

سوال اول

Fault: یک نقص استاتیک (عنوز به الفعل نشده) در نرم افزار
Error: یک حالت (state) درونی نادرست که نمود ظاهری عیب است.
Failure: با propagate شدن error ایجاد می شود و یک رفتار بیرونی نادرست نسبت به الزامات یا سایر توضیحات رفتار مورد انتظار است.

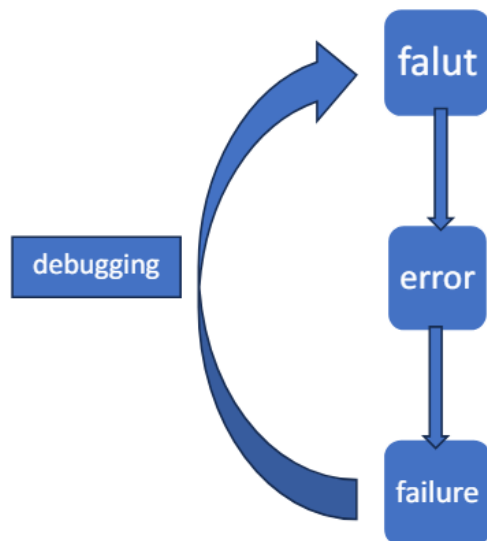
مثال: به عنوان مثال قطعه کد زیر می خواهیم تعداد اعداد فرد در ارایه را بشمارد:

```
odd_counter(int[] arr):  
    counter=0  
    For(int i=1; i<arr.length; i++):  
        If (arr[i]%2==1):  
            counter++  
    return counter;
```

- **Fault:** قسمت $i=1$ در حلقه for غلط است چون باید از 0 شروع شود.
- **Error:** اولین جایی که state یکی از متغیرهای برنامه اشتباه می شود زمانی است که به $i=1$ در خط سوم در حلقه for میرسیم.
- **Failure:** اگر در ارایه ورودی اولین عضو مجموعه عددی فرد باشد - به عنوان مثال $\{1,2,3\}$ - خروجی نهایی کد اشتباه می شود و به جای 2، عدد 1 را به عنوان خروجی می گیریم.

سوال دوم

Fault ها مشکلات یا نقض هایی در کد هستند، failure ها رویدادهای خارجی نادرست هستند یا به زبان دیگر خروجی های نادرست (غیر منطبق با خروجی مورد انتظار). آزمایش فرآیندی است برای ایجاد خرابی یا نشان دادن اینکه آنها با سرعت قابل قبولی رخ می دهند. در مقابل، اشکال زدایی یا debugging یک فرآیند تشخیصی است که در آن، با توجه به یک شکست، تلاش برای یافتن fault مربوطه انجام می شود. تسویر زیر گویای این روند است:



سوال سوم

کد ۱:

الف) `Max_val=arr[0]` ممکن است اصلاً آرایه خالی باشد و عضو اول نداشته باشد. اگر آرایه `null` باشد نیز `exception` برگردانده میشود.

ب) امکان ندارد چون بخش مربوط به `fault` در ابتدای برنامه است و در هر صورت اجرا میشود.

پ) هر آرایه‌ی غیر خالی و غیر `null` مثلاً `[1,2,3]`

ت) امکان ندارد. هیچ `state` ای نیست که نسبت به حالت درست برنامه متفاوت باشد و منجر به `failure` نشود.

ث) آرایه خالی منجر به خطا میشود. ارور `indexOutOfBounds` برگردانده میشود اما توقع داریم `null` یا `None` مثلاً برگردانده شود.

ج)

مانند کتاب و اسلاید فرض میکنیم در قطعه کد ها `null exception` هندل شده.

اگر `null` بودن هندل نشده باشد در ابتدای تابع یک `if` برای بررسی `null` بودن `arr` (در پایتون `None` محسوب میشود) قرار میدهیم که همان `If arr is None` است. سپس درون `if` یک عبارت `return None` قرار میدهیم.

کد ۲:

الف) خط `average = total/len(arr)` اگر آرایه خالی باشد تقسیم بر `۰` داریم و ارور میدهد در حالی که طبق کامنت باید `۰` برگرداند.

ب) امکان ندارد و در تمام حالات `fault` اجرا میشود (در بدنه اصلی برنامه است و قبل از آن حالتی برای `return` یا ارور نداریم)

پ) `arr=[1,2,3,4]` در واقع هر آرایه با طول بیشتر از `۰`.

ت) امکان ندارد.

ث) `[]` چون تقسیم بر `۰` رخ میدهد و ارور `division by zero` داریم.

ج) مانند کتاب و اسلاید فرض میکنیم در قطعه کد ها `null exception` هندل شده و صرفاً `۰` برگرداندن در صورت خالی بودن آرایه را تصحیح میکنیم

اگر `null` بودن هندل نشده باشد در ابتدای برنامه یک `if` برای بررسی `null` بودن `arr` (در پایتون `None` محسوب میشود) قرار میدهیم که همان `If arr is None` است. سپس درون `if` یک عبارت `return None` قرار میدهیم.

کد ۴:

این کد هیچ مشکلی ندارد!