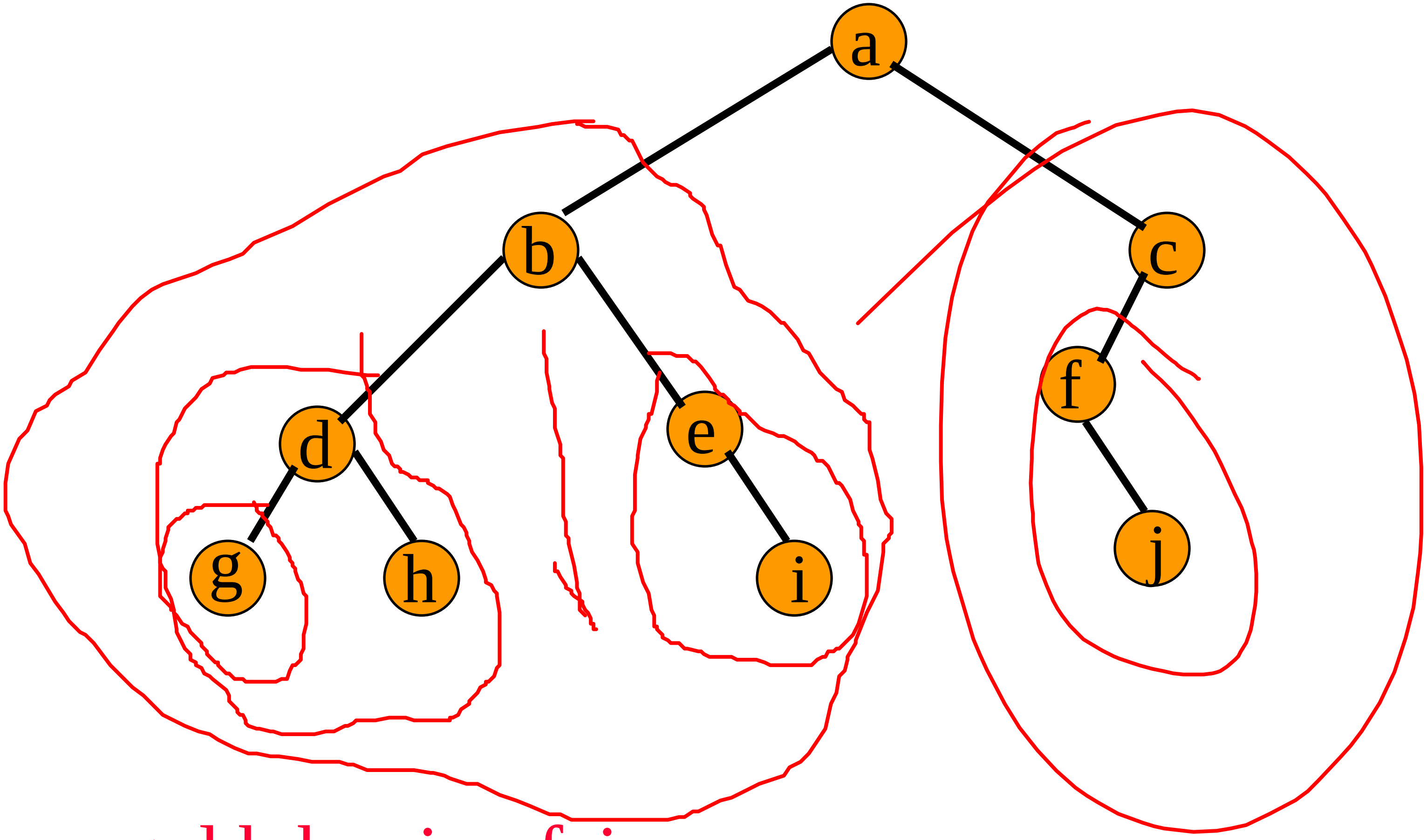
```
public static void inOrder(BinaryTreeNode t)
{
    if (t != null)
    {
        inOrder(t.leftChild);
        visit(t);
        inOrder(t.rightChild);
    }
}
```

Inorder жишээ (visit = print)



