تمرین دوم برنامه نویسی پیشرفته

استاد مرضیه ملکی مجد

مبحث : مقدمات شی گرایی، آرایه، متد

طراح ها :

@IamAmirSalimi - امیرحسین سلیمی - سوال 1 و 2

@FatemehBagherii - فاطمه سادات باقری - سوال 3

@LibreArtisan - میلاد زارعی - سوال 4

زمستان 1403

1. پیدا کردن کوچکترین عدد صحیح مثبت گمشده

برنامه ای بنویسید که کوچکترین عدد صحیح مثبت که در آرایه وجود ندارد را پیدا کند.

- آرایه میتواند شامل اعداد منفی، صفر و مثبت باشد.
- از لوپ تو در تو استفاده نکنید.

راهنمایی :

به جای استفاده از حافظه اضافی، از خود آرایه ورودی برای ذخیره اطلاعات استفاده کنید. اعداد نامربوط را مدیریت کنید و از اندیس ها به عنوان راهی برای نشان دادن وجود اعداد کمک بگیرید. سپس، اولین جایی که نشانه ای از وجود عدد ندارد، جواب شماست. اگر همه چیز نشانه گذاری شده بود، جواب بیرون از محدوده ی آرایه است.

تذکر مهم:

اگر از آرایه کمکی یا ساختارهای داده های اضافی (مانند HashSet) استفاده کنید، نصف نمره را دریافت خواهید کرد.

هدف این است که مسئله را فقط با استفاده از آرایه ورودی و بدون حافظه اضافی حل کنید.

ورودی:

یک آرایه از اعداد صحیح (nums) با طول n.

خروجی:

کوچکترین عدد صحیح مثبت که در آرایه وجود ندارد.

مثالها:

ورودی: { 3, 4, -1, 1 } ، خروجی: 2

ورودی: { 7, 8, 9, 11, 12 } ، خروجی: 1

2.مدیریت پروژه‌ها

در این تمرین، شما باید یک سیستم مدیریت پروژه بسازید که در آن کار با آرایه‌ها و انجام عملیات‌های پیچیده بر روی داده‌ها بررسی شود. این تمرین شامل استفاده از کلاس‌ها، آرایه‌ها، متدهای مختلف و محاسبات پیچیده می‌باشد.

مشخصات سیستم:

1. **کلاس `Employee`**:
   - ایجاد یک کلاس به نام `Employee` که ویژگی‌های زیر را داشته باشد:
     - `name` (نام کارمند)
     - `id` (شناسه کارمند)
     - `role` (نقش کارمند در تیم): نوع داده `string` (مثلاً: Developer, Manager, Designer)
   - این کلاس باید شامل یک سازنده (Constructor) باشد که مقادیر ویژگی‌ها را هنگام ایجاد شی مقداردهی کند.
   - همچنین این کلاس باید شامل متدی به نام `getEmployeeInfo` باشد که اطلاعات کامل کارمند را نمایش دهد.
2. **کلاس `Project`**:
   - ایجاد یک کلاس به نام `Project` که ویژگی‌های زیر را داشته باشد:
     - `name` (نام پروژه)
     - `startDate` (تاریخ شروع پروژه): نوع داده `DateTime`
     - `endDate` (تاریخ پایان پروژه): نوع داده `DateTime`
     - `teamMembers` (اعضای تیم): یک آرایه از اشیاء `Employee` که اعضای تیم پروژه را ذخیره می‌کند.

- این کلاس باید شامل متدهای زیر باشد:
  - `addTeamMember`: این متد باید یک شی از کلاس `Employee` را به تیم پروژه اضافه کند.
  - `getProjectDuration`: این متد باید طول مدت زمان پروژه (تعداد روزها) را محاسبه کند.
  - `getAverageTeamAge`: این متد باید میانگین سن اعضای تیم پروژه را محاسبه کند.
  - `getMaxTeamMemberAge`: این متد باید بیشترین سن را در بین اعضای تیم پیدا کند.
  - `getProjectDetails`: این متد باید جزئیات کامل پروژه (نام، تاریخ شروع، تاریخ پایان، اعضای تیم و میانگین سن اعضای تیم) را نمایش دهد.

3. **کلاس `ProjectManager`:**
   - ایجاد یک کلاس به نام `ProjectManager` که ویژگی‌های زیر را داشته باشد:
     - `projects` (پروژه‌ها): یک آرایه از اشیاء `Project` که پروژه‌های مختلف شرکت را ذخیره می‌کند.
     - `projectCount` (تعداد پروژه‌ها): تعداد پروژه‌های موجود در شرکت.
   - این کلاس باید شامل متدهای زیر باشد:
     - `addProject`: این متد باید یک شی از کلاس `Project` را به آرایه پروژه‌ها اضافه کند.
     - `findProjectByName`: این متد باید پروژه‌ای را با نام خاص پیدا کند و اطلاعات آن را نمایش دهد.

