

Лабораторийн ажил 9: Алдаа барих, файлтай ажиллах

Лабораторийн ажлын зорилго:

Алдаа барих блокыг бичих, файл руу өгөгдөл бичих, файлаас өгөгдөл унших үйлдлүүдийг гүйцэтгэж сурах

Лабораторийн ажлын суралцахуйн үр дүнгүүд:

Энэ лабораторийн ажлыг гүйцэтгэснээр оюутан дараах чадваруудтай болсон байна.

д/д	Суралцахуйн үр дүнгүүд	Суралцахуйн үр дүнг илэрхийлэх үйл үг	Суралцахуйн үр дүнгийн түвшин (Блумын)	CLOs хамаарал
1	Алдаа барих	Ашиглах (Use)	Хэрэглээ	2, 8
2	Файл руу байт, анхдагч өгөгдлийн төрөл, тэмдэгт мөр хэлбэрээр өгөгдөл бичих	Ашиглах (Use)	Хэрэглээ	
3	Файлаас өгөгдөл унших	Ашиглах (Use)	Хэрэглээ	2, 8
4	Файл руу объект бичих, унших	Ашиглах (Use)	Хэрэглээ	2, 8
5	ЮМЛ диаграм зурах	Зурах (Draw)	Синтез	11
6	Тайлан бичих	Зохион бичих (Compose)	Синтез	12
7	Тайлан хамгаалах	Хамгаалах (defend)	Синтез	12
8	Англи хэл дээр холбогдох материал бусад эх үүсвэрээс унших	Унших (Read)	Ойлголт	13

Ашиглах програм хангамж/техник хангамж, бусад хэрэглэгдэхүүнүүд:

Лабораторийн компьютер эсвэл өөрийн зөөврийн компьютерийг ашиглана.

Онолын ойлголтууд:

Алдаа барих

Exception нь програмын энгийн ажиллагааг тасалдуулагч event юм. Програмд алдаа гарахад програмын ажиллагаа зогсож, системийн алдааны мэдээллийг харуулдаг. Харин алдааг барьснаар програмын ажиллагаа зогсохоос сэргийлж, алдааны мэдээллийг хэрэглэгчид ойлгомжтой байдлаар харуулах боломжтой болно.

Жишээ:

```
System.out.println("Гарчгийн дугаарыг оруулна уу");  
id = sc.nextInt();
```

Зураг 1. Гараас утга оруулах кодын хэсэг

Дээрх жишээнд бид гараас int тоо унших ёстой байна. Харин гараас тэмдэгт оруулахад доорх системийн алдаа гарч програмын ажиллагаа зогсоно.

```
Гарчгийн дугаарыг оруулна уу
w
Can only enter input while your programming is running
java.util.InputMismatchException
    at java.util.Scanner.throwFor(Scanner.java:864)
    at java.util.Scanner.next(Scanner.java:1485)
    at java.util.Scanner.nextInt(Scanner.java:2117)
    at java.util.Scanner.nextInt(Scanner.java:2076)
    at Selling.main(Selling.java:40)
```

Зураг 2. Алдааны мэдээлэл

Дээрх кодонд try{ } catch{ } блокыг нэмж алдааг барина.

```
try {
    System.out.println("Гарчгийн дугаарыг оруулна уу");
    id = sc.nextInt();
}
catch(Exception eo) {
    System.out.println("Гарчгийн дугаарыг тоогоор оруулна уу");
}
```

Зураг 3. Алдаа барих блокыг кодонд нэмсэн байдал

Програмыг ажиллуулж өмнөх туршилттай ижил тэмдэгт оруулахад алдааг барьж, алдааны мэдээллийг харуулаад програмын ажиллагаа тасалдалгүй цааш үргэлжлэн ажилласан байна.

```
Гарчгийн дугаарыг оруулна уу
e
Гарчгийн дугаарыг тоогоор оруулна уу
Гарчгийн дугаарыг оруулна уу
Гарчгийн дугаарыг тоогоор оруулна уу
Авах номын дугаар:2
Авах номын тоо:20
```

Зураг 4. Програмын үр дүн

Алдааг дараах шатлалаар барина.

```
try
{
    .....
}
catch(ArithmeticException e1)
{
    //арифметик алдаа үүсэхэд
}
```

```

catch(NullPointerException e2)
{    //хоосон заагчийн алдаа    }
catch(FileNotFoundException e3)
{    //файл байхгүй байх үеийн алдаа    }
catch(IOException e4)
{    //оролт гаралтын үеийн алдаа    }
catch(RuntimeException e5)
{    //ажиллах үеийн алдаа    }
catch(Exception e6)
{    //Бүх алдааг барьж чадна. Өмнөх блокуудаар алдаа барьж
чадаагүй үед
}
finally
{    //үргэлж хамгийн сүүлд ажиллана.Алдаа гарсан ч дараа нь
ажиллана.
}

```

Файлтай ажиллах

- `FileOutputStream`, `FileInputStream` – байтуудыг файлд бичих, файлаас унших
- `DataOutputStream`, `DataInputStream` – анхдагч өгөгдлийн төрөл бүхий утгыг файлд бичих, файлаас унших
- `PrintWriter`, `BufferedReader` текст өгөгдлийг файлд бичих, файлаас унших
- `Scanner` ашиглан текст файл унших
- `ObjectOutputStream`, `ObjectInputStream` – объектыг файлд бичих, файлаас унших
- `FileReader` – текстийг тэмдэгтийн урсгал байдлаар уншина

Файлтай ажиллахдаа эдгээрээс алиныг нь ашиглах вэ гэдэг нь ашиглаж буй Жавагийн хувилбар, байт унших уу, тэмдэгт унших уу, файлын хэмжээ зэргээс хамаарна.

File классын зарим методууд:

exists()	файл байгаа эсэх. Байвал True, байхгүй бол False
getName()	Файлын нэр
getAbsolutePath()	Файлын абсолют зам
canWrite()	Файлд бичих боломжтой эсэх
canRead()	Файлыг унших боломжтой эсэх
length()	Файлын хэмжээ
list()	Хавтас байх үед доторх файлын жагсаалтыг харуулна

setReadable(boolean)	файлыг унших горимд оруулах
SetWritable(boolean)	файлыг бичих горимд оруулах

Жишээ. Байт өгөгдлийг файлд бичих

```
//файл болон stream-г үүсгэх
File outFile = new File("sample1.data");
FileOutputStream outputStream = new FileOutputStream( outFile );
//хадгалах өгөгдөл
byte[] byteArray = {10, 20, 30, 40, 50, 60, 70, 80};
//stream-д өгөгдлийг бичих
outputStream.write( byteArray );
//гаралт дуусахад stream-г хаана
outputStream.close();
```

Жишээ: Scanner ашиглан текст файлаас унших

```
import java.io.*;
import java.util.Scanner;
public class FileIn {
    public static void main(String[] args) {
        try {
            File file = new File("test.txt");    //test.txt файлаас уншина
            Scanner sc = new Scanner(file);
            while (sc.hasNextLine()) {          //унших мөр байгаа үед давтана
                String data = sc.nextLine();    //мөрийн өгөгдлийг уншиж data
                                                хувьсагчид хадгална.
                System.out.println(data);      //data-н утгыг хэвлэнэ
            }
            sc.close();
        } catch (FileNotFoundException e) {
            System.out.println("Ийм файл олдохгүй байна");
        }
    }
}
```

Жишээ: Объектыг файлд бичих

Pictures класс

```
import java.io.Serializable;

public class Pictures implements Serializable    //тухайн классын
                                                объектыг файл руу бичих бол заавал бичнэ

{
    private int id;
    private String name;
    public Pictures(int i, String name) {
        id = i;
        this.name = name;
    }
    public String getName() {
        return name;
    }
    public String toString() {
        return "id: " + id + "\t name: " + name;
    }
}
```

FileOut класс

```
import java.io.*;

public class FileOut
{
    public static void main(String[] args) throws IOException {
        //файл руу бичих объектуудаа үүсгэнэ
        Pictures p1 = new Pictures(1,"Image1");
        Pictures p2 = new Pictures(2,"Image2");
        //файлтай ажиллах үед гарах алдааг барихын тулд try блокт
        бичье
        try {
            FileOutputStream f =
                new FileOutputStream(new File("test.txt"));
            ObjectOutputStream o = new ObjectOutputStream(f);
            //өмнө үүсгэсэн picture объектуудыг файл руу бичье
            o.writeObject(p1);
        }
```

```

o.writeObject(p2);
o.close();
//бичсэн объектуудаа файлаас буцаан уншъя
FileInputStream fIn =
    new FileInputStream(new File("test.txt"));
ObjectInputStream oIn = new ObjectInputStream(fIn);
//объект уншиж байгаа тул объектод хадгална.
p2 =(Pictures)oIn.readObject();
p1=(Pictures)oIn.readObject();
oIn.close();
//объектуудын мэдээллийг хэвлэх
System.out.println("p2\t"+p2);
System.out.println("p1\t"+p1);
}
catch(FileNotFoundException e1) {
    System.out.println("Файл байхгүй");
}
catch(IOException e2) {
    System.out.println("Файл руу бичих үед алдаа гарлаа");
}
catch(ClassNotFoundException e3) {
    System.out.println("Класс байхгүй");
}
}
}

```

Ажил гүйцэтгэх дараалал:

1. Өмнө үүсгэсэн классынхаа кодын шаардлагатай хэсгүүдэд try, catch блокыг нэмж, ажиллуулж турших
2. Файл руу байт, тэмдэгт, int, double гэх мэт төрлийн өгөгдлүүдийг бичиж турших
3. Өмнө үүсгэсэн классаас объект үүсгэж, үүнийгээ файл руу бичиж, буцаан уншиж турших