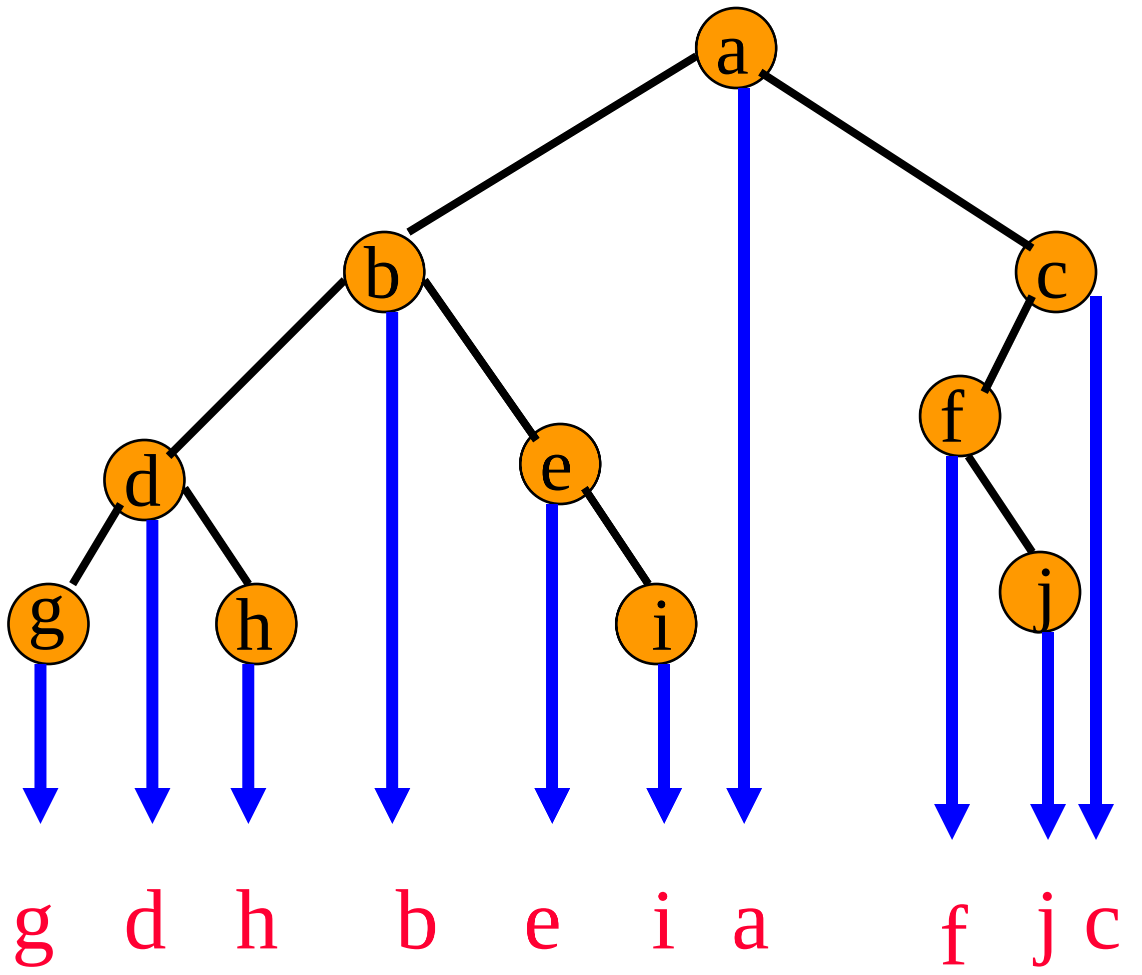
b a c

Inorder жишээ (visit = print)

