

Даалгавар 1

МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ

Хэрэглээний Шинжлэх Ухаан Инженерчлэлийн Сургууль

Леки, семинар: Dipl.-Ing., Dipl.-Inf. Б.Батням



Нийт оноо 40. (улирлын эцсийн үнэлгээнд хувьчилж тооцогдоно)

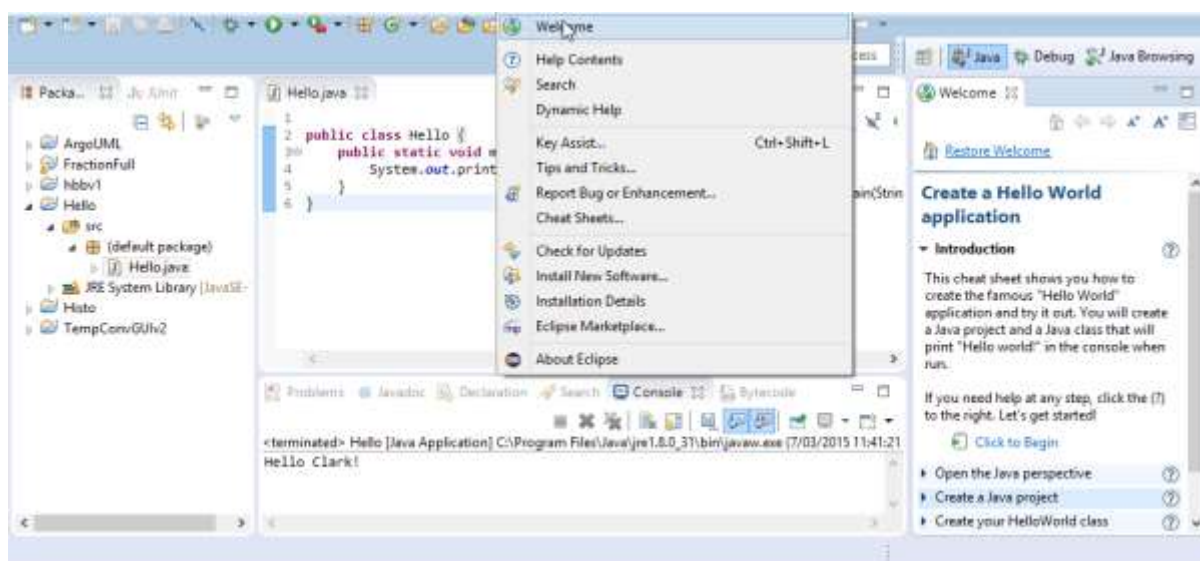
Зорилго: Жава програмчлалын хэл, Eclipse IDE болон ОХЗохиомжийн суурь ойлголтыг сэргээх, бататгах.

Дасгал 1: Өөрийн орчинг тохируулах.

Дараах програмыг зааврын дагуу татаж суулгана.

1. Java SE 8 runtime environment (JRE)
2. Java development kit (JDK) (SE7 or SE8)
3. Eclipse (Luna)

Eclipse дээр “Hello world” програм үүсгэн ажиллуулна. Тус програм нь Help/Welcome цэсэнд байна.



