

سوال ۱

برای هر یک از سیستم‌های زیر حداقل دو نمونه برای موجودیت‌ها Entities ویژگی‌ها Attributes فعالیت‌ها Activities رویدادها Events و متغیرهای حالت State Variables نام ببرید. در بخش رویدادها، یک رویداد داخلی Endogenous Event و یک رویداد خارجی Exogenous Event باید ذکر شود.

جواب سوال ۱

پیام‌رسان

- موجودیت‌ها (Entities): کاربران، پیام‌ها
- ویژگی‌ها (Attributes): نام کاربری، زمان ارسال پیام
- فعالیت‌ها (Activities): ارسال پیام، خواندن پیام
- رویدادها (Events):
 - رویداد داخلی (Endogenous Event): دریافت تاییدیه خواندن پیام
 - رویداد خارجی (Exogenous Event): دریافت پیام جدید از یک کاربر دیگر
- متغیرهای حالت State Variables: تعداد پیام‌های خوانده نشده، وضعیت آنلاین یا آفلاین بودن کاربر

بانک

- موجودیت‌ها (Entities): حساب، مشتری
- ویژگی‌ها (Attributes):
 - حساب: شماره حساب، موجودی
 - مشتری: نام، شناسه مشتری
- فعالیت‌ها (Activities): واریز و برداشت پول، انتقال وجه، درخواست وام
- رویدادها (Events):
 - رویداد داخلی (Endogenous Event): تغییر نرخ بهره بانکی
 - رویداد خارجی (Exogenous Event): ورود یک چک جدید برای وصول
- متغیرهای حالت (State Variables): تعداد کل حساب‌ها، مجموع موجودی‌های بانک

سایت رزرو هتل

- موجودیت‌ها (Entities) : مشتریان، اتاق‌های هتل
- ویژگی‌ها (Attributes) : تاریخ ورود و خروج، قیمت اتاق
- فعالیت‌ها (Activities) : جستجو برای اتاق، رزرو اتاق
- رویدادها (Events) :
 - رویداد داخلی (Endogenous Event) : انقضای زمان رزرو اتاق
 - رویداد خارجی (Exogenous Event) : تغییر قیمت اتاق به دلیل تغییرات فصلی
- متغیرهای حالت (State Variables) : تعداد اتاق‌های موجود، تعداد رزروهای انجام‌شده

کارخانه‌ی ساخت مبل

- موجودیت‌ها (Entities) : خط تولید، مبل‌ها
 - ویژگی‌ها (Attributes) : مدل مبل، زمان تولید
 - فعالیت‌ها (Activities) : برش چوب، روکش‌کاری
 - رویدادها (Events) :
 - رویداد داخلی (Endogenous Event) : خرابی ماشین آلات
 - رویداد خارجی (Exogenous Event) : تغییر در تقاضای بازار
 - متغیرهای حالت (State Variables) : تعداد مبل‌های در دست ساخت، میزان مواد اولیه موجود
-

سوال ۲

الف

درباره صحیح یا غلط بودن جملات زیر استدلال کنید (در صورت غلط بودن برای آن مثال نقض بیاورید)

- ۱- مدل‌های شبیه‌سازی گسسته را فقط برای مدل کردن سیستم‌های گسسته می‌توان استفاده کرد.
- ۲- مدل‌های شبیه‌سازی پیوسته را فقط برای مدل کردن سیستم‌های پیوسته می‌توان استفاده کرد.

ب

طبق اسلایدها می‌دانیم هنگامی که محیط تاثیر اندکی روی سیستم دارد، می‌توانیم به سه صورت آن را در شبیه‌سازی استفاده کنیم. این روش‌ها را شرح داده و برای هر یک مثالی که استفاده از آن روش مطلوب است را بیان کنید.