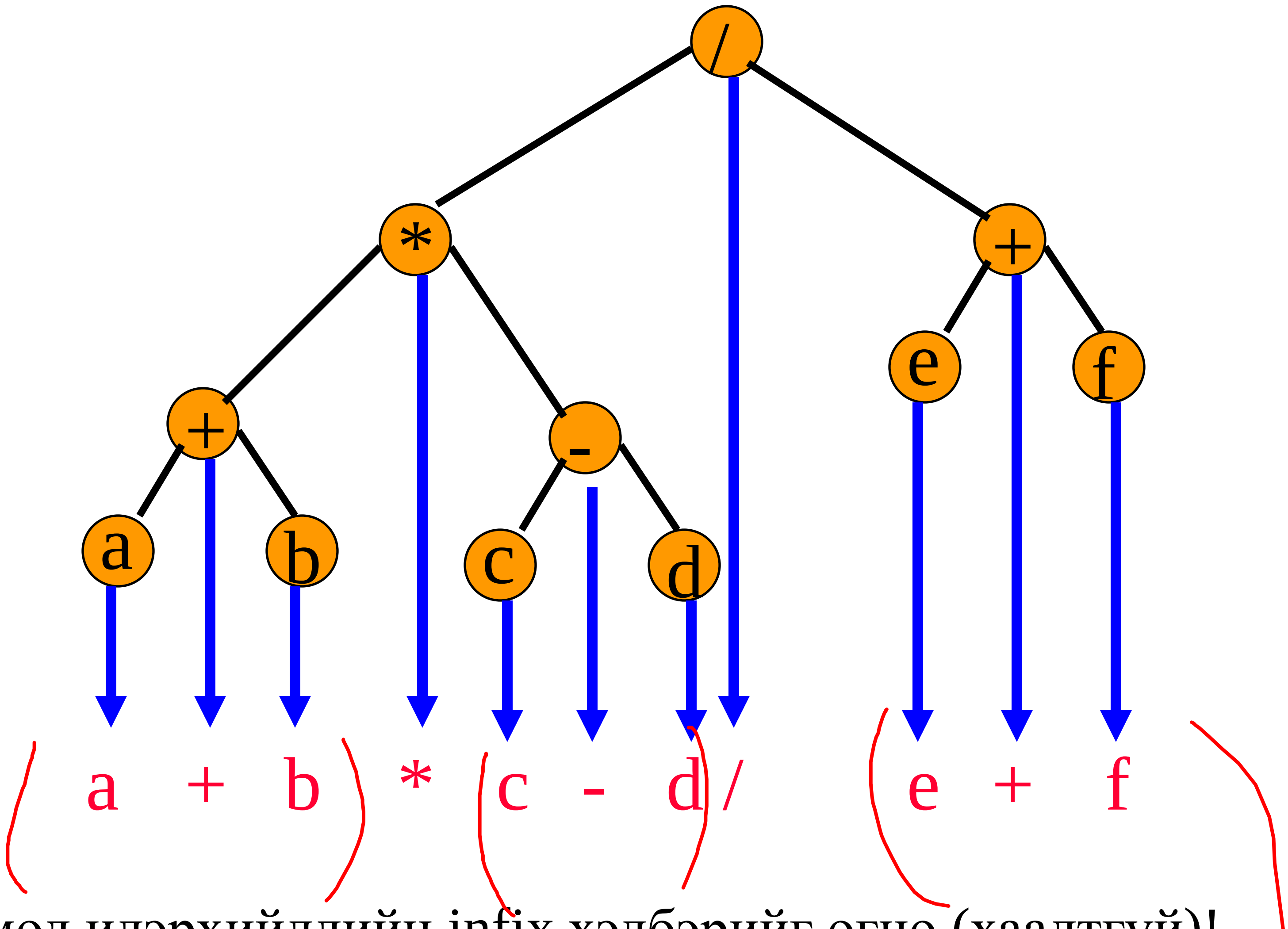


g d h b e i a f j c

Inorder тусгалаар (Squishing)



Inorder илэрхийллийн мод

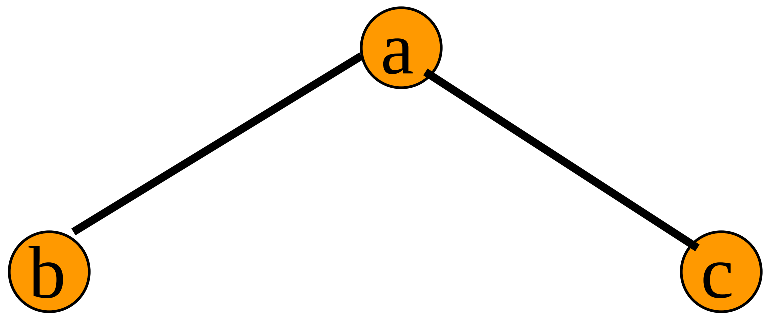


Энэ мод илэрхийллийн infix хэлбэрийг өгнө (хаалтгүй)!

