



مقدمه

هدف از این تمرین آشنایی شما با مفاهیم اولیه طراحی شیء‌گرایی یک مسئله است. از آنجایی که استفاده از این مفاهیم در پیاده‌سازی سایر تمرین‌های درس لازم است، پیشنهاد می‌شود به این تمرین زمان کافی را اختصاص دهید.

مسافرت خسرو

خسرو قرار است بعد از ۲۰ سال دوستان دانشگاهی خود را ملاقات کند. هرکدام از این افراد در شهرهای مختلفی زندگی می‌کنند؛ در نتیجه تصمیم می‌گیرند در جایی که به همه آنها نزدیک است، یکدیگر را ملاقات کنند. خسرو برای این مسافرت، قصد دارد از قطارهای بین شهری استفاده کند اما به دلیل آشنا نبودن با این سیستم، از شما می‌خواهد که به او در شناخت این شبکه کمک کنید تا سفر راحتی داشته باشد.

شرح تمرین

در این تمرین، شما به شبیه‌سازی شبکه قطارهای بین شهری می‌پردازید. در این شبیه‌سازی یک ساعت اصلی داریم که نمادی از زمان سپری شده است. با استفاده از این زمان متوجه می‌شویم که در چه مرحله‌ای از شبیه‌سازی هستیم با استفاده از این کار قرار است محل و شرایط افراد و قطارها را در هر زمان به دست بیاوریم.

فرایند شبیه‌سازی

دوستان خسرو در این شبیه‌سازی ابتدا در ایستگاه مبدا خود قرار دارند و تا زمانی که به ایستگاه مقصد برسند، در هر مرحله یا در ایستگاهی منتظر رسیدن قطار هستند یا سوار قطار شده‌اند و منتظر رسیدن قطار به ایستگاه بعدی هستند. دوستان خسرو از قدیم به شکلات اعتیاد داشتند و باید در طول ۱ ساعت میزان مشخصی شکلات بخورند ولی برای این که کسی متوجه اعتیادشان به شکلات نشود در هر ایستگاه شکلات خرید می‌کنند. قطارها یا منتظر رفع خستگی راننده قطار هستند یا در مسیر به سمت مقصد در حال حرکت هستند.

- زمان سفر بین دو ایستگاه برای هر قطار مشخص شده است.
- هر شخص با توجه به میزان اعتیادش به شکلات در هر ساعت به تعداد مشخصی شکلات از ایستگاه می‌خرد و می‌خورد.
- قیمت شکلات در ایستگاه‌های مختلف متفاوت است.
- هر قطار بعد از وارد شدن به ایستگاه مقدار مشخصی می‌ایستد تا دوباره به سمت ایستگاه بعدی حرکت کند. این زمان میزان استراحت قطار است.
- در شروع شبیه‌سازی قطارها در وضعیت استراحت هستند و پس از گذشتن مدت زمان استراحت حرکت می‌کنند.
- هنگامی که قطاری به ایستگاهی می‌رسد در همان زمان مسافران آن قطار پیاده می‌شوند و وارد ایستگاه می‌شوند و در واحد زمان بعدی مسافران جدید سوار این قطار می‌شوند.
- مسافران ابتدا وارد ایستگاه می‌شوند و با گذر یک واحد زمانی در صورتی که قطاری به سمت مقصد بعدی آن‌ها منتظر باشد سوار قطار می‌شوند.
- مسافران پس از سوار قطار شدن دیگر شکلات نمی‌خرند.
- به ترتیب اضافه شدن افراد و قطارها، به آنها شناسه یکتا که برای هر کدام از یک شروع می‌شود اختصاص می‌دهیم.
- در سیستم قطارها برخورد ندارند و می‌شود فرض کرد از هم رد می‌شوند.
- اگر در ایستگاهی بیش از یک قطار منتظر مسافر برای حرکت به سمت ایستگاه یکسان بودند، مسافران سوار قطار با شناسه یکتای کمتر می‌شوند.
- شبیه‌ساز یک ساعت اصلی دارد که از صفر شروع می‌شود.
- اعتیاد هرکس به شکلات عددی حسابی است.

دستورات

اضافه کردن فرد

بعد از دستور `add_passenger`، به ترتیب میزان اعتیاد فرد به شکلات و ایستگاه‌هایی که مسیر حرکت آن فرد را مشخص می‌کند آمده است. در زمان صفر فرد در ایستگاه اول است.

فرمت خروجی	دستور ورودی
OK	<code>add_passenger <taffy addiction> <1st station name> <2nd station name> <3rd station name> ...</code>

مثال خروجی	مثال ورودی
OK	<code>add_passenger 2 zanzan tehran isfahan</code>

اضافه کردن ایستگاه

بعد از دستور add_station، نام ایستگاه و قیمت شکلات در آن ایستگاه آمده است.

فرمت خروجی	دستور ورودی
OK	add_station <name> <taffy price>

مثال خروجی	مثال ورودی
OK	add_station zanzan 20

اضافه کردن قطار

بعد از دستور add_train، به ترتیب نام ایستگاهی که از آنجا شروع می کند، نام ایستگاهی که به سمت آن می رود، مدت زمانی که طول میکشد یکبار مسافت بین دو ایستگاه را طی کند و مدت زمانی که در هر ایستگاه استراحت می کند آمده است.