جواب سوال ۲

الف

- ۱- مدل‌های شبیه‌سازی گسسته را فقط برای مدل کردن سیستم‌های گسسته می‌توان استفاده کرد.
 - پاسخ: این جمله غلط است. گاهی اوقات برای ساده‌سازی مدل‌سازی یک سیستم پیوسته، می‌توانیم از مدل‌های گسسته استفاده کنیم. مثلاً در مدل‌سازی جریان ترافیک که به طور واقعی پیوسته است، می‌توانیم از مدل‌های گسسته برای شبیه‌سازی حرکت خودروها در بازه‌های زمانی معین استفاده کنیم.
- ۲- مدل‌های شبیه‌سازی پیوسته را فقط برای مدل کردن سیستم‌های پیوسته می‌توان استفاده کرد.
 - پاسخ: این جمله نیز غلط است. در برخی موارد، برای به دست آوردن نتایج دقیق‌تر و همچنین برای مدل‌سازی تغییرات گسسته با دقت بالا، می‌توان از مدل‌های شبیه‌سازی پیوسته استفاده کرد. به عنوان مثال، می‌توانیم سیستم صف گسسته را با استفاده از مدل پیوسته شبیه‌سازی کنیم تا نحوه تغییر تراکم صف در طول زمان را ببینیم.

ب

- هنگامی که محیط تاثیر اندکی روی سیستم دارد، سه رویکرد برای مدل‌سازی و شبیه‌سازی وجود دارد:
- الف) در نظر گرفتن موارد خارجی به عنوان ورودی‌ها: در این روش، فرض بر این است که موارد خارجی می‌توانند به صورت ورودی‌های کنترل‌شده‌ای در مدل گنجانده شوند که تاثیر آنها قابل پیش‌بینی و مدیریت است.
مثال: در شبیه‌سازی پخش یک بیماری، تعداد افراد وارد شونده به یک شهر می‌تواند به عنوان ورودی مدل در نظر گرفته شود که بر میزان گسترش بیماری تاثیر می‌گذارد.
- ب) گسترش تعریف سیستم برای شامل کردن عوامل خارجی: این رویکرد شامل توسعه حدود سیستم برای گنجاندن عوامل خارجی به عنوان بخشی از خود سیستم است.
مثال: در مدل‌سازی یک کسب‌وکار، عوامل محیطی مانند رقابت و تقاضای بازار می‌توانند به عنوان بخشی از سیستم کسب‌وکار تعریف شوند تا تاثیرات آنها به صورت دقیق‌تری مورد بررسی قرار گیرد.
- ج) نادیده گرفتن عوامل خارجی: در صورتی که تاثیر محیط بر سیستم بسیار ناچیز باشد، می‌توان آنها را کاملاً نادیده گرفت.
مثال: برای مدل‌سازی فرایند داخلی یک دستگاه تولیدی که در محیط کنترل‌شده قرار دارد، ممکن است تاثیرات محیطی نظیر دما و رطوبت خارجی قابل صرف نظر باشند.

سوال ۳

در مورد فلوچارت موجود در اسلایدهای ۲۸ و ۲۹ سری اول، به سوالات زیر پاسخ دهید:

الف) تفاوت دو مرحله‌ی *Verification* و *Validation* چیست؟

ب) در مرحله‌ی ۱۰، چه زمانی به تعداد اجرای بیشتری نیاز داریم؟ یک سناریو را ذکر کنید که در آن نیاز به تعداد اجراهای بیشتری داشته باشیم.

ج) آیا *Conceptualization Model* اثری بر نوع داده‌هایی دارد که باید جمع‌آوری شود؟

د) پس از انجام شبیه‌سازی در مرحله‌ی ۱۰، چه زمانی به مرحله‌ی ۸ (*Design Experimental*) می‌رویم و چه زمانی به مرحله‌ی ۹ (*Analysis and Runs Production*) می‌رویم؟

جواب سوال ۳

۱ تفاوت **Validation** و **Verification**: **Verification** به فرآیندی گفته می‌شود که در آن بررسی می‌کنیم که آیا مدل به درستی ساخته شده است یا خیر، یعنی «آیا ما مدل را به درستی ساخته‌ایم؟». این مرحله بر کیفیت و صحت فنی مدل تمرکز دارد و شامل تست‌هایی برای اطمینان از برنامه‌نویسی صحیح و رفع اشکالات احتمالی است. در مقابل، **Validation** به سنجش اینکه آیا مدل ساخته شده واقعاً نماینده‌ی صحیحی از سیستم واقعی است می‌پردازد یعنی «آیا ما مدل را به درستی ساخته‌ایم؟». این مرحله شامل مقایسه نتایج مدل با داده‌های واقعی و تأیید صحت کلی مدل است.