- `sortProjectsByDuration:` این متد باید پروژه‌ها را بر اساس مدت زمان پروژه‌ها مرتب کند (بر اساس تعداد روزها).
- `getProjectWithMaxDuration`: این متد باید پروژه‌ای که بیشترین مدت زمان را داشته باشد پیدا کند و اطلاعات آن را نمایش دهد.
- `getAllProjectsDetails`: این متد باید تمام پروژه‌ها را به همراه جزئیات آنها (نام پروژه، تاریخ شروع، تاریخ پایان، و اعضای تیم) نمایش دهد.

4.**کلاس `CompanyApp`**:

- در این کلاس، باید یک منوی داینامیک برای تعامل با کاربر داشته باشید که از طریق آن کاربر بتواند عملیات مختلف را انجام دهد:
  1. نمایش جزئیات تمام پروژه‌ها.
  2. جستجو پروژه بر اساس نام.
  3. مرتب‌سازی پروژه‌ها بر اساس مدت زمان.
  4. پیدا کردن پروژه با بیشترین مدت زمان.
  5. افزودن پروژه جدید به سیستم.
  6. افزودن اعضای جدید به تیم پروژه.

وظایف شما:

1. برای پیاده‌سازی ابتدا کلاس‌های `Employee` و `Project` را طراحی کنید.
2. سپس یک کلاس `ProjectManager` برای مدیریت پروژه‌ها بنویسید.
3. یک منوی داینامیک در `CompanyApp` ایجاد کنید که با کاربر تعامل داشته باشد و از طریق آن بتوانید عملیات‌های مختلف را انجام دهید.

نکات اضافی:

- از **آرایه‌ها** برای ذخیره پروژه‌ها و اعضای تیم استفاده کنید.
- از **الگوریتم مرتب‌سازی** برای مرتب‌سازی پروژه‌ها بر اساس مدت زمان استفاده کنید.

سوال 3: سیستم مدیریت سفارش‌های رستوران

هدف تمرین:

در این تمرین شما موظف به پیاده‌سازی یک سیستم ساده برای مدیریت سفارش‌های یک رستوران هستید. در این سیستم، اطلاعات منو (غذاها) و سفارش‌ها ثبت شده و عملیات‌های افزودن، حذف، جستجو و نمایش اطلاعات انجام می‌شود. هدف این تمرین تقویت مفاهیم شی‌گرایی (تعریف کلاس، سازنده، متد و ...) به همراه کار با آرایه‌ها (ذخیره و مدیریت مجموعه‌ای از اشیاء) می‌باشد.

1. تعریف کلاس MenuItem

ویژگی‌ها:

- `itemId` (شناسه آیتم): عدد صحیح
- `name` (نام غذا): رشته
- `description` (توضیحات غذا): رشته
- `price` (قیمت غذا): عدد اعشاری (double)

سازنده:

- یک سازنده که تمامی ویژگی‌ها را هنگام ایجاد شی از کلاس MenuItem مقداردهی کند.

متد:

`displayItemInfo`: متدی که اطلاعات غذا را نمایش دهد:

2. تعریف کلاس Order

ویژگی‌ها:

- `orderId` (شناسه سفارش): عدد صحیح
- `tableNumber` (شماره میز): عدد صحیح
- `items` (آرایه‌ای از اشیاء MenuItem): آرایه با ظرفیت ثابت (مثلاً 10 آیتم) برای ذخیره غذاهای سفارش داده شده.
- `itemCount` (تعداد آیتم‌های سفارش): عدد صحیح که تعداد آیتم‌های فعلی در سفارش را نگهداری می‌کند.
- `status` (وضعیت سفارش): رشته (مانند "در حال آماده‌سازی"، "سفارش تحویل داده شده")

سازنده:

- سازنده‌ای که `orderId` و `tableNumber` را مقداردهی کند و آرایه `items` به اندازه مشخص (مثلاً 10) ایجاد شده و `itemCount` صفر تنظیم شود. همچنین مقدار اولیه `status` را می‌توان به عنوان "در حال آماده‌سازی" قرار داد.

متدها:

- `addItem`:
  - این متد یک شی MenuItem را به آرایه `items` اضافه می‌کند.
  - در صورت پر بودن آرایه، پیام مناسب نمایش داده شود.

- `removeItemById`:

- این متد آیتمی از آرایه سفارش حذف می‌کند که شناسه آن برابر با `itemId` ورودی باشد.
- در صورت عدم یافتن آیتم، پیام مناسبی نمایش داده شود.

- `calculateTotal`:
  - متدی که مجموع قیمت تمامی آیتم‌های موجود در آرایه `items` را محاسبه و برمی‌گرداند.

