

عنوان: گزارشکار شیوه انجام پروژه تور مسافرتی

عنوان درس: برنامه نویسی پیشرفته و کارگاه آن

اساتید: استاد مینوفام - استاد فرشچی

نام و نام خانوادگی : ملیکا بلبل آبادی

شماره دانشجویی: 981113010

دانشکده فنی مهندسی دانشگاه علم و فرهنگ

مقدمه :

این پروژه برنامه نویسی به زبان جاوا میباشد و موضوع آن مدیریت تور مسافرتی است. معماری استفاده شده در کد نویسی آن معماری MVC میباشد و از 3package و 32class برای ساختار آن استفاده شده است . تمامی اطلاعات هم در پایگاه داده MySQL به نام traveltour ذخیره و نگهداری میشود. اطلاعات ما در این پروژه شامل موارد زیر میباشد :

- اطلاعات تورهای موجود
- اطلاعات افراد ثبت نامی در تورها
- اطلاعات هتل های طرف قرارداد تورها
- اطلاعات وسایل نقلیه مورد استفاده
- اطلاعات مکان های دیدنی
- اطلاعات مدیران تور

در این پروژه امکان ثبت ، ویرایش و حذف هرکدام از اطلاعات توسط ادمینِ تور مسافرتی موجود است.

شرح معماری پروژه:

همانطور که گفته شد معماری این پروژه MVC میباشد و برای پیاده سازی آن از 3 پکیج به نام های زیر استفاده شده است:

- Tourproject
- View
- Controller

در ادامه به توضیح هریک از پکیج ها میپردازم:

پکیج : tourproject

این پکیج که خود شامل 6 پکیج دیگر به نام های زیر میباشد، برای پیاده سازی لایه model ایجاد شده است :

- HOTEL •
- MANAGERS •
- PLACES •
- PEOPLES •
- VEHICLES •
- tour •

و هرکدام از این 6 پکیج شامل 3 کلاس برای پیاده سازی لایه model معماری MVC میباشد، یعنی:

- Entity •
- Repository •
- Service •

در کلاس Entity متعلق به هر پکیج ، برای متغیرها از Encapsulation استفاده شده است. و در کلاس های Repository دستورات پایگاه داده برای ارتباط برقرار کردن با دیتابیس مورد نظر، ثبت ، ویرایش و حذف اطلاعات مورده استفاده قرار گرفته است. در کلاس های Service هم متد های ثبت و ویرایش و حذف اطلاعات از پایگاه داده با استفاده از کلاس های Repository و Service ایجاد شده است.

پکیج : View

این پکیج شامل 7 کلاس میباشد که به منظور پیاده سازی لایه View معماری MVC ایجاد شده است.

اسامی کلاس ها به صورت زیر میباشد:

- HotelForm •
- LoginForm •
- MainForm •
- PeopleForm •
- PlaceForm •
- VehicleForm •
- TourForm •

در این کلاس ها از رابط کاربری SWING استفاده شده است به طوری که اولین صفحه برنامه از کلاس loginform شروع میشود و سپس بقیه کلاس ها و متد ها طبق خواسته ما فراخوانی میشوند.

پکیج Controller :

همانطور که مشخص است از این پکیج برای پیاده سازی لایه controller معماری MVC استفاده شده است و کلاس های هم نام کلاس های پکیج View دارد. و هر کلاس متد main را به منظور اجرای برنامه داراست و در آن متد هم نام خود در لایه View را فراخوانی میکند. هر کلاس در این پکیج قابلیت Run شدن به تنهایی را دارد ولی در اصل از کلاس LoginForm آغاز میشود و با وارد کردن یوزرنیم و پسورد admin باقی تب ها با توجه به انتخاب و نیاز ما نمایش داده میشود.

شرح پایگاه داده پروژه:

از پایگاه داده MySQL استفاده شده است و نام آن traveltour میباشد و طبق جدول زیر ، جدول ها و هر یک از کامپوننت ها ساخته شده است.

Hotels	code	name	address	facilities	payment	-	-	-	-
Managers	name	username	password	-	-	-	-	-	-
Peoples	codemelli	fullname	phone	address	tourcode	-	-	-	-
Places	code	name	-	-	-	-	-	-	-
Vehicles	code	name	property	-	-	-	-	-	-
tours	code	name	range	vehicle	people	totalpayment	manager	hotelname	places

شرح Class ها:

Tourproject.HOTEL.EntityHotel: دارای 6 متغیر از نوع `String` , `int` , `long` است و هرکدام را با استفاده از Encapsulation و متدهای `getter` و `setter` گرفته و ست کرده ایم. این کلاس و متدهایش در کلاس های `Repository` و `Service` متعلق به پکیج خودش فراخوانی شده و مورد استفاده قرار میگیرد.

