



پروژه سیستم رزرواسیون بلیط هواپیمایی

مقدمه

هدف از این پروژه، طراحی و پیاده‌سازی یک سیستم رزرواسیون بلیط هواپیما با استفاده از مفاهیم برنامه‌نویسی شیء‌گرا (OOP) در زبان C++ است. در این پروژه شما باید با طراحی کلاس‌های مناسب، سیستم مدیریت مسافران، پروازها و رزروها را پیاده‌سازی کرده و ذخیره‌سازی و بازیابی اطلاعات را با استفاده از فایل‌های متنی TXT یا CSV انجام دهید. این پروژه به شما کمک می‌کند تا با اصول پایه‌ای طراحی نرم‌افزار و پردازش فایل‌ها آشنا شوید.

توضیحات کلی پروژه

- تمامی داده‌های سیستم (مسافران، پروازها، رزروها) باید در فایل‌های متنی TXT یا CSV ذخیره شوند.
- فرمت ذخیره‌سازی داده‌ها باید به گونه‌ای باشد که در اجرای مجدد برنامه، اطلاعات قبلی به درستی بازیابی شوند.
- استفاده از پایگاه‌داده خارجی (مانند SQLite) مجاز نیست.

مدیریت فایل

- داده‌ها باید در سه فایل جداگانه با فرمت txt یا csv ذخیره شوند:
 ۱. passengers.txt یا passengers.csv
 ۲. flights.txt یا flights.csv
 ۳. reservations.txt یا reservations.csv
- در صورت عدم وجود فایل‌ها، برنامه باید فایل‌های جدید ایجاد کند.

ویژگی‌های اصلی مورد نیاز

۱. مدیریت مسافران

• ثبت مسافر جدید

- passenger_id (شناسه منحصر به فرد - تولید خودکار)
- نام و نام خانوادگی
- شماره گذرنامه
- کد ملی
- ملیت
- موجودی کیف پول
- جستجوی مسافر بر اساس هر یک از شاخص‌های زیر:
 - passenger_id
 - کد ملی
 - شماره گذرنامه
 - نام (یا بخشی از نام)
- نمایش لیست تمام مسافران
- ویرایش اطلاعات مسافر (به‌روزرسانی موجودی، تغییر گذرنامه و...)
- ذخیره و بارگذاری خودکار اطلاعات مسافران در فایل

۲. مدیریت پروازها

• ثبت پرواز جدید

- flight_id (شناسه منحصر به فرد - تولید خودکار)
- شماره پرواز
- مبدا و مقصد
- تاریخ و ساعت پرواز
- تعداد صندلی‌های خالی
- قیمت ~~تک~~ ^{تک}
- جستجوی پرواز(ها) بر اساس هر یک از شاخص‌های زیر:
 - flight_id
 - شماره پرواز
 - مبدا و مقصد
 - تاریخ پرواز

- کاهش صندلی‌های خالی پس از رزرو
- نمایش لیست پروازهای موجود
- ذخیره و بارگذاری خودکار اطلاعات پروازها در فایل

۳. رزرویشن (Reservation)

برنامه‌ی شما باید یک سیستم ساده برای رزرو بلیت پرواز داشته باشد که شامل اطلاعات زیر در هر رزرو باشد:

- reservation_id (شناسه منحصر به فرد – تولید خودکار)
- passenger_id (شناسه مسافر)
- flight_id (شناسه پرواز)
- amount_paid (مبلغ پرداخت‌شده)

دستورات

- رزرو یک پرواز

با وارد کردن حداقل passenger_id و flight_id، کاربر باید بتواند یک بلیت رزرو کند.

ابتدا یک پیش‌نمایش شامل اطلاعات کامل مسافر و پرواز نمایش داده می‌شود. پس از تأیید، رزرو انجام شده و اطلاعات کامل بلیت ثبت و نمایش داده می‌شود.

- شرایط خطا:

- **PassengerNotFound**: اگر مسافر با شناسه واردشده یافت نشود.
- **FlightNotFound**: اگر پرواز با شناسه واردشده یافت نشود.
- **FlightFull**: اگر پرواز ظرفیت ندارد و همه صندلی‌ها پر شده‌اند.
- **FlightExpired**: اگر تاریخ پرواز گذشته باشد و امکان رزرو وجود نداشته باشد.
- **InsufficientBalance**: اگر موجودی کیف پول مسافر برای خرید بلیت کافی نباشد.

- جستجو در بلیت‌ها

با وارد کردن reservation_id، اطلاعات کامل بلیت شامل اطلاعات رزرو، مسافر و پرواز نمایش داده می‌شود.

- در صورت عدم وجود رزرو:

- **ReservationNotFound**: اگر رزرو با شناسه واردشده یافت نشود.

• استرداد بلیت (لغو رزرو)

با وارد کردن reservation_id، کاربر می‌تواند بلیت خود را لغو کند. ابتدا اطلاعات مربوط به بلیت نمایش داده میشوند در صورت تأیید، درصدی از مبلغ پرداخت‌شده به کیف پول مسافر برگشت داده می‌شود.