Postorder нэвтрэлт

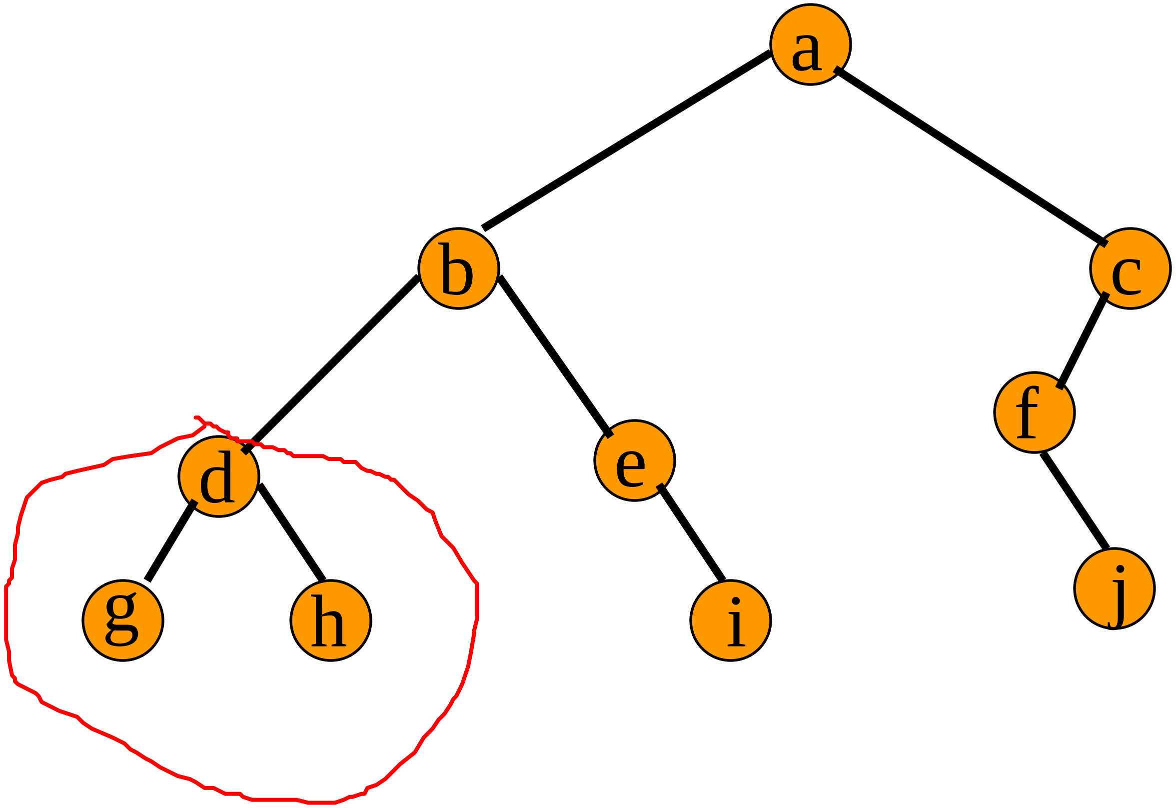
```
public static void postOrder(BinaryTreeNode t)
{
    if (t != null)
    {
        postOrder(t.leftChild);
        postOrder(t.rightChild);
        visit(t);
    }
}
```

Postorder жишээ (visit = print)



