گزارش‌کار تمرین کامپیوتری سوم آزمون نرم‌افزار

مجید فریدفر 810199569

اولدوز نیساری 810199505

لینک ریپازیتوری: https://github.com/maj2idfar/Baloot

هش آخرین کامیت: `6fcc381`

سوال اول

قطعه کد زیر را در نظر بگیرید. آیا می تواند یک آزمایه تولید کنید که پوشش جمله ۱۰۰ درصدی داشته باشد، اما پوشش شاخه ۱۰۰ درصدی نداشته باشد؟ کدام تکه کد این امکان را فراهم میسازد؟

```
public boolean equals(Object obj) {
    if (obj instanceof Order order) {
        return id == order.id;
    }
    return false;
}
```

پاسخ:

خیر، انجام این کار ممکن نیست. به این علت که در این کد تا statement داریم:

1. return id == order.id;
2. return false;

استیتمنت اول زمانی اجرا می‌شود که شرط if درست باشد، پس برای پوشش این استیتمنت لازم است که branch اول if در پوشش دهیم. استیتمنت دوم هم، زمانی اجرا می‌شود که وارد branch دوم if شویم (شرط آن false باشد). پس برای پوشش صددرصدی استیتمنت‌ها مجبوریم تمام برنچ‌ها را پوشش بدهیم.

اگر کد را به صورت زیر تغییر دهیم چطور؟ می توان آزمایه ای که در قسمت اول آمده را ایجاد کرد؟

```
public boolean equals(Object obj) {
    var result = false;
    if (obj instanceof Order order) {
        result = id == order.id;
    }
    return result;
}
```

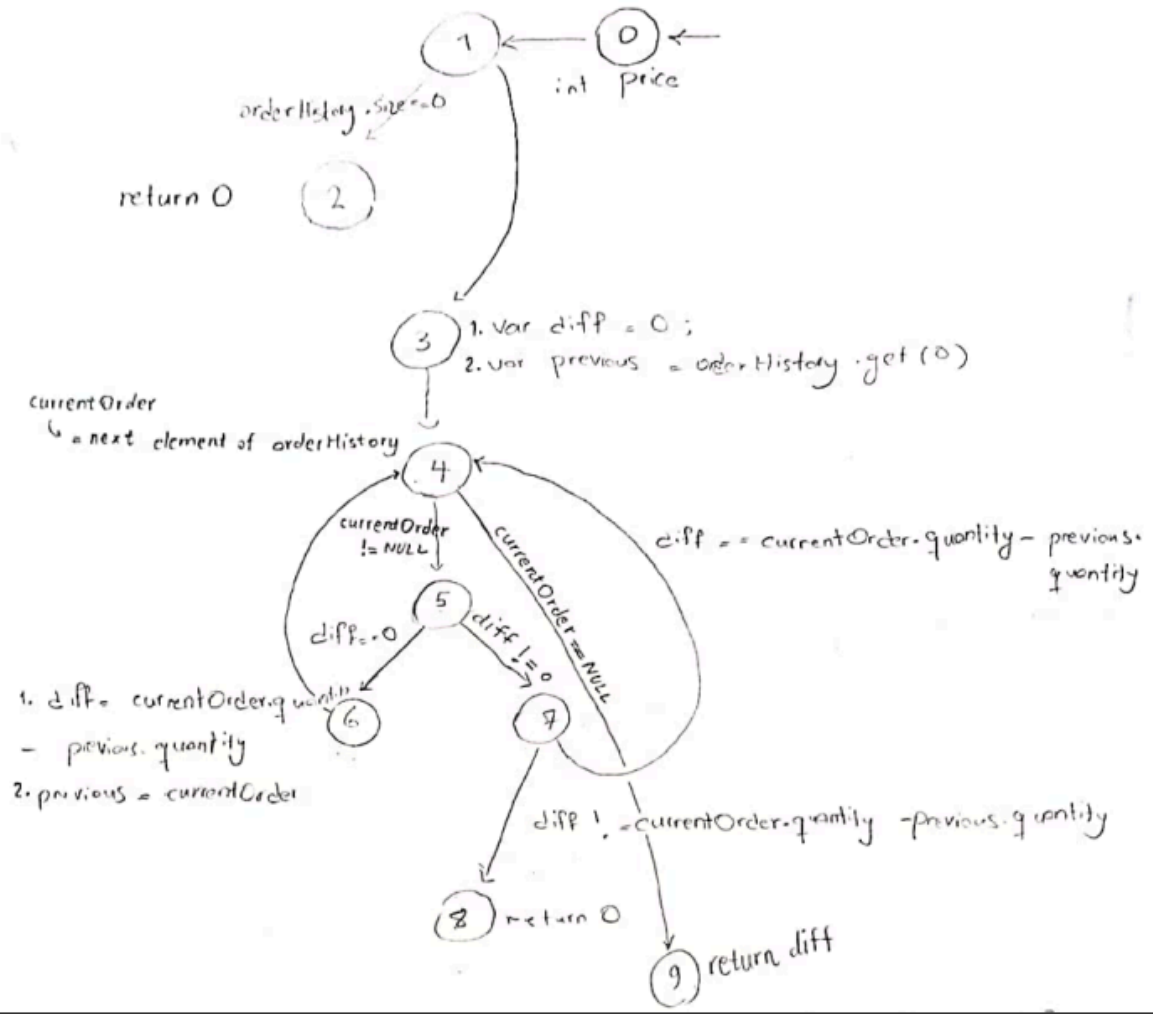
بله. کافی است در یک تست به عنوان ورودی یک آبجکت Order به این تابع بدهیم. در اجرای این تست هر سه استیتمنت اجرا می‌شوند (پوشش صددرصدی استیتمنت‌ها) در حالی که یکی از برنچ‌ها بررسی نشده (این که وارد if نشود).