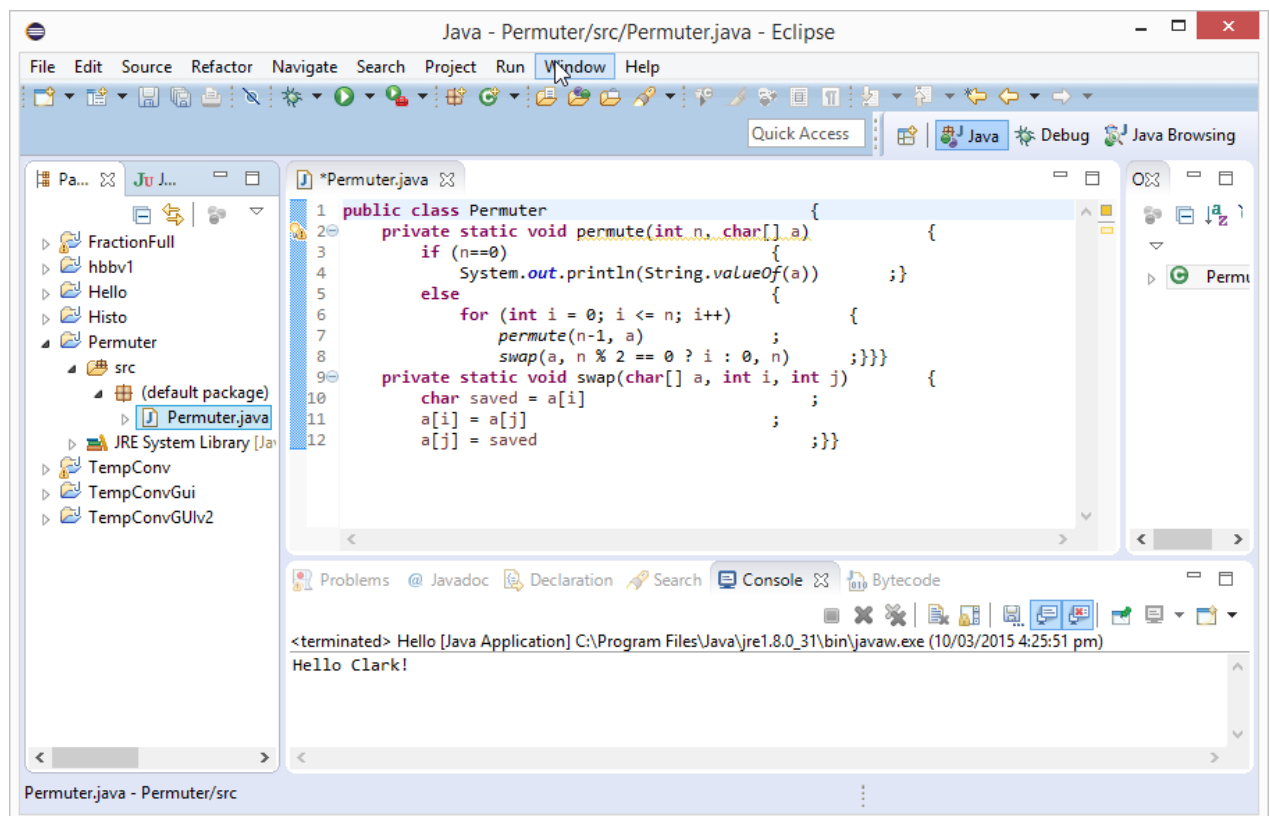
Дасгал 2: Eclipse болон Жавагийн удиртгал дасгал.

- 1) (1 оноо) Шинэ word баримт үүсгэн **өөрийн нэр, оюутны дугаар, даалгаврын дугаарыг** эхний хуудсанд бич.
- 2) (4 оноо) Дасгал 1-д гүйцэтгэсэн “Hello world” програмыг өөрчлөн “Hello XXX” болгон хэвлэдэг болго. Энд XXX нь таны нэр байх болно. Програмын үр дүнгийн дэлгэцийн зургийг баримтад хавсарга.

- 3) Жавагийн гарын авлага “[A Closer Look at the "Hello World!" Application](#)” –ийн “Source Code Comments” хэсгийг унш. Өмнөх асуулт дээр үүсгэсэн Hello.java класст гүйцэтгэж буй үйл ажиллагаанд тохирох баримтын тайлбар үүсгэнэ (монгол хэлээр галиглаж бич). Hello.java файлын бүх агуулгыг хуулбарлан баримтад хавсаргана.
- 4) (4 оноо) “Permuter” нэртэй классыг агуулсан permuter нэртэй прожект үүсгэ. Eclipse-ээр автоматаар үүсгэсэн кодыг дараах Python форматаар өгөгдсөн кодоор сольж бич.

```
public class Permuter
{
    private static void permute(int n, char[] a)
    {
        if (n==0)
            System.out.println(String.valueOf(a));
        else
            for (int i = 0; i <= n; i++)
                permute(n-1, a);
        swap(a, n % 2 == 0 ? i : 0, n);
    }
    private static void swap(char[] a, int i, int j)
    {
        char saved = a[i];
        a[i] = a[j];
        a[j] = saved;
    }
}
```

Таны Eclipse дараах хэлбэрээр харагдана.