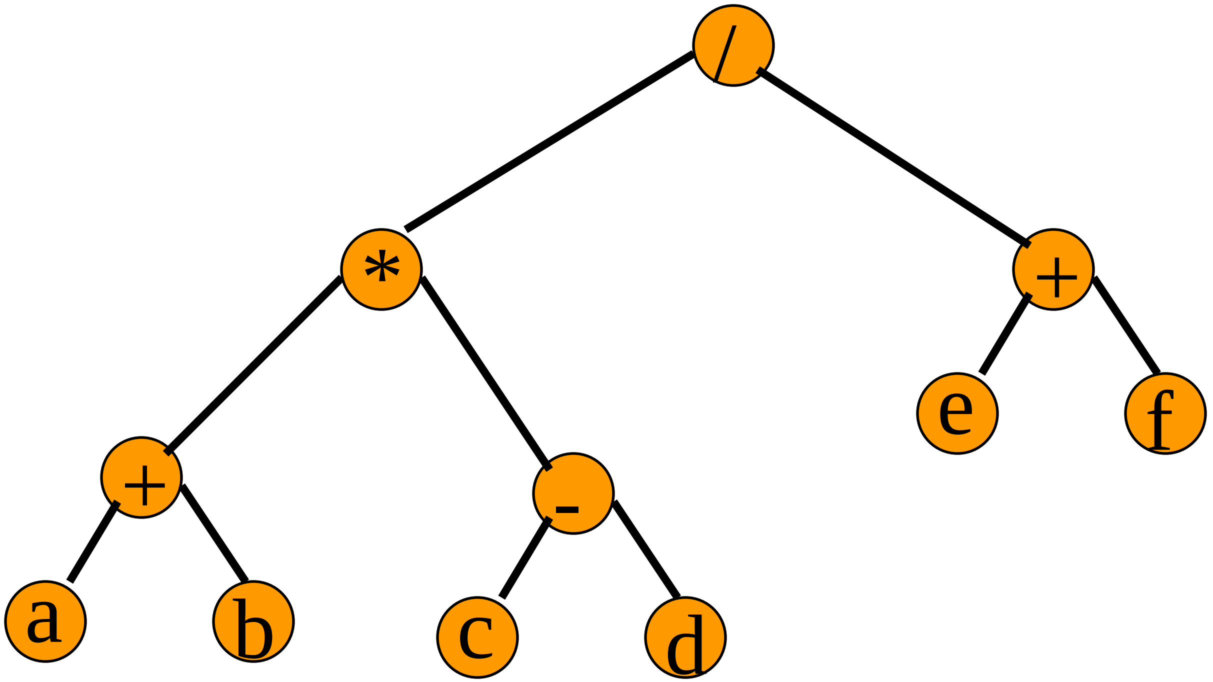
b c a

Postorder жишээ (visit = print)



g h d i e b j f c a

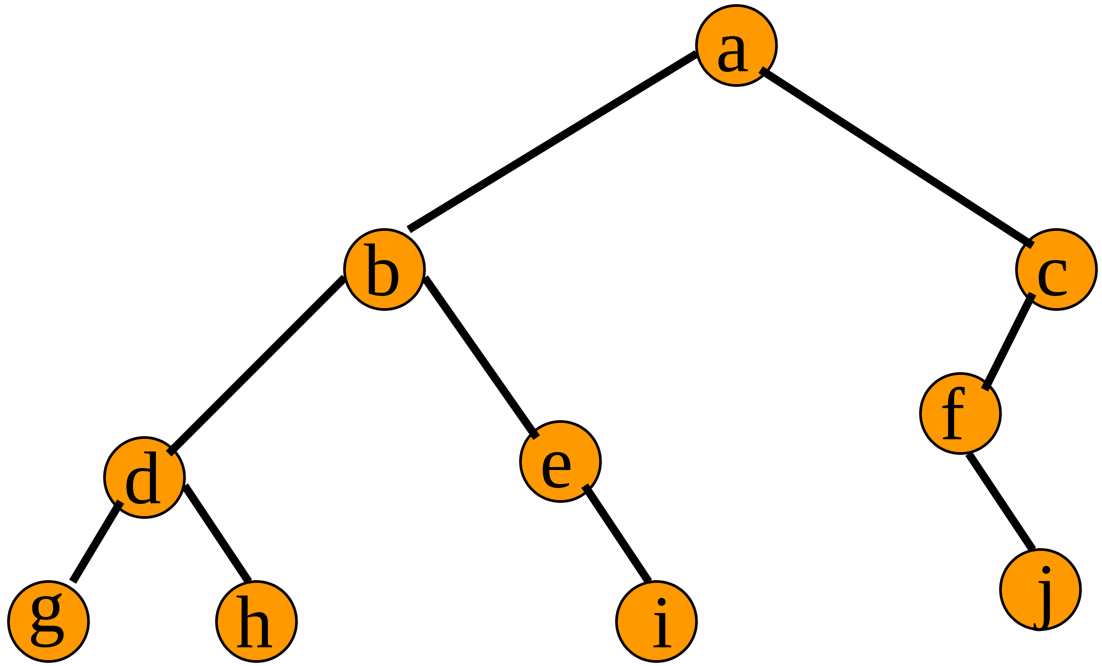
Postorder илэрхийллийн мод



a b + c d - * e f + /

Энэ мод илэрхийллийн postfix хэлбэрийг өгнө!