فرمت خروجی	دستور ورودی
OK	add_train <starting station name> <second station name> <travel time> <cooldown time>

مثال خروجی	مثال ورودی
OK	add_train tehran tabriz 3 1

جلو بردن زمان شبیه سازی

با اجرای این دستور، زمان شبیه سازی به تعداد عدد داده شده که در آرگومان دوم دستور قرار دارد به جلو می رود.

فرمت خروجی	دستور ورودی
OK	advance_time <time_steps>

مثال خروجی	مثال دستور ورودی
OK	advance_time 1

اتمام کار تمام مراحل

در هر لحظه از اجرای برنامه، کاربر می تواند با وارد کردن این دستور، شبیه سازی را تا زمانی که همه ی افراد به مقصد برسند ادامه دهد.

خروجی	دستور ورودی
OK	finish

نمایش اطلاعات افراد

با وارد کردن دستور `show_passenger_info` و بعد از آن شناسه آن فرد باید سه خروجی در سه خط مختلف مانند نمونه ی زیر چاپ شود:

- در ابتدا جلوی عبارت `status` یکی از سه مقدار زیر می آید:
 - `on train`: رای زمانی که شخص سوار قطار است (توجه کنید که به محض رسیدن قطار شخص سوار می شود و منتظر می شود که قطار راه بیفتد)
 - `in station`: شخص در ایستگاه منتظر قطار است
 - `arrived`: شخص به مقصد رسیده است.
- در خط دوم جلوی `spent` مجموع مقدار هزینه ای که شخص تا الان با آن شکلات خریده آمده است.
- در خط سوم عبارت `current` با کلمه ی `station` یا `train` می آید که متناسب با مکان فعلی شخص است و بعد از آن شناسه آن قطار یا نام ایستگاه می آید. (اگر به مقصد رسیده باشد نام ایستگاه مقصد می آید)

فرمت خروجی	دستور ورودی
<code>status : <on train/in station/arrived></code> <code>spent : <amount></code> <code>current <station/train> <name/id></code>	<code>show_passenger_info <person id></code>

مثال خروجی	مثال ورودی
<code>status : on train</code> <code>spent : 40</code> <code>current train 3</code>	<code>show_passenger_info 5</code>

نمایش اطلاعات قطارها

با اجرای دستور `show_train_info`، می بایست اطلاعات زیر در مورد قطاری که شناسه آن در ادامه این دستور آمده است، مانند نمونه زیر چاپ شود:

- در خط اول جلوی `status`، وضعیت قطار چاپ می شود. قطار میتواند یکی از دو حالت زیر باشد:
 - `resting`: در حال استراحت است.
 - `moving`: در حال حرکت و بین دو ایستگاه هستیم.
- در خط دوم جلوی `estimated remaining time`، مدت زمانی که طول می کشد تا قطار به ایستگاه بعدی و اگر قطار در استراحت است باید زمان باقی مانده از استراحت در نظر گرفته شود.
- در خط سوم جلوی `next station`، نام ایستگاه بعدی که قطار به سمت آن حرکت می کند چاپ می شود.
- در خط چهارم تعداد مسافرانی که سوار قطار هستند نشان داده می شود.

فرمت خروجی	دستور ورودی
<code>status : <resting/moving></code> <code>estimated remaining time : <time></code> <code>next station : <station name></code> <code>passengers count : <number of passengers></code>	<code>show_train_info <train id></code>

مثال خروجی	مثال ورودی
<code>status : resting</code> <code>estimated remaining time : 10</code> <code>next station : oxford</code> <code>passengers count : 4</code>	<code>show_train_info 3</code>

نکات تکمیلی

- به ورودی و خروجی‌های نمونه که در صفحه درس بارگذاری شده‌اند برای درک بهتر توجه کنید.
- تضمین می‌شود در هنگام سنجش صحت عملکرد برنامه، تمام ورودی‌ها معتبر خواهند بود.. در نتیجه نیازی به بررسی حالات نامعتبر نیست.
- دقت شود که هدف از تمرین طراحی شیء‌گرای مسئله است. بدیهی است به سایر پیاده‌سازی‌ها نمره‌ای تعلق نخواهد گرفت.
- دقت کنید شما مجاز به استفاده از مفاهیم ارث‌بری و چندریختی نیستید. مفاهیمی که در آینده در کلاس درس می‌خوانید را استفاده نکنید و به هدف تمرین توجه داشته باشید.
- در صورت وجود سوال یا ابهام می‌توانید در فروم پروژه یا از طریق ایمیل آن را مطرح کنید.

نحوه تحویل

- کد خود را در قالب یک فایل با نام A4-SID.cpp در صفحه eLearn درس بارگذاری کنید که SID شماره دانشجویی شماست؛ برای مثال اگر شماره دانشجویی شما ۸۱۰۱۹۹۹۹۹ باشد، نام پرونده شما باید A4-810199999.cpp باشد که شامل کد شما است.
- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- تمیزی کد، شکستن مرحله به مرحله مسئله و طراحی مناسب، در کنار تولید خروجی دقیق و درست، بخش مهمی از نمره شما را تعیین خواهد کرد.

- درستی برنامه شما از طریق آزمون‌های خودکار سنجیده می‌شود؛ بنابراین پیشنهاد می‌شود که با استفاده از ابزارهایی مانند diff خروجی برنامه خود را با خروجی‌هایی که در اختیارتان قرار داده شده است مطابقت دهید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.