۲ نیاز به تعداد اجرای بیشتر: در مرحله‌ی ۱۰، نیاز به تعداد اجرای بیشتر معمولاً زمانی پیش می‌آید که واریانس نتایج شبیه‌سازی بسیار بالا باشد و بخواهیم اطمینان حاصل کنیم که نتایج ما قابل اعتماد هستند. سناریو: اگر مدل شبیه‌سازی یک کسب‌وکار جدید است و می‌خواهیم اثر متغیرهای تصادفی مانند تقاضای مشتریان را در شرایط مختلف بررسی کنیم، ممکن است نیاز به اجرای متعدد شبیه‌سازی باشد تا بتوانیم یک تصویر دقیق از توزیع نتایج و ریسک‌های احتمالی به دست آوریم.

۳ اثر **Model Conceptualization** بر جمع‌آوری داده‌ها: قطعاً **Model Conceptualization** اثری بر نوع داده‌هایی دارد که باید جمع‌آوری شود. این مرحله شامل تعریف مسئله و ساختار کلی مدل است و بر اساس آن، ما می‌توانیم مشخص کنیم که چه نوع داده‌هایی لازم است تا مدل بتواند سیستم واقعی را به درستی نمایش دهد.

۴ مراحل بعدی پس از انجام شبیه‌سازی: پس از انجام شبیه‌سازی در مرحله‌ی ۱۰، اگر نتایج به دست آمده نیاز به بهبود داشته باشند یا اگر فرضیات مدل نیاز به تغییر داشته باشند، به مرحله‌ی ۸ (*Experimental Design*) باز می‌گردیم. در این مرحله، ممکن است نیاز به تنظیم دوباره‌ی طرح آزمایشات و تغییر پارامترها یا فرضیات باشد. اگر نتایج مطلوب باشند و نیاز به بررسی‌های بیشتری نباشد، به مرحله‌ی ۹ (*Analysis and Runs Production*) می‌رویم که در آن تعداد بیشتری از اجراها برای تأیید نتایج و تحلیل‌های نهایی انجام می‌شود.

سوال ۴

مرکز واکسن

در این سوال، یک مرکز واکسن را بررسی می‌کنیم. هر روز تعدادی نفر به مرکز مراجعه می‌کنند. بسته به موجودی واکسن، ممکن است نوبت واکسن مراجعین به روزهای بعد منتقل شود.

| Probability | time lead |
|-------------|-----------|
| ۰/۵ | ۱ |
| ۰/۴ | ۲ |
| ۰/۱ | ۳ |

این مرکز یک مسئول دارد که تنها هر ۴ روز یکبار انبار واکسن را بررسی می‌کند و واکسن سفارش می‌دهد. تعداد واکسن‌های سفارش داده شده از فرمول زیر پیروی می‌کند:

$$\text{ordered vaccines} = 6 - \#(\text{remaining vaccines}) + \#(\text{vaccine shortage})$$

که در آن، $\#(\text{vaccine shortage})$ تعداد کل مراجعینی است که واکسن خود را هنوز دریافت نکرده‌اند. مثال: اگر در انتهای روز چهارم ۳ واکسن در انبار باقی مانده باشد و کمبودی نداشته باشیم، ۳ واکسن سفارش داده می‌شود. و اگر تنها ۲ واکسن کمبود داشته باشیم و انبار خالی باشد، ۸ واکسن سفارش داده می‌شود. تعداد مراجعین (تقاضا) هر روز از جدول آبی پیروی می‌کند. برای بدست آوردن مقدار رندوم، از لیست ۱ مشابه اسلایدهای درس استفاده کنید.

هنگام سفارش واکسن، حداکثر تا ۳ روز طول می‌کشد تا واکسن‌ها وارد انبار شوند. زمان مورد نیاز برای افزایش ذخیره انبار از جدول قرمز پیروی می‌کند. برای اعداد رندوم، از لیست ۲ استفاده کنید.