- `displayOrderInfo`:
  - متدی که جزئیات سفارش شامل شماره سفارش، شماره میز، اطلاعات هر آیتم (با استفاده از متد `displayItemInfo` کلاس MenuItem)، مجموع قیمت و وضعیت سفارش را نمایش دهد.

3. تعریف کلاس Restaurant

ویژگی‌ها:

- `menu`: آرایه‌ای از اشیاء MenuItem که منوی رستوران را تشکیل می‌دهد (ظرفیت ثابت مثلاً 20 آیتم)
- `orders`: آرایه‌ای از اشیاء Order جهت ذخیره سفارش‌های دریافت شده (ظرفیت ثابت مثلاً 50 سفارش)
- `menuItemCount`: تعداد آیتم‌های فعلی منو

- `orderCount`: تعداد سفارش‌های ثبت شده

متدها:

- `addMenuItem`:
  - این متد شی جدیدی از کلاس MenuItem را به آرایه `menu` اضافه می‌کند.
  - در صورت پر بودن منو، پیام مناسبی نمایش داده شود.

- `displayMenu`:
  - متدی که اطلاعات تمامی آیتم‌های منوی رستوران (با استفاده از متد `displayItemInfo`) را نمایش دهد.

- `addOrder`:
  - متدی برای افزودن یک سفارش جدید به آرایه `orders`.

- `removeOrderById`:
  - این متد سفارش با شناسه‌ی مشخص شده را از آرایه حذف می‌کند.
  - در صورت عدم یافتن سفارش با شناسه ورودی، پیام مناسبی نمایش داده شود.

- `searchOrderByTable`:

- متدی که آرایه `orders` را جستجو کرده و سفارش‌هایی با شماره میز برابر با ورودی را پیدا و جزئیات آن‌ها را نمایش دهد.
- اگر سفارش مربوطه یافت نشد، پیام مناسبی نمایش داده شود.

- `displayAllOrders`:
  - متدی که اطلاعات تمامی سفارش‌های موجود در آرایه `orders` (با استفاده از متد `displayOrderInfo` کلاس Order) را نمایش دهد.

4. تعریف کلاس RestaurantApp

وظیفه:

- در این کلاس، یک منوی داینامیک پیاده‌سازی کنید که از کاربر ورودی دریافت کرده و با توجه به انتخاب‌های او، عملیات‌های مختلف زیر را انجام دهد:
  - **مدیریت منو:**
    - افزودن یک آیتم جدید به منو
    - نمایش منوی رستوران
  - **مدیریت سفارش‌ها:**
    - ثبت سفارش جدید (با مشخص کردن شماره میز و شناسه سفارش)
    - افزودن آیتم (غذا) به سفارش موجود
    - حذف سفارش بر اساس شناسه سفارش
    - جستجو سفارش‌ها بر اساس شماره میز
    - نمایش جزئیات تمامی سفارش‌ها

- محاسبه و نمایش مجموع مبلغ یک سفارش خاص (با استفاده از متد `calculateTotal`)

نکات اضافی:

- از آرایه‌ها برای ذخیره آیتم‌های منو و سفارش‌ها استفاده کنید.
- بررسی‌های لازم (مانند عدم اضافه شدن آیتم در صورت پر بودن آرایه یا عدم یافتن سفارش/آیتم برای حذف) را انجام دهید.
- سعی کنید طراحی منو به گونه‌ای باشد که کاربر بتواند به راحتی بین گزینه‌های مختلف جابجا شود.

تمرین 4: سیستم ثبت‌نام رویداد

هدف تمرین:

در این تمرین، شما موظفید تا یک سیستم ثبت‌نام برای رویدادها (مانند کنفرانس، کارگاه یا همایش) پیاده‌سازی کنید. این سیستم باید با استفاده از مفاهیم شی‌گرایی و آرایه‌ها، اطلاعات رویدادها و شرکت‌کنندگان در هر رویداد را مدیریت کند. عملیات‌های مورد نیاز شامل افزودن، حذف، جستجو و نمایش جزئیات اطلاعات است.

---

1. تعریف کلاس Participant (شرکت‌کننده)

ویژگی‌ها:

- `participantId` (شناسه شرکت‌کننده): عدد صحیح
- `name` (نام شرکت‌کننده): رشته
- `email` (ایمیل شرکت‌کننده): رشته

سازنده:

- یک سازنده که هنگام ایجاد شی از کلاس Participant، تمامی ویژگی‌ها (شناسه، نام، ایمیل) را مقداردهی کند.

