



تمرین کامپیوتری  
شماره ۴  
آشنایی با پوشش منطقی،  
Input Space Partitioning  
و تست API

آزمون نرم افزار - پاییز ۱۴۰۱

مهلت تحویل:

دانشکده مهندسی برق و کامپیوتر

طراح تمرین: **سینا کمالی**،  
**سینا نگارنده**

پایان روز ۲۸ آذر

استاد: دکتر احسان خامس پناه

---

هدف از این پروژه آشنایی با مفاهیم پوشش منطقی، Input space partitioning، و همینطور API Testing می باشد.

در بخش اول، به سوالی در رابطه با پوشش منطقی جواب می دهید. پوشش منطقی به ما کمک می کند که تمام حالاتی که شامل متغیری با مقدار True می شوند را بررسی کنیم. اهمیت این موضوع در اینجاست که با ما کمک می کند که مطمئن شویم در بخش های مربوط به منطق کد مانند شرطها اتفاق های ناخواسته ای رخ نمی دهند.

در ادامه، به پاسخ دادن سوالاتی در رابطه با Input Space Partitioning می پردازید، که بخش بسیار مهمی از تست ها است. در هر برنامه ای ورودی های ما می توانند در بازه هایی مشخص باشند که مقدارهای مختلفی از این مقادیر می توانند تاثیراتی بر روی یکدیگر بگذارند. با انجام ISP ما مطمئن می شویم که تمامی حالات ممکن از ورودی ها در برنامه بدون مشکل اجرا می شوند، به خصوص مقادیر مرزی! در نهایت در بخش عملی به طراحی آزمون هایی در حیطه API Testing می پردازید.

**بخش اول (گزارش کار):** برای predicate عنوان شده به سوال زیر پاسخ دهید.

$$p = (\neg a \wedge \neg b) \vee (a \wedge \neg c) \vee (\neg a \wedge c)$$

- clause ها را مشخص کنید.
- جدول درستی را بنویسید.
- تمام جفت ردیف‌هایی را از جدول درستی که GACC را برای هر clause برآورده می‌کند فهرست کنید.
- تمام جفت ردیف‌هایی را از جدول درستی که CACC را برای هر clause برآورده می‌کند فهرست کنید.
- تمام جفت ردیف‌هایی را از جدول درستی که RACC را برای هر clause برآورده می‌کند فهرست کنید.
- تمام tuple ها از ردیف‌های جدول درستی که GICC را برای هر clause برآورده می‌کند فهرست کنید.
- تمام tuple ها از ردیف‌های جدول درستی که RICC را برای هر clause برآورده می‌کند فهرست کنید.

**بخش دوم (گزارش کار):** کلاس Queue را در نظر بگیرید. این Queue بیش از ظرفیت capacity عضوی دریافت نمی‌کند. size نیز تعداد عناصر داخل این Queue است. برای این دو مشخصه، بلوک‌ها را با مقادیرشان و همچنین بلوک پایه را مشخص کنید و سپس پوشش Base Choice را پیاده‌سازی کنید. (برای نوشتن کد تست‌ها از شبه کد استفاده کنید. در هر تست کیس، با توجه به مسئله و متدهای داده شده، assertion مناسب را انجام دهید).

**نکته:** حداقل و حداکثر مقادیر مجاز برای capacity و size را صفر و Integer.MAX\_VALUE در نظر بگیرید.

```
class Queue {  
    private int capacity, size = 0;  
  
    public Queue(int capacity) {}  
    public void enqueue(Object value) {}  
    public Object dequeue() {}  
}
```

بخش سوم (قسمت عملی): برای کلاس CourseController با استفاده از @WebMvcTest یا @SpringBootTest تست API بنویسید. توجه کنید که نحوه تست نباید به شکل فراخوانی مستقیم متدهای کنترلر باشد و باید در به صورت وب تست پیاده سازی شوند.

در نهایت در گزارش کار خود مشخص کنید که تفاوت بین دو انوتیشن @WebMvcTest و @SpringBootTest چیست؟

### نکات تکمیلی

- پروژه در قالب گروه‌های دو نفره انجام شود.
- برای انجام پیاده‌سازی‌های خود مخزن را fork کرده و تغییرات را در مخزن شخصی خود انجام دهید.
- برای این پروژه و پروژه‌های بعدی یک شاخه (branch) جدید با نام ca#projectNumber ایجاد کرده و تغییرات خود را روی این شاخه اعمال کنید.
- گزارش کار در قالب یک فایل pdf با نامی شامل شماره دانشجویی اعضای گروه باشد.
- برای تحویل کفایت یکی از اعضای گروه گزارش پروژه که شامل آدرس مخزن شخصی و شناسه آخرین کامیت می‌باشد را در صفحه درس بارگذاری نماید.
- کاربر UTTest-Fall01 را به مخزن خود اضافه کنید.
- برای ارتباط با دستیاران آموزشی می‌توانید از ایمیل‌های [sininoir@gmail.com](mailto:sininoir@gmail.com) و [sina.negarandeh@ut.ac.ir](mailto:sina.negarandeh@ut.ac.ir) استفاده کنید.