مرحله‌ی اول

مینی پروژه - آزمایشگاه

طراحان : ملیکا شیریان - محمد کاظم هرندی

هدف پروژه:

در این پروژه به کمک مفاهیم واسط‌ها ، قصد داریم شبیه ساز آزمایشگاه خون را ایجاد کنیم. در این برنامه باید افراد بتوانند آزمایش مورد نظر خود را انجام دهند و بتوانند نتیجه‌ی آن را در برگه‌ی پزشکی مشاهده کنند همچنین آزمایشگاه باید بتواند برروی نمونه خون های افراد آمار گیری انجام دهد.

درصورت عدم استفاده از واسط‌ها و کلاس ابسترکت ، iteration, comparable, clonable ،نمره ای به شما تعلق نخواهد گرفت.

شرح پروژه:

موجودیت ها:

- **فرد آزمایش دهنده:**
  - کدملی
  - لیست آزمایش ها
  - بیماری زمینه ای
  - سن
- **آزمایشگاه :**
  - اسم آزمایشگاه
  - لیست آزمایش های انجام شده
  - لیست افراد آزمایش دهنده

- لیست آزمایش های آمار گیری

- **آزمایش(جواب و روند):**
  - اطلاعات فرد آزمایش دهنده
  - تاییده‌ی آزمایش توسط فرد
  - کد رهگیری(هر آزمایش مختص به خودش دارد)
  - و ویژگی های دیگر که در تکه کد زیر نمایش داده شده است :

```
1
2       private int CBC;
3       private int BMP;
4       private int bloodPressure;
5
```

- **سلسله مراتب ارثبری آزمایش ها:**
  - آزمایش گروه خونی
    - تاییدیه توسط آزمایشگاه
  - آزمایش ایدز
    - تاییدیه توسط آزمایشگاه
    - بیماری قلبی(رشته)
  - آزمایش تیروئید
    - مقدار عددی فشار قبل
  - آزمایش کم خونی

توجه : این آزمایشگاه قادر به انجام چهار نوع آزمایش می باشد که هر کدام از آزمایش ها علاوه بر مقادیر اصلی آزمایش، شامل مقادیر به خصوص خود می باشند.

نکات پیاده سازی

- کد ملی هر شخص باید مختص همان فرد باشد.
- ما فقط و تنها فقط یک آزمایشگاه داریم در نتیجه باید بتوان فقط یک نمونه‌ی آزمایشگاه ساخت.
- تحویل آزمایش ها فقط با داشتن کد ملی انجام می شود.
- بستر پیاده سازی کد شما در گیت هاب می باشد.

نکات تکمیلی

- برنامه شما نباید در طول اجرا از برنامه خارج شود.
- رعایت نکات کدنویسی تمیز و نام گذاري ها داراي بارم مستقیم بوده و درصورت عدم رعایت از یک سوم تا نیمی از نمره کسر خواهد شد.
- هر تابعی که پیاده سازي می کنید باید یک کار مشخص انجام دهد ، یک تابع نباید چند کار همزمان انجام دهد و نباید بیش از حد طولانی باشد.
- توجه شود که تمامی توابع هر سه مرحله باید در مین تست شود و موقع ارائه‌ی کد به طور صحیح کار کنند.

در این مرحله شما باید کلاس ها را پیاده سازی کنید و کلاس آن‌را ارسال کنید.

مرحله‌ی دوم

توابع اولیه مورد نیاز پروژه:

در این قسمت می‌خواهیم متود های مورد نیاز هر یک از کلاس‌هارا پیاده سازی کنیم.قبل از انجام مراحل ، ابتدا به نکته‌ی زیر توجه کنید :

توابع زیر را باید طوری دسته بندی کنید ، که بعضی از آن ها در متد کلاس پدر برای ارث بری قرار گیرند ، و باقی توابع باید در واسط مرد نظر خود دسته بندی شوند . برای اینکه راحت تر بتوانید دسته بندی درست را انجام بدهید به کاربرد کلاس های abstract و interface ها مراجعه کنید .