Тус код нь Python хэлбэржүүлэлттэй байгаа учир тус кодын хэлбэрийг Source/format командыг ашиглан засна.

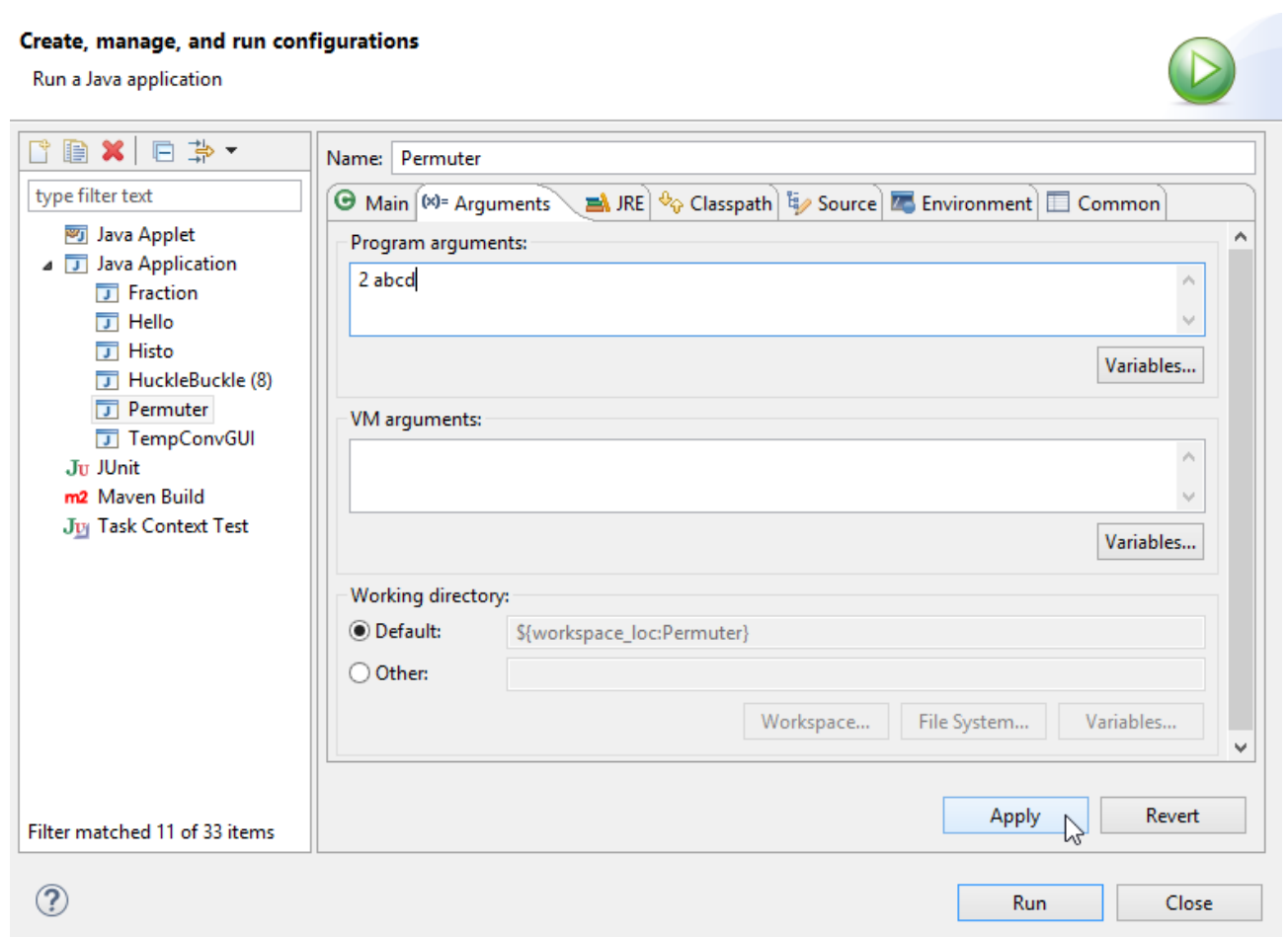
Тус програмын 2 дах мөрд анхааруулга шар зурвас гарч байна. Хулганаар тус зурвас дээр дарж уншина. Тус зурвас нь permute() аргыг тус классын бусад арга дотор дуудаж ашиглаагүй мөн тус арга нь хувийн арга учир өөр ямар нэгэн классаас ашиглах боломжгүй болохыг сануулна. Тус аргыг дуудах main() аргыг тодорхойлсноор энэхүү асуудлыг шийдвэрлэх боломжтой.