متد:

`displayInfo`: متدی که اطلاعات شرکت‌کننده را به صورت فرمت زیر نمایش دهد:

Participant ID: participantId

Name: name

Email: email

2. تعریف کلاس Event (رویداد)

ویژگی‌ها:

- `eventId` (شناسه رویداد): عدد صحیح
- `title` (عنوان رویداد): رشته
- `date` (تاریخ برگزاری): رشته (یا در صورت تمایل از نوع DateTime استفاده شود)
- `location` (محل برگزاری): رشته
- `participants`: آرایه‌ای از اشیاء Participant جهت ذخیره شرکت‌کنندگان (ظرفیت ثابت، مثلاً 50 نفر)
- `participantCount`: تعداد فعلی شرکت‌کنندگان ثبت‌شده در رویداد

سازنده:

- سازنده‌ای که `date`، `title`، `eventId` و `location` را مقداردهی کند و آرایه `participants` را با ظرفیت تعیین‌شده ایجاد کرده و `participantCount` را برابر با صفر تنظیم کند.

متدها:

- `addParticipant`:
  - این متد یک شی جدید از کلاس Participant را به آرایه `participants` اضافه می‌کند.
  - در صورت پر بودن ظرفیت آرایه، پیام مناسبی نمایش داده شود.

- `removeParticipantById`:

- این متد شرکت‌کننده‌ای با شناسه مشخص را از آرایه حذف می‌کند.
- در صورت عدم یافتن شرکت‌کننده با آن شناسه، پیام مناسب نمایش داده شود.

- `searchParticipantByName`:
  - این متد آرایه `participants` را برای یافتن شرکت‌کننده‌ای که نامش با ورودی مطابقت دارد جستجو کرده و در صورت یافتن، اطلاعات او را نمایش می‌دهد.
  - در صورت عدم تطابق، پیام "شرکت‌کننده‌ای با این نام یافت نشد" نمایش داده شود.

- `displayEventInfo`:
  - این متد جزئیات رویداد شامل شناسه، عنوان، تاریخ، محل برگزاری و لیست تمامی شرکت‌کنندگان (با فراخوانی متد `displayInfo` هر شرکت‌کننده) را نمایش می‌دهد.

3. تعریف کلاس EventManager (مدیریت رویدادها)

ویژگی‌ها:

- `events`: آرایه‌ای از اشیاء Event جهت ذخیره رویدادهای موجود (ظرفیت ثابت مثلاً 20 رویداد)
- `eventCount`: تعداد رویدادهای ثبت‌شده

متدها:

- `addEvent`:
  - این متد شی جدید از کلاس Event را به آرایه `events` اضافه می‌کند.
  - در صورت پر بودن آرایه، پیام مناسبی نمایش داده شود.

- `removeEventById`:
  - این متد رویدادی با شناسه مشخص را از آرایه حذف می‌کند.
  - در صورت عدم یافتن رویداد با شناسه ورودی، پیام مناسبی نمایش داده شود.

- `searchEventByTitle`:
  - این متد آرایه `events` را جستجو کرده و رویدادی با عنوان مطابق با ورودی را پیدا می‌کند و جزئیات آن را (با فراخوانی `displayEventInfo`) نمایش می‌دهد.
  - در صورت عدم یافتن رویداد، پیام "رویدادی با این عنوان یافت نشد" نمایش داده شود.

- `displayAllEvents`:
  - این متد اطلاعات تمامی رویدادهای موجود در آرایه `events` را به همراه جزئیات آن‌ها نمایش می‌دهد.

4. تعریف کلاس EventRegistrationApp (برنامه ثبت‌نام رویداد)

وظیفه:

- در این کلاس یک منوی داینامیک پیاده‌سازی کنید که از کاربر ورودی دریافت کرده و بر اساس انتخاب‌های او عملیات‌های زیر را انجام دهد:

بخش مدیریت رویدادها:

- افزودن رویداد جدید (با وارد کردن شناسه، عنوان، تاریخ و محل برگزاری)
- حذف رویداد بر اساس شناسه
- جستجو رویداد بر اساس عنوان
- نمایش تمامی رویدادها

- **بخش مدیریت ثبت‌نام در یک رویداد:**

  - برای یک رویداد انتخاب شده (بر اساس شناسه یا عنوان)
    - افزودن یک شرکت‌کننده جدید (با وارد کردن شناسه، نام و ایمیل)
    - حذف شرکت‌کننده بر اساس شناسه
    - جستجو شرکت‌کننده بر اساس نام
    - نمایش جزئیات کامل رویداد به همراه لیست شرکت‌کنندگان

نکات اضافی:

- از آرایه‌ها برای ذخیره رویدادها و شرکت‌کنندگان استفاده کنید.
- در هر متد بررسی‌های لازم مانند بررسی پر بودن آرایه یا عدم یافتن مورد برای حذف/جستجو را انجام دهید.
- طراحی منو باید به گونه‌ای باشد که کاربر به راحتی بین گزینه‌ها حرکت کرده و عملیات مورد نظر را انتخاب کند.