شما باید سیستم را برای ۲۰ روز شبیه‌سازی کرده و جدول مربوطه را رسم کنید. فرض کنید در انتهای روز صفر، ۳ واکسن در انبار موجود بوده و کمبودی نداریم. همچنین این روز، روز بررسی انبار است. به عبارت دیگر، در پایان این روز، ۳ سفارش واکسن انجام می‌گیرد. جدول شما باید شامل ستون‌های زیر باشد:

- روز
- ذخیره انبار در ابتدای روز
- تعداد مراجعین روز
- ذخیره انبار در انتهای روز
- میزان کمبود در انتهای روز
- روز سفارش
- مقدار سفارش
- تعداد روزهای مانده تا افزایش ذخیره انبار

برای ستون‌هایی که میانگین معنادار دارد، آن را نمایش دهید. استفاده از دستورات اکسل در این مسئله مجاز نیست و جدول باید به صورت دستی پر شود.

List ۱ : ۶۸, ۳۲, ۸۳, ۶, ۸۲, ۲۱, ۱۲, ۳۲, ۵۹, ۴۸, ۵۸, ۱۲, ۱۸, ۴۸, ۲۲, ۵۷, ۱۸, ۸۹, ۸۴, ۵۱

List ۲ : ۷۶, ۳۰, ۹۶, ۳۷, ۴۸, ۷۰

| Probability | Demand |
|-------------|--------|
| ۰/۱ | ۰ |
| ۰/۲۵ | ۱ |
| ۰/۴۵ | ۲ |
| ۰/۲ | ۳ |

جواب سوال ۴

سوال ۵

| Probability | time lead | time lead | time lead | time lead | time lead |
|-------------|-----------|-----------|-----------|-----------|-----------|
| ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۱ |
| ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۱ |
| ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۱ |
| ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۰/۵ | ۱ |

جواب سوال ۵

سوال ۶

سوال عملی تمرین ۱، کد شبیه‌سازی...

جواب سوال ۶

کد ارائه شده برای شبیه‌سازی عملیات دو سرور در یک سیستم کوئو (صف) است. سیستم دارای دو سرور با نام‌های "Able" و "Baker" است که برای پذیرش و سرویس‌دهی به مشتریان مورد استفاده قرار می‌گیرند. این شبیه‌سازی شامل موارد زیر است:

- توزیع زمان بین ورودی‌ها و زمان سرویس: این داده‌ها از دو جدول توزیع احتمالاتی استخراج شده و در قالب دیکشنری‌هایی در کد قرار داده شده‌اند. زمان بین ورودی‌ها و زمان سرویس برای هر سرور مطابق با این توزیع‌های احتمالاتی تولید می‌شوند.

- تولید زمان بین ورودی‌ها: تابع `generate_interarrival_time` به صورت تصادفی زمان‌های بین ورود مشتریان را مطابق با توزیع احتمالاتی مربوطه تولید می‌کند.
- تولید زمان سرویس: تابع `generate_service_time` به صورت تصادفی زمان سرویس برای هر سرور را مطابق با توزیع احتمالاتی خودش تولید می‌کند.
- شبیه‌سازی: کد از طریق حلقه‌ای برای هر مشتری، بررسی می‌کند که آیا هر سرور در زمان ورود مشتری آزاد است یا خیر. اگر آزاد باشد، مشتری را سرویس می‌دهد و اگر نباشد، مشتری باید منتظر بماند.
- محاسبه زمان بیکاری و استفاده: برای هر سرور، کد زمان بیکاری و زمان استفاده را به صورت دوره‌ای محاسبه می‌کند. این امر از طریق ثبت زمان‌هایی که سرور بیکار است و زمان‌هایی که سرور مشغول سرویس‌دهی است، انجام می‌شود.
- ترسیم نمودار: کد برای هر سرور نمودارهایی از استفاده و زمان بیکاری را ترسیم و ذخیره می‌کند.
- میانگین استفاده و زمان بیکاری: برای هر سرور، میانگین زمان استفاده و زمان بیکاری را محاسبه و چاپ می‌کند.