Permuter.java классад дараах main() аргыг тодорхойлж өг.

```
public static void main(String[] args) {  
    permute(Integer.parseInt(args[0]), args[1].toCharArray());  
}
```

Permuter програмыг компайлдаж ажиллуулна. Тус програмыг ажиллуулах явцад `ArrayIndexOutOfBoundsException` алдаа гарах бөгөөд энэ нь ямар нэгэн аргумент дамжуулж өгөөгүйгээс үүдэлтэй. Тус `main()` арга нь `n` гэсэн бүхэл тоон болон `n`-ээс багагүй урттай тэмдэгт мөр параметрт авна.

Тиймээс “Run/Run Configurations...” гэсэн цэсийг сонгож тохируулга хийнэ. “Program Arguments” гэсэн талбарт “2 abcd” гэж бичиж өгөөд “Apply” болон “Run” товчийг дарж ажиллуулна.



- a) Даалгавар: Програмын үр дүнг текст хэлбэрээр хуулж баримтад хавсаргана.
- b) Даалгавар: Дээрх програмын аргументыг өөрчлөн дараах асуултад хариулна.
 - i. Таны сонгосон програмын аргумент юу вэ?
 - ii. Тус аргументыг оруулан ажиллуулахад гарсан үр дүн эсвэл алдааны зурвасыг оруулж хэрхэн ажилласан талаар тайлбарлана уу?

iii. Яагаад энэ аргументыг сонгосныг тайлбарлана.

- 5) (4 оноо) Эхлээд дараах холбоосноос класс гэж юу вэ? Объект гэж юу вэ? гэдгийг [эндээс](#) уншиж танилцана. Мөн *Variable* болон *Operator*-ийн талаар уншаарай.

Дараа нь <http://fpl.cs.depaul.edu/jriely/java4python.pdf> -аас Section 4.3-ийг уншиж 4 мөр Python кодыг 19 мөр TempConv.java файл хэрхэн болгосныг олж харна. Мөн importing packages, declaring variables, and reading console input буюу хэрхэн багц импорт хийх, хувьсагч хэрхэн зарлах, хэрхэн консолоос утга унших талаар олж уншина.

TempConv нэртэй прожект үүсгэнэ. TempConv.java класс үүсгэн 19 мөр код бүхий жишээг хуулж бичиж ажиллуулна. (Консолын оролтын хэсэгт хулганаар товшиж оролтын утгыг өгнө.)

- а) Даалгавар: Програмыг ажиллуулж оролтын утга өгч оролт болон гаралтын үр дүнг баримтад хавсаргана. 2 дах println()-ийг дараах байдлаар өөрчил. String.format("%.1f", cel);
- б) Даалгавар: Өмнөх өөрчилсөн програмыг ажиллуулж оролтын утга өгч оролт болон гаралтын үр дүнг баримтад хавсаргана.
- 6) (4 оноо) Өмнөх Section 4.3-ийг үргэлжлүүлэн уншина. TempConvGUI гэсэн Swing програмыг олж Eclipse-д үүсгэн ажиллуулна.