شرح توابع:

1.**تابع runOperation** : برای تمامی آزمایش های متمایز، این متود پیاده‌سازی می‌شود که پیاده سازی آن برای هریک از آزمایش ها به صورت زیر است:

- **آزمایش تیروئید**: در این آزمایش ابتدا یک عدد رندوم بین سه تا هفت تولید کرده و سپس مقدار فاکتوریل ضربدر ده آن را در `CBC` و اختلاف `CBC` با عدد رندوم را در `BMP` قرار می‌دهیم و برای فشار خون، حاصل تقسیم `CBC` بر `BMP` را قرار می‌دهیم. سپس عددر رندوم را معادل فشار قلب در نظر می گیریم.
- **آزمایش کم خونی**: در این آزمایش ابتدا یک عدد رندوم بین یک تا پنجاه تولید کرده و سپس مقدار جذر آن را در `CBC` و حاصل تقسیم CBC با عدد دو را در `BMP` قرار می‌دهیم و برای فشار خون، حاصل تقسیم `CBC` بر `BMP` را قرار می‌دهیم.
- **آزمایش ایدز** :برای `CBC` شما باید عدد رندومی ایجاد بکنید که تقریبا 80 درصد مواقع عددی بین 20 تا 180 باشد. برای `BMP` نیز تقریبا 80 درصد مواقع عددی بین 35 تا 340 به ما بدهد همچنین برای blood pressure در 80 درصد مواقع بین 4 تا 150 به ما بدهد. برای بیماری قلبی آرایه ای شامل :

```
HEALTHY ,CORONARY ,STROKE ,PERIPHERALARTERIAL ,AORTIC ,NOT ,ERROR ,FINDFAILD
```

داشته باشید سپس یک عدد رندوم بین رنج اندیس های آرایه بسازید و مقدار داخل اندیس را بخوانید و در بیماری قلبی قرار دهید.

- **آزمایش گروه خونی** :برای CBC شما باید عدد رندومی ایجاد بکنید که تقریبا 80 درصد مواقع عددی بین 20 تا 160 باشد. برای BMP نیز تقریبا 80 درصد مواقع عددی بین 35 تا 210 به ما بدهد همچنین برای blood pressure در 80 درصد مواقع بین 4 تا 100 به ما بدهد.

2.**تابع sendResult** : در این تابع یک کپی از نتیجه‌ی آزمایش برای فرد آزمایش دهنده ارسال می شود. همچنین نسخه‌ی اصلی آزمایش، در آزمایشگاه ذخیره می‌شود.

3.**تابع checking و wrongAnswer** : این تابع فقط مختص به آزمایش های ایدز و گروه خونی است. برای تاییدیه ی نهایی عملیات آزمایش پس از اجرای آزمایش تابع checking صدا زده می‌شود که در تابع checking مقادیر بدست آمده در متود runOperation را با مقادیر استاندارد مقایسه می‌کند و اگر هر یک از آنها با مقادیر رنج متوسط آن در تداخل باشد، تابع wrongAnswer صدا زده می‌شود که اگر تعداد تکرار آزمایش کمتر از سه بار برای ایدز و اگر کمتر از دوبار برای گروه خونی بود آزمایش را دوباره انجام دهد. اگر در زمان checking تمامی مقادیر درست بودند ویژگی تاییدیه را true کرده و اگر بعد از چک کردن مکرر همچنان مقادیر اشتباه بودند ویژگی تاییده را برابر با false قرار داده و آن فرد باید دوباره آزمایش بدهد.

مقادیر بدست آماده را با مقادیر دو enum زیر چک کنید:

```
enum  AIDSRange{
    CBC(20 , 180),
    BLOOD_PRESSURE(4 , 150),
    BMP(35 , 340);
    private enum HeartDisease{
            HEALTHY,
            CORONARY,
            STROKE,
            PERIPHERAL_ARTERIAL,
            AORTIC ;
    }
    private int range_one ;
    private int range_two;

```

```
    AIDSRange(int range_one, int range_two) {
            this.range_one = range_one;
            this.range_two = range_two;
    }
    public int getRange_one() {
    return range_one;
    }
    public int getRange_two() {
    return range_two;
    }
}
```

```
enum  BloodTypeRange{
    CBC(20 , 160),
    BLOOD_PRESSURE(4 , 100),
    BMP(35 , 210);

    private int range_one ;
    private int range_two;

    BloodTypeRange(int range_one, int range_two) {
        this.range_one = range_one;
        this.range_two = range_two;
    }
    public int getRange_one() {
        return range_one;
    }
    public int getRange_two() {
            return range_two;
    }

}
```

4.**تابع makingPrivate و gettingTheRequests**:بعضی از آزمایش ها ویژگی هایی دارند که باید از عموم افراد جامعه محفوظ بماند. به کمک تابع makingPrivate به دلخواه دو مورد از ویژگی های آن آزمایش را خصوصی کنید و پاسخ جدید آزمایش را به فرد نمایش دهید. توجه داشته باشید که پاسخ اصلی آزمایش

باید در آزمایشگاه نگهداری شود.