Нэвтрэлтийн хэрэглээ



- Хувилах - clone.
- Өндрийг олох.
- Зангилааны тоог олох.

LevelOrder нэвтрэлт

t модны үндэс.

while (t != null)

{

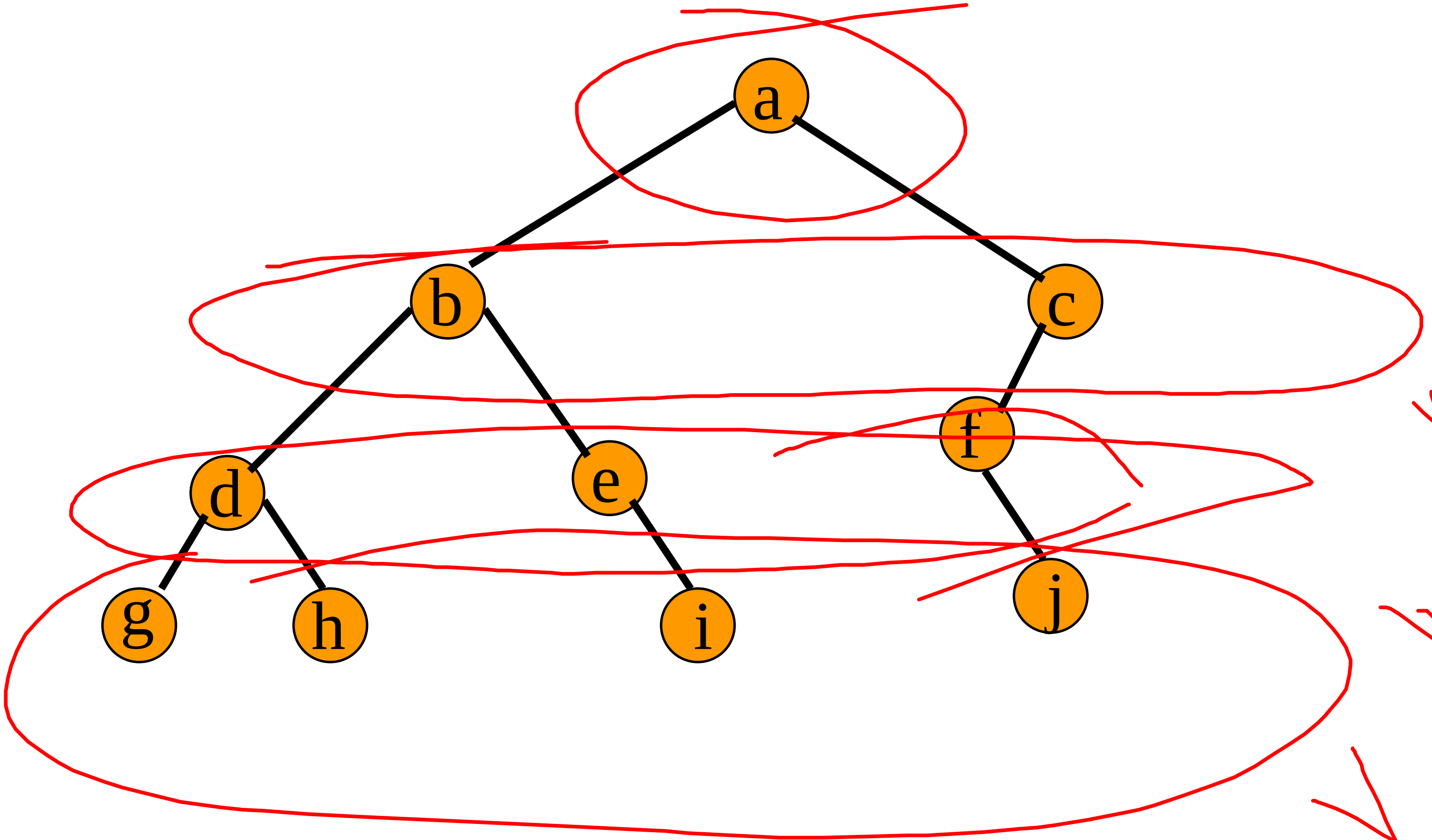
t –д зочлоод хүүхдүүдийг нь FIFO дараалалд хийнэ;