Баримтын тайлбарыг оруулна. Үүний тул д кодын эхний нэг мөрийг сонгон Source/Generate Element Comment гэсэн сонголтыг хийнэ. Энэ нь JavaDoc-д @author гэсэн шошгийг нэмдэг бөгөөд өөрийн нэрээр дараах тайлбарыг үүсгэж хэлбэржүүлнэ.

```
/**
 * GUI which converts a temperature from °F to °C
 * <p>
 * Adapted from http://interactivepython.org/courselib/static/java4python
 *
 * @author Clark Thomborson (cth0065)
 * @version 1.0
 */
```

- Даалгавар: Тодорхойгүй давталтаар олон утгыг дараалан хөрвүүлэх боломжтой болгоно. Оролтын утга 0 үед шалгалт хийж ямар нэгэн утгагүй дан °C –г хэвлэхгүй байлгана.
- Кодын тайлбарыг өөрчлөн засч Version –ийн хувилбарыг шинэчилж бичнэ.

Эх код болон програмын шинэ боломжуудыг (оролт гаралт) харуулсан дэлгэцийн зургуудыг баримтад хавсаргана.

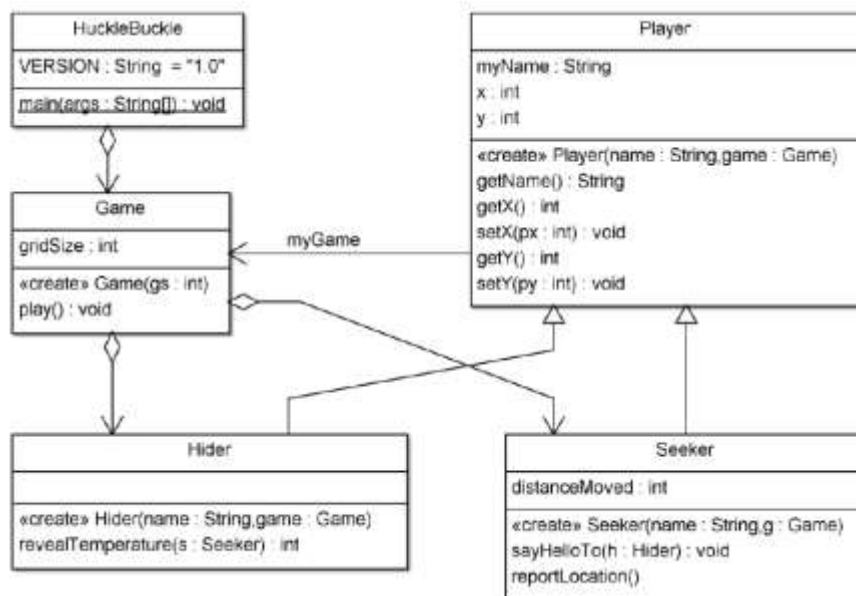
Дасгал 3: JAR файл ажиллуулах

- 7) (4 оноо) [binary distribution of ArgoUML v0.34](#) -ийг татаж суулга. *.zargo файлыг хэрхэн нээх болон .png файл хэрхэн экспорт хийх талаар судална. umlexamples.zargo файлыг нээн Class Diagram 7 –г зураг болгон баримтад хавсаргана.

- 8) (4 оноо) Company гэсэн нэртэй классын диаграмыг шинээр үүсгээд Class Diagram 7-г дуурайлган зурна. increaseEmployeePay() аргыг Manager класст нэмж өгнө. Employee нь Manager биш бөгөөд Manager л зөвхөн хэн нэгний цалинг нэмэгдүүлэх боломжтой гэж үзсэн. Тэгвэл хэсэг хугацаанд бодоод шинээр ямар класст ямар арга (method)-ийг нэмж өгөж болохыг тогтооно. Шалтгаанаа өгүүлбэрээр тайлбарлаж, өөрчилж үүсгэсэн класс диаграмыг баримтад хавсаргана.

Дасгал 4: Huckle Buckle Beanstalk

- 9) (4 оноо) Huckle Buckle Beanstalk буюу hbbv1.jar файлыг Eclipse-д импорт хийн оруулж ямар нэг аргумент өгөлгүй ажиллуулна. Тус кодоос дараах класс диаграмыг хэрхэн хэрэгжүүлсэнийг харна.