تحلیل خروجی‌ها:

: Able

- **Average Utilization (میانگین استفاده):** $0/68$ یا 68% از زمان، سرور Able مشغول سرویس‌دهی به مشتریان است. این به این معنی است که سرور اکثر اوقات فعال است و نسبتاً بار کاری بالایی دارد.
- **Average Idle Time (میانگین زمان بیکاری):** $0/32$ یا 32% از زمان، سرور Able بیکار است. این نشان‌دهنده تعادل خوبی بین زمان بیکاری و استفاده است که از کارایی مناسب سرور حکایت دارد.

: Baker

- **Average Utilization:** $0/81$ یا 81% از زمان، سرور Baker در حال سرویس‌دهی است. این بدان معنی است که سرور Baker نسبت به سرور Able فعال‌تر است و بار کاری سنگین‌تری دارد.
- **Average Idle Time:** $0/19$ یا 19% از زمان، سرور Baker بیکار است. این سرور کمترین زمان بیکاری را دارد و ممکن است نشان‌دهنده نیاز به منابع بیشتر برای جلوگیری از بیش‌باری باشد.

با توجه به این خروجی‌ها، می‌توان نتیجه گرفت که سرور Baker نیاز به توجه بیشتری دارد تا از اضافه‌بار جلوگیری شود. ممکن است لازم باشد برنامه‌ریزی برای این سرور مجدداً انجام شود تا بتواند با بار کاری کنونی خود به خوبی کنار آید یا شاید نیاز به افزایش ظرفیت یا تعداد سرورها باشد.

Figure 1

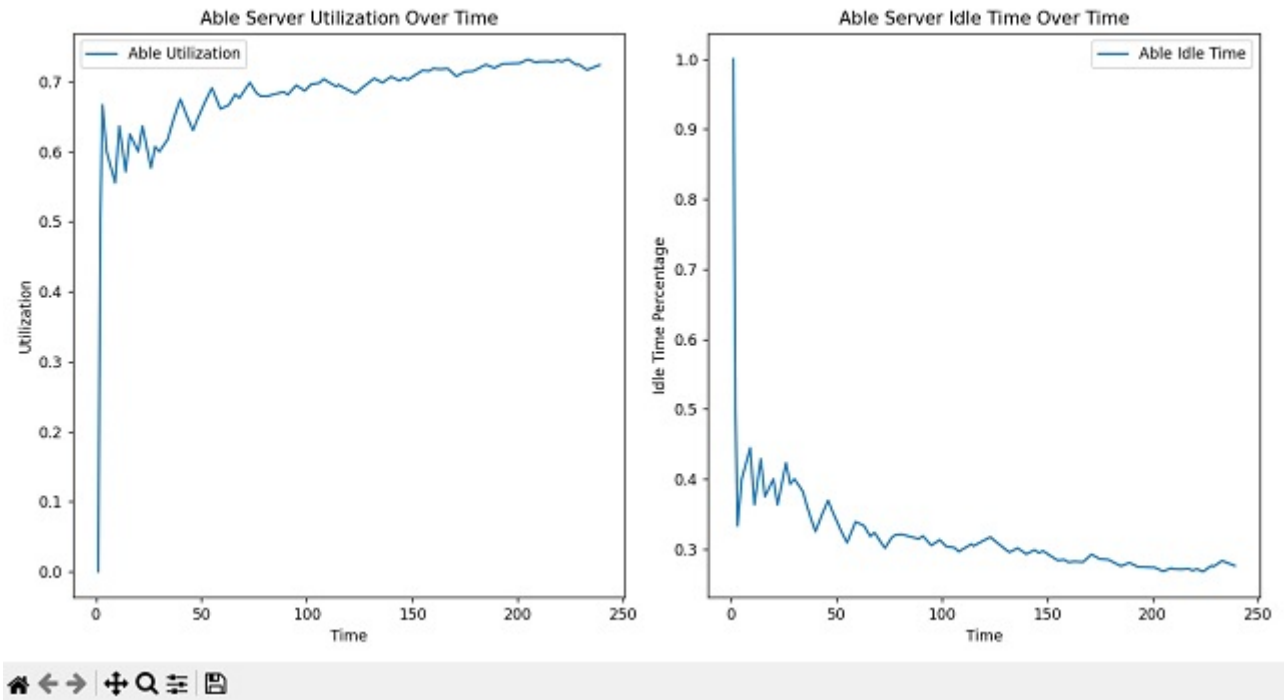


Figure 1