سوال دوم

برای قطعه کد زیر CFG را رسم کنید. سپس تمامی prime path و du path ها را لیست کنید.

```
int getQuantityPatternByPrice(int price) {
    if (orderHistory.size() == 0) {
        return 0;
    }

    var diff = 0;
    var previous = orderHistory.get(0);

    for (Order currentOrder : orderHistory) {

        if (diff == 0) {
            diff = currentOrder.quantity - previous.quantity;
            previous = currentOrder;
        } else if (diff != currentOrder.quantity - previous.quantity) {
            return 0;
        }
    }

    return diff;
}
```

پاسخ:



---

prime path: [1,2], [1,3,4,5,6], [1,3,4,5,7,8], [1,3,4,7]

[4,5,6,4], [4,5,7,4], [5,6,4,5], [5,7,4,5], [6,4,5,6], [7,4,5,7]

[5,7,4,7], [5,6,4,7], [6,4,5,7,8], [7,4,5,6]

du-paths for price: — (no use)

du-paths for diff: [3,4,5,6], [6,4,5,6], [3,4,7]

du-paths for previous: [3,4,5,6], [6,4,5,6]

du-paths for currentOrder: [4,5,6]

du-paths for orderHistory: — (no def)

سوال سوم

آیا می توانید مجموعه آزمایه ای تولید کنید که تمامی du path ها را پوشش دهد، اما prime path ای داشته باشیم که پوشش داده نشده است؟

پاسخ:

بله. می‌توان این کار را کرد. مثلا فرض کنید کلاسی، فیلد پرایوتی به نام a دارد که در کانستراکتور دیفاین شده. حالا در متودی که می‌خواهیم آن را تست کنیم، چنین استیتمنتی داریم (در ////، فقط یک سری متغیر دیفاین شده‌اند و برنچی ایجاد نمی‌شود. در واقع این if اولین برنچ را تولید می‌کند):

```
////
if(a == 2) {
  return 0;
}
```

- دیفاین a در این متود نیامده.

پرایم‌پثی را در نظر بگیرید که از ابتدای متود شروع شده، وارد ایف می‌شود و با استیتمنت return 0 پایان می‌یابد. با پوشش دادن duها نمی‌توان این پث را پوشش داد. به این علت که شامل هیچ duای نیست. یعنی نمی‌توان duای را در نظر گرفت که در ادامه‌ی ران شدن به این return برسد. (تمام یوزها در برنچ دیگر if قرار دارند)

سوال چهارم

همانطور که می دانید معیار prime path تضمین می کند که du path ها نیز پوشش داده میشوند. با توجه به اینکه پیدا کردن du path ها کار پر هزینه تری است، چرا به سراغ این معیار می رویم؟

پاسخ:

در بسیاری موارد این حالت رخ می دهد که گراف data flow برنامه ما بسیار پیچیده می شود ، در این حالت پیدا کردن و پوشش دادن prime path ها بسیار دشوار می شود ، در این حالت استفاده از definition usage path ها که کوتاه تر هستند ، میتواند کارا باشد به خصوص از این منظر که du path ها معمولا باعث می شوند test suite ها کوچکتر شوند و در بررسی جریان داده در کد مفید تر واقع شوند. (هم چنین اگر چه اگر prime path ها پوشش داده می شوند ، در این حالت تمرکز زیادی روی کنترل جریان داده است در حالی که

یکی از نکات مهمی هم که باید در این روند به آن توجه کنیم مسائل مربوط به جزئیات متغیر هاست یعنی باید از مدیریت داده ها و متغیر ها اطمینان حاصل کنیم . مثلا یکی از نکاتی که باید به آن توجه کنیم این است که آیا متغیر ها قبل از استفاده تعریف شده اند یا خیر . با استفاده از معیار هایی مثل all du path coverage می توانیم از تحقق همچین مسائلی اطمینان حاصل کنیم.)