зангилааг FIFO дарааллаас устгаж, дуудна t;

// дараалал хоосон бол устгал null –г буцаана

}

Level-Order жишээ (visit = print)



a b c d e f g h i j

Хоёртын модыг байгуулах

- Хоёртын модны элемент бүр ялгаатай гэж тооцьё.
- Өгөгдсөн нэвтрэлтийн дарааллаар хоёртын модыг байгуулж болох уу?
- Нэвтрэлтийн дараалалд нэгээс их элемент байгаа бол цорын ганц хоёртын мод байхгүй.
- Иймд гарсаж авсан дарааллаар яг тэр чигээр модыг сэргээх байгуулах боломжгүй.

Зарим жишээ

preorder

= ab

inorder

= ab

postorder

= ab

level order

= ab

Хоёртын модыг байгуулах

- Өгөгдсөн нэвтрэлтийн хоёр дарааллаар хоёртын модыг байгуулж болох уу?
- Ямар хоёр дараалал өгөгдсөнөөс хамаарна.

Preorder ба Postorder

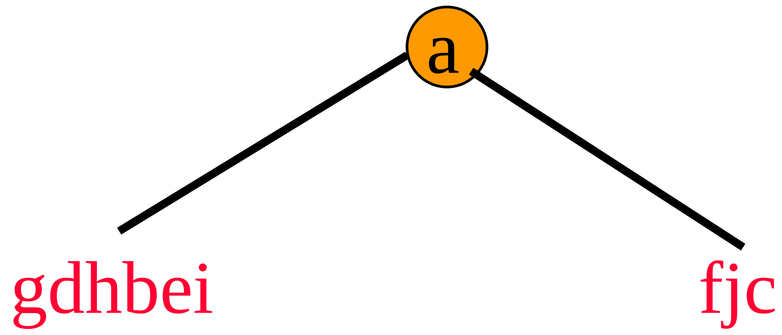
preorder = ab

postorder = ba

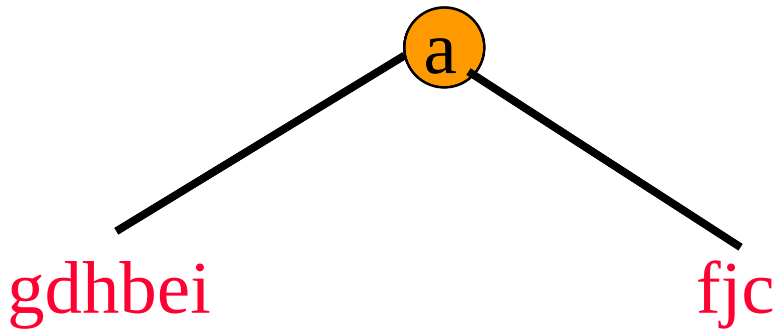
- Preorder ба postorder давтагдашгүй хоёртын модыг тодорхойлж болохгүй.
- Preorder ба level order –оор болохгүй(дээрх жишээ).
- Postorder ба level order –оор болохгүй(дээрх жишээ).

Inorder ба Preorder

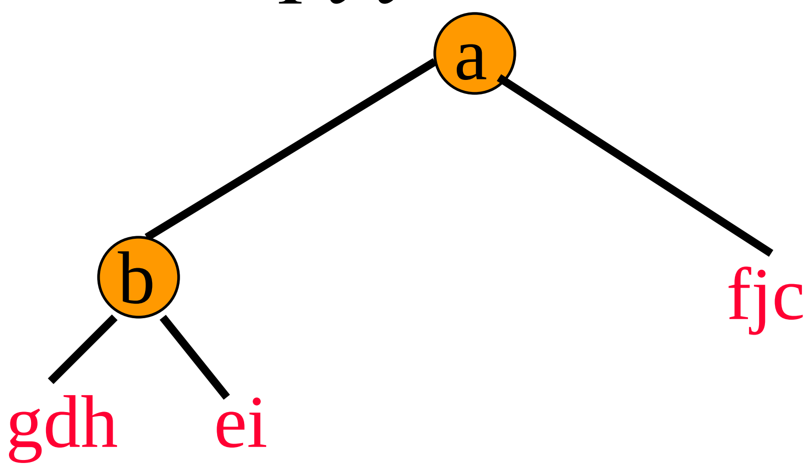
- inorder = g d h b e i a f j c
- preorder = a b d g h e i c f j
- preorder –г зүүнээс баруун тийш шинжэхдээ inorder –г ашиглаж зүүн, баруун дэд моднуудыг салгана.
- a бол үндэс; gdhbei зүүн дэд мод; fjc баруун дэд мод.