سه آزمایش تیروئید، کم خونی و ایدز از این ویژگی برخوردارند ولی گاهی اوقات، آزمایش دهنده اطلاعات کامل آزمایش خودرا می‌خواهد پس آزمایش دهنده می‌تواند یک درخواست به آزمایشگاه برای نسخه ی اصلی خود ارسال کند.(توجه کنید اگر قبلا این درخواست ارسال شده است دوباره نمی تواند درخواست دهد.)

پس در آزمایشگاه یک لیستی از درخواست ها ذخیره سازی کنید و یک تابع برای رسیدگی به درخواست ها ایجاد کنید و به کمک تابع gettingTheRequests درخواست را تایید یا رد کنید. برای آزمایش تیروئید اگر فرد دارای بیماری زمینه ای باشد اجازه ی دیدن اطلاعات اصلی به او را ندهید همچنین برای آزمایش کم خونی اگر فرد دارای سن کمتر از 18 سال باشد، برای آزمایش ایدز هر دو مورد باید بررسی شود. اگر صلاحیت تایید شد نتیجه ی آزمایش باید با اطلاعات کامل جایگزین نتیجه ی قبلی فرد آزمایش دهنده شود.

بعد از اتمام این مرحله کد خودرا بارگزاری گنید.

مرحله‌ی سوم

آمارگیری

در این مرحله بخش آمار گیری آزمایشگاه پیاده سازی می‌شود:

در بخش آمار گیری پس از به اتمام رسیدن اجرای عملیات هر یک از آزمایش ها و تایید نهایی آن ها ،آن هارا در لیست آمار گیری ذخیره می‌کنیم. توجه داشته باشید برای ذخیره سازی ابتدا یک نسخه ی کپی از نتیجه ی اصلی گرفته و تمامی اطلاعات شخص آزمایش دهنده را از آن حذف کرده و در لیست مورد نظر ذخیره کنید.

آزمایشگاه یک تابع برای آمار گیری بر روی تمام نمونه ها را دارد. با صدا زدن این تابع،برای دسترسی به لیست آزمایش های آمارگیری باید نمونه ی سینگلتون آزمایشگاهتان را صدا بزنید و روی آن پیمایش انجام دهید (پیمایش باید روی آرایه ی آزمایش های آمارگیری باشد.) تا آنهارا با یکدیگر مقایسه کنید و نتیجه را باز گردانید.

شیوه ی انجام مقایسه:

شما باید به کمک BMP ، CBC و blood pressure هر یک از آزمایش هارا با یکدیگر مقایسه بکنید:

- ابتدا BMP دو آزمایش را با یکدیگر مقایسه کنید. اگر مقادیر مساوی نبودند (یعنی کوچکتر یا بزرگتر بود) نتیجه را بازگردانید.
- در صورت مساوی بودن مقادیر BMP باید CBC دو آزمایش را با یکدیگر مقایسه کنید. اگر مقادیر مساوی نبودند (یعنی کوچکتر یا بزرگتر بود) نتیجه را بازگردانید.
- در صورت مساوی بودن مقادیر CBC باید blood pressure دو آزمایش را با یکدیگر مقایسه کنید و نتیجه را بازگردانید.

برای نشان دادن نتیجه ی نهایی آمارگیری، باید نتیجه را در یک آرایه ی دو بعدی ذخیره کنید و به همان شکل ماتریسی نمایش دهید. شیوه ی نمایش به این صورت است که در هر یک از سطر ها مقادیری که با یکدیگر برابر بودند قرار دارند و در هر یک از ستون ها مقدار ها به صورت صعودی نوشته می شود.

برای درک بهتر شکل خروجی ارایه‌ی دوبعدی ، به مثال زیر توجه کنید :

```
1 1 4 5
2 2 6
3 3
4 7 8
```

که در مثال بالا ، در هر یک سطر ها ، مقادیر مشابه آزمایشی هستند و معادل یک شبیه ساز برای کشیدن نمودار است.برای مثال در این شکل، نتیجه ازمایش ها با کد های رهگیری ۱و۴و۵ با یکدیگر برابر بوده اند و نتیجه‌ی آزمایش های ۲و۶ در حالت مطلوب تری از سطر بالایی خود یعنی آزمایش های ۱و۴و۵ داشته است . **توجه داشته باشید ، برای ساخت این آرایه ، از ارایه های منظم نمیتوانید استفاده کنید.**

و درنهایت :

شما باید در قسمت مین کد خود تمامی امور مربوط به این آزمایشگاه اعم از انجام تمام آزمایش ها بر روی حداقل 5 نفر را مورد بررسی قرار دهید و در مرحله ی آخر آمارگیری بر روی آزمایش ها انجام شود.