- قوانین بازگشت وجه:

- زمان باقی‌مانده تا پرواز: بیش از ۴۸ ساعت، درصد بازگشت: ۹۰٪
- زمان باقی‌مانده تا پرواز: بین ۲۴ تا ۴۸ ساعت، درصد بازگشت: ۵۰٪
- زمان باقی‌مانده تا پرواز: کمتر از ۲۴ ساعت، غیرقابل استرداد: ۰٪

- شرایط خطا:

- **ReservationNotFound**: اگر رزرو با شناسه وارد شده یافت نشود.
- **FlightCompleted**: اگر پرواز انجام شده و دیگر امکان استرداد وجود ندارد.
- **RefundNotAllowed**: اگر شرایط استرداد (زمان کم مانده به پرواز یا دیگر شرایط) فراهم نباشد.

۴. سیستم خطوط هوایی (AirlineSystem)

برنامه‌ی شما باید به‌گونه‌ای طراحی شده باشد که به کاربر کمک کند سیستم خطوط هوایی را مدیریت کرده و بتواند مسافران، پروازها و رزروها را کنترل نماید. همچنین باید امکان گزارش‌گیری فراهم باشد تا این فرایندها با دقت بیشتری انجام شوند.

دستورات

۱. مدیریت فایل

برنامه باید به‌گونه‌ای طراحی شود که در هنگام خروج (چه به‌صورت عادی و چه به‌صورت ناگهانی) بتواند تمام اطلاعات را ذخیره کرده و در اجرای مجدد، تمامی داده‌های قبلی را به‌درستی بازیابی نماید. این شامل اطلاعات مربوط به مسافران، پروازها و رزروها می‌باشد.

۲. افزودن، حذف و جستجو

- **افزودن**: کاربر باید بتواند مسافر یا پرواز جدیدی ثبت کند. اطلاعات ورودی باید دارای اعتبارسنجی (validation) باشند تا از ورود داده‌های اشتباه جلوگیری شود.

- **حذف:** حذف باید به صورت **soft delete** انجام شود؛ یعنی اطلاعات حذف شده از دید کاربر مخفی شده و در جستجوها نمایش داده نشوند، اما به طور کامل از پایگاه داده پاک نشوند.

۳. تولید گزارش ها

برنامه باید بتواند گزارش های متنوعی تولید کرده و آنها را در قالب فایل **CSV** ذخیره نماید. قبل از ذخیره سازی، باید پیش نمایشی از داده ها به کاربر نمایش داده شود.

- **گزارش های رزرو:**

- مشاهده ی تمامی رزروها
- مشاهده ی رزروهای آینده
- مشاهده ی رزروهای انجام شده
- مشاهده ی رزروهای استرداد شده

- **گزارش های پرواز:**

- لیست تمامی مسافران یک پرواز
- لیست پروازهای یک روز مشخص
- لیست پروازهای آینده
- لیست پروازهای گذشته

- **گزارش های مسافر:**

- لیست تمام بلیت های خریداری شده و پروازهای انجام شده
- لیست سفرهای آینده
- لیست سفرهای استرداد شده

نحوه تست و ارزیابی

- برنامه باید همراه با یک فایل تست (**test.cpp**) ارائه شود که تمامی قابلیت‌های سیستم را بررسی کند.
- می‌توانید از **testing framework** ها مانند **Catch2** استفاده کنید یا یک تابع **main()** جداگانه در فایل تست برای تست بنویسید. برای اطلاعات بیشتر در خصوص استفاده از **Catch2** می‌توانید به [لینک](#) مراجعه کنید.
- فایل تست باید شامل موارد ثبت و جستجوی مسافر، اضافه کردن و کاهش ظرفیت پرواز، خرید و استرداد بلیط، خطاهای احتمالی (مانند پرواز پر یا موجودی ناکافی)، ذخیره و بازیابی خودکار داده‌ها و دیگر ویژگی‌های کلاس‌ها را پوشش دهد.

نمرات

- **تمیزی کد**
 - رعایت کردن نام‌گذاری صحیح و انسجام
 - عدم استفاده از متغیرهای **global**
 - استفاده صحیح از متغیرهای ثابت به جای **Magic Value** ها
- **درستی کد**
 - برنامه باید بدون خطا و هشدار (**warning**) کامپایل شود.

- **طراحی:**

- **منطق برنامه نباید در **main()** باشد** بلکه باید در کلاس‌ها و توابع مجزا پیاده‌سازی شود.
- شکستن به کلاسهای مناسب و تخصیص مسئولیتهای صحیح به هر کلاس
- رعایت سطح دسترسی (**public/private**) در ویژگی‌های کلاس

نکات پایانی

- دقت کنید که کد شما به گونه ای زده شده باشد که نتایج قابلیت بازتولید داشته باشد.
- توجه کنید این تمرین باید به صورت تک نفره انجام شود. کد تکراری یا کپی‌شده از دیگران موجب نمره صفر می‌شود.