- 10) (4 оноо) [Хэрхэн *.jar файл үүсгэх](#) талаар уншиж судална. Jar файл нь шууд ажиллуулах боломжтой болон боломжгүй гэсэн 2 төрөл байдаг.

Hbbv1.1 гэсэн шинэ прожект үүсгээд Hbbv1-ээс бүх эх кодыг импорт хийж хувилбарын дугаарыг 1.1 болгож өөрчилж ажиллуулна.

Дараах баримтын тайлбарыг HuckleBuckle.java файлд нэмж өг.

```

* <p>
* Second optional command-line parameter: ngames, the number of games to be
* played on this run of the program, an integer between 0 and 9 inclusive.
* Note: if only one command-line parameter is supplied, it is interpreted as
* gridsize rather than as ngames, as indicated by the nested bracketing 9
* on the usage comment below.
* <p>
* Usage: hbbv1.1 [gridsize [ngames]]
    
```

Дээрх тайлбар тодорхойлолтын дагуу эх кодыг өөрчилнө. (хэрвээ 2 аргументтай байвал ngames-д 2-р аргументын утгыг өгөх бөгөөд энэ нь тоглоомыг хичнээн удаа давтан тоглохыг заана.)

- Hbbv1.1-ийн эх кодыг тайлбарлан оруулна.
- Тоглоомыг ngames-ийн тоогоор давтан тоглуулдаг болгосон даталтыг харуулна.
- Тус програмд 3 2 гэсэн аргумент дамжуулан өгч ажилласан үр дүнг оруулна.
- Тус програмд 4 0 гэсэн аргумент дамжуулан өгч ажилласан үр дүнг оруулна.

Тоглоомыг олон дахин тоглодог болгохдоо тоглоом эхлэж буйг харуулах зорилгоор “Дахин тоглоцгоое, дараа тоглоцгоое” гэсэн нэмэлтүүд оруулаарай.

11) (3 оноо) Seeker классын seek арга дотор if() –ийг дахин давтан ашигласан байна. Hbbv1.2 гэсэн шинэ прожект үүсгээд hbbv1.1-ийг хуулж өг. Үүний дараа Seeker классын seek аргыг өөрчлөн moveTo() гэсэн шинэ байрлал руу зөөх аргыг тодорхойлж өгнө. Seek арга нь дараах мөр кодтой болно.

```
for (int y = 0; y < getGame().getGridSize(); y++) {  
    for (int cx = 0; cx < getGame().getGridSize(); cx++) {  
        // cx: counts the number of times I have moved on row y  
        int x = (moveRight ? cx : getGame().getGridSize() - cx - 1);  
        moveTo(x,y);  
    }  
}
```

тус moveTo() арга нь талбарын дурын байрлалд зөөх боломжтой бөгөөд тус арга нь distanceMoved –ийн утгыг хуучин байрлалаас шинэ байрлал хүртэлх зайгаар нэмэгдүүлнэ. Нэмэгдүүлсэн утга нь талбарын хэмжээнээс хэтэрсэн эсэх нь чухал биш гэж үзээрэй. Мөн тус арга нь байрлалыг өөрчилсөний дараа хаана очсоныг мэдээлдэг байна. (reportLocation())

Тус аргад distance() арга хэрэг болох бөгөөд энэ нь Hider классд байгаа бөгөөд үүнийг Player класс руу зөөж бичсэнээр Seeker класс ашиглах боломжтой болно. Мөн нэгэнт бичсэн кодыг 2 газар давхардуулан бичих шаардлагагүй. Өмнө бичсэн кодыг сайжруулах (шинэ арга үүсгэх, зөөх,...) –ийг re-factoring буюу дахин байгуулах гэнэ. Энэ нь програм бичихэд шаардлагатай чухал чадварын нэг юм.

Баримтад өөрчилсөн Seeker классын эх кодыг бүтнээр хавсаргана.