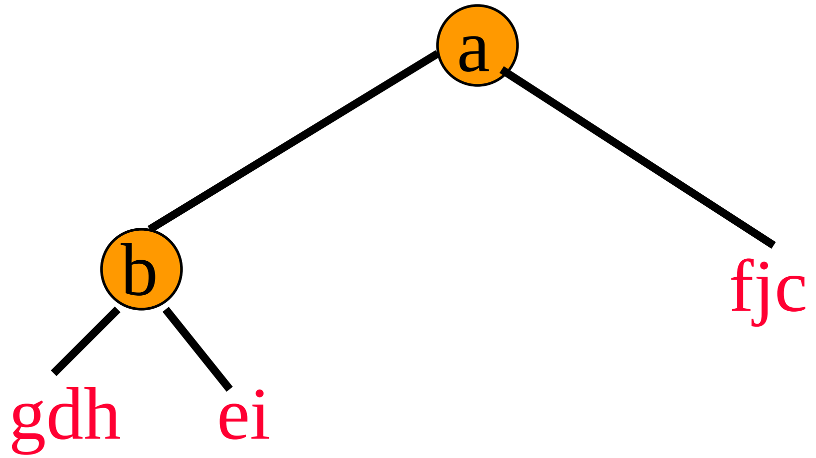
Inorder ба Preorder



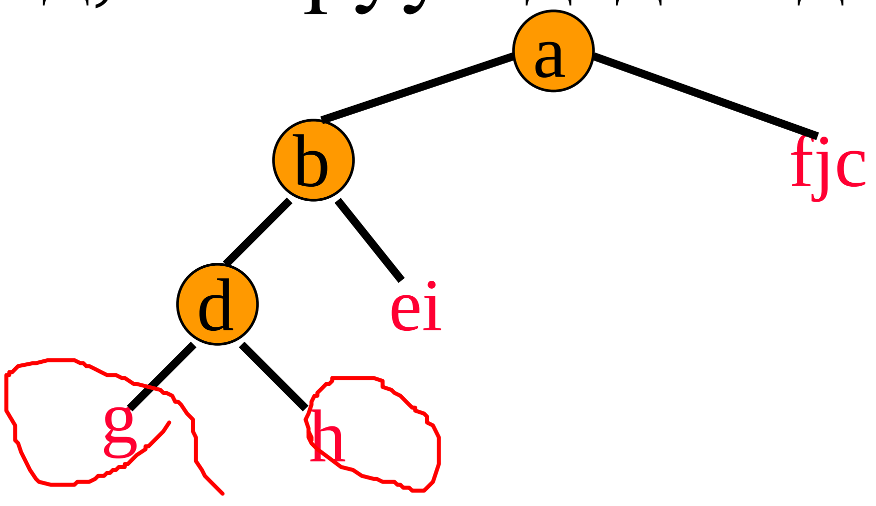
- preorder = a b d g h e i c f j
- b бол дараачийн үндэс; gdh зүүн дэд мод; ei баруун дэд мод.



Inorder ба Preorder



- preorder = a b d g h e i c f j
- d бол дараачийн үндэс; g зүүн дэд мод; h баруун дэд мод.



Inorder ба Postorder

- postorder –г баруунаас зүүн тийш шинжээхдээ inorder –г ашиглаж зүүн, баруун дэд моднуудыг салгана.
- inorder = g d h b e i a f j c
- postorder = g h d i e b j f c a
- a модны үндэс; gdhbei зүүн дэд мод; fjc баруун дэд мод.

Inorder ба Level Order

- level order –г зүүнээс баруун тийш шүүрдэхдээ inorder –г ашиглаж зүүн, баруун дэд моднуудыг салгана.
- inorder = g d h b e i a f j c
- level order = a b c d e f g h i j
- Модны үндэс a; gdhbei зүүн дэд мод; fjc баруун дэд мод.