Tourproject.HOTEL.RepositoryHotel: در این کلاس از 5 کلاس استفاده شده است. که اولی `constructor` است و در بدنه اش متد `connect` فراخوانی شده است. متد `connect` به منظور اتصال و ارتباط برقرار کردن با دیتابیس `traveltour` ایجاد شده است. متد `insertHotel` به منظور ذخیره اطلاعات در جدول `hotels` ایجاد شده و متدهای `updateHotel` و `deleteHotel` به ترتیب برای ویرایش و حذف اطلاعات از جدول مذکور ایجاد شده اند.

Tourproject.HOTEL.ServiceHotel: در این کلاس 3 متد به منظور ذخیره و ویرایش و حذف اطلاعات با فراخوانی و شی سازی از کلاس `Repository` و متدهای آن ایجاد شده است. متد `save` با شی ساختن از کلاس `Repository` و فراخوانی متد `insertHotel` ، متد `edit` با شی ساختن از کلاس `Repository` و فراخوانی متد `updateHotel` و متد `delete` با شی ساختن از کلاس `Repository` و فراخوانی متد `deleteHotel` به ترتیب کار ذخیره و ویرایش و حذف اطلاعات را انجام میدهند.

Tourproject.HOTEL.EntityManager: دارای 3 متغیر از نوع `String` است و هرکدام را با استفاده از Encapsulation و متدهای `getter` و `setter` گرفته و ست کرده ایم. این کلاس و متدهایش در کلاس های `Repository` و `Service` متعلق به پکیج خودش فراخوانی شده و مورد استفاده قرار میگیرد.

Tourproject.HOTEL.RepositoryManagers: در این کلاس از 5 کلاس استفاده شده است. که اولی `constructor` است و در بدنه اش متد `connect` فراخوانی شده است. متد `connect` به منظور اتصال و

ارتباط برقرار کردن با دیتابیس traveltour ایجاد شده است. متد insertManagers به منظور ذخیره اطلاعات در جدول Managers ایجاد شده و متدهای updateManagers و deleteManagers به ترتیب برای ویرایش و حذف اطلاعات از جدول مذکور ایجاد شده اند.

Tourproject.HOTEL.ServiceManager: در این کلاس 3 متد به منظور ذخیره و ویرایش و حذف اطلاعات با فراخوانی و شی سازی از کلاس Repository و متدهای آن ایجاد شده است. متد save با شی ساختن از کلاس Repository و فراخوانی متد insertManager، متد edit با شی ساختن از کلاس Repository و فراخوانی متد updateManager و متد delete با شی ساختن از کلاس Repository و فراخوانی متد deleteManager به ترتیب کار ذخیره و ویرایش و حذف اطلاعات را انجام میدهند.

Tourproject.HOTEL.EntityPeople: دارای 5 متغیر از نوع String, long است و هر کدام را با استفاده از Encapsulation و متدهای getter و setter گرفته و ست کرده ایم. این کلاس و متدهایش در کلاس های Repository و Service متعلق به پکیج خودش فراخوانی شده و مورد استفاده قرار میگیرد.

Tourproject.HOTEL.RepositoryPeoples: در این کلاس از 5 کلاس استفاده شده است. که اولی constructor است و در بدنه اش متد connect فراخوانی شده است. متد connect به منظور اتصال و ارتباط برقرار کردن با دیتابیس traveltour ایجاد شده است. متد insertPeoples به منظور ذخیره اطلاعات در جدول Peoples ایجاد شده و متدهای updatePeoples و deletePeoples به ترتیب برای ویرایش و حذف اطلاعات از جدول مذکور ایجاد شده اند.

Tourproject.HOTEL.ServicePeoples: در این کلاس 3 متد به منظور ذخیره و ویرایش و حذف اطلاعات با فراخوانی و شی سازی از کلاس Repository و متدهای آن ایجاد شده است. متد save با شی ساختن از کلاس Repository و فراخوانی متد insertPeople، متد edit با شی ساختن از کلاس Repository و فراخوانی متد updatePeople و متد delete با شی ساختن از کلاس Repository و فراخوانی متد deletePeople به ترتیب کار ذخیره و ویرایش و حذف اطلاعات را انجام میدهند.

Tourproject.HOTEL.EntityPlace : دارای 2 متغیر از نوع String , long است و هرکدام را با استفاده از Encapsulation و متدهای getter و setter گرفته و ست کرده ایم. این کلاس و متدهایش در کلاس های Repository و Service متعلق به پکیج خودش فراخوانی شده و مورد استفاده قرار میگیرد.

