

Дасгал ажил 8

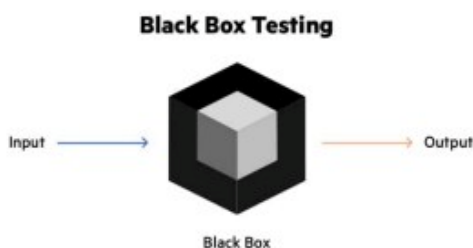
МОНГОЛ УЛСЫН ИХ СУРГУУЛЬ

Хэрэглээний Шинжлэх Ухаан Инженерчлэлийн Сургууль

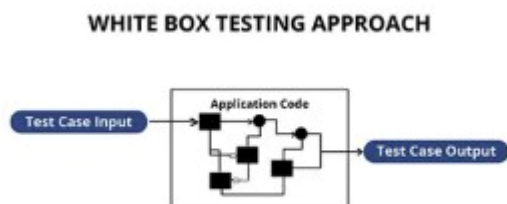
Б.Батням маг., Д.Энхзол маг., Ц.Лхамролом

Хар хайрцагны шалгалт (Black box testing) болон Цагаан хайрцагны шалгалт (White box testing) хоорондын ялгааг тайлбарла.

/Үүрэг хандлагат/ Black box testing нь дотоод бүтэц, дизайн, хэрэгжилт зэргийг шалгадаггүй ба зөвхөн системийн гадаад үйл ажиллагааг үнэлдэг. Системийн хүлээн авсан оролт, түүний гаргаж буй гаралт эсвэл хариуг шалгадаг. Дотоод бүтэц нь мэдэгдэхгүй үед ашиглана.



/Бүтэц хандлагат/ White box testing нь турших гэж буй бүтээгдэхүүний код, бүтцийг судалж дотоод бүтцийн үйл ажиллагаа зөв эсэхийг шалгадаг. Дотоод бүтцэд үндэслэсэн тестын тохиолдлын сонголт юм.



Хар хайрцагны шалгалт (Black box testing) болон Цагаан хайрцагны шалгалт (White box testing) -ын “Алтан стандарт” (Gold standard) гэж юу вэ?

White box testing-ийн Gold standard- Exhaustive Path Testing
Black box testing-ийн Gold standard- Exhaustive Path Testing.

A1) TestJUnit.java нэртэй шалгалтын класс үүсгэ. testPrintMessage() шалгалтын тохиолдолд алдааг хүлээн авах (Adding expected exception) боломжтой болго.

```
@Test
(expected = ArithmeticException.class)
public void testPrintMessage() {
    System.out.println("Inside testPrintMessage()");
    messageUtil.printMessage();
}
```

A2-1) ExpectedException нь шинэ дүрэм тодорхойлох боломж олгодог бөгөөд өөрийн шалгалтад тус

Програм хангамжийн хөгжүүлэлт (ICSI402)

алдаа үүсэх үед алдааны зурвас өгөх боломжтой байдаг. Хэрвээ new Person()-д сөрөг нас өгвөл алдаа гэдгийг мэдээлдэг ExpectedException rule бүхий шалгалтын тохиолдол үүсгэ.

```
@Test
public void testExpectedException() {
    exception.expect(IllegalArgumentException.class);
    exception.expectMessage("Invalid age");
    new Person("Joe", -1);
}
```

A2-2) Дээрх алдааг @Test тэмдэглэгээ ашиглан тодорхойл // сөрөг утга олгож шалгалт хийх

```
@Test(expected = IllegalArgumentException.class)
public void testExpectedException2() {
    new Person("Joe", -1);
}
```

A2-3) try-catch команд ашигла (дээр шалгахад гарсан алдааг барьж авч мэдээлэх)

```
@Test
public void testExpectedException3() {
    try {
        new Person("Joe", -1);
        fail("Should have thrown an IllegalArgumentException because age is invalid!");
    } catch (IllegalArgumentException e) {
        assertEquals(e.getMessage(), "Invalid age:-1");
    }
}
```

A3) JUnit-эд гүйцэтгэлийг шалгах нь хамгийн төвөгтэй байдаг. JUnit4 нь тус асуудлыг бүрэн шийдээгүй ч хэрэгцээтэй тусламжаар хангадаг: параметр нь хугацаанаас хэтэрвэл (timeout) мэдээлэх боломжтой. Хэрвээ шалгалт хийгдэж байхад хугацаа нь дуусвал гүйцэтгэл удаан байна гэж үзнэ. Performance.java –ийн run() функц-д timeout value =500 байхаар турш.

```
@Test(timeout=500)
```

```
PersonTest [Runner: JUnit 5] (0.000 s)
  testExpectedException2 (0.000 s)
  testExpectedException3 (0.000 s)
  testExpectedException (0.000 s)
```