Tourproject.HOTEL.RepositoryPlaces : در این کلاس از 5 کلاس استفاده شده است. که اولی constructor است و در بدنه اش متد connect فراخوانی شده است. متد connect به منظور اتصال و ارتباط برقرار کردن با دیتابیس traveltour ایجاد شده است. متد insertPlaces به منظور ذخیره اطلاعات در جدول Places ایجاد شده و متدهای updatePlaces و deletePlaces به ترتیب برای ویرایش و حذف اطلاعات از جدول مذکور ایجاد شده اند.

Tourproject.HOTEL.ServicePlaces : در این کلاس 3 متد به منظور ذخیره و ویرایش و حذف اطلاعات با فراخوانی و شی سازی از کلاس Repository و متدهای آن ایجاد شده است . متد save با شی ساختن از کلاس Repository و فراخوانی متد insertPlace ، متد edit با شی ساختن از کلاس Repository و فراخوانی متد updatePlace و متد delete با شی ساختن از کلاس Repository و فراخوانی متد deletePlace به ترتیب کار ذخیره و ویرایش و حذف اطلاعات را انجام میدهند.

Tourproject.HOTEL.EntityTour : دارای 9 متغیر از نوع String , long است و هرکدام را با استفاده از Encapsulation و متدهای getter و setter گرفته و ست کرده ایم. این کلاس و متدهایش در کلاس های Repository و Service متعلق به پکیج خودش فراخوانی شده و مورد استفاده قرار میگیرد.

Tourproject.HOTEL.RepositoryTour : در این کلاس از 5 کلاس استفاده شده است. که اولی constructor است و در بدنه اش متد connect فراخوانی شده است. متد connect به منظور اتصال و

ارتباط برقرار کردن با دیتابیس `traveltour` ایجاد شده است. متد `insertTour` به منظور ذخیره اطلاعات در جدول `Tour` ایجاد شده و متدهای `updateTour` و `deleteTour` به ترتیب برای ویرایش و حذف اطلاعات از جدول مذکور ایجاد شده اند.

`Tourproject.HOTEL.ServiceTour`: در این کلاس 3 متد به منظور ذخیره و ویرایش و حذف اطلاعات با فراخوانی و شی سازی از کلاس `Repository` و متدهای آن ایجاد شده است. متد `save` با شی ساختن از کلاس `Repository` و فراخوانی متد `insertTour`، متد `edit` با شی ساختن از کلاس `Repository` و فراخوانی متد `updateTour` و متد `delete` با شی ساختن از کلاس `Repository` و فراخوانی متد `deleteTour` به ترتیب کار ذخیره و ویرایش و حذف اطلاعات را انجام میدهند.

`Tourproject.HOTEL.EntityVehicle`: دارای 3 متغیر از نوع `int`، `String` است و هرکدام را با استفاده از `Encapsulation` و متدهای `getter` و `setter` گرفته و ست کرده ایم. این کلاس و متدهایش در کلاس های `Repository` و `Service` متعلق به پکیج خودش فراخوانی شده و مورد استفاده قرار میگیرد.

`Tourproject.HOTEL.RepositoryVehicles`: در این کلاس از 5 کلاس استفاده شده است. که اولی `constructor` است و در بدنه اش متد `connect` فراخوانی شده است. متد `connect` به منظور اتصال و ارتباط برقرار کردن با دیتابیس `traveltour` ایجاد شده است. متد `insertVehicles` به منظور ذخیره اطلاعات در جدول `Vehicles` ایجاد شده و متدهای `updateVehicles` و `deleteVehicles` به ترتیب برای ویرایش و حذف اطلاعات از جدول مذکور ایجاد شده اند.

`Tourproject.HOTEL.ServiceVehicles`: در این کلاس 3 متد به منظور ذخیره و ویرایش و حذف اطلاعات با فراخوانی و شی سازی از کلاس `Repository` و متدهای آن ایجاد شده است. متد `save` با شی ساختن از کلاس `Repository` و فراخوانی متد `insertVehicle`، متد `edit` با شی ساختن از کلاس `Repository` و فراخوانی متد `updateVehicle` و متد `delete` با شی ساختن از کلاس `Repository` و فراخوانی متد `deleteVehicle` به ترتیب کار ذخیره و ویرایش و حذف اطلاعات را انجام میدهند.

View.LoginForm: اولین پنجره ای که نمایش داده میشود در این کلاس با رابط کاربری swing نمایش داده شده است. در این پنجره با وارد کردن یوزرنیم و پسورد admin میتوانیم وارد پنجره بعدی که توسط کلاس MainForm طراحی شده است شویم. در این کلاس از کامپوننت هایی مانند textfield برای وارد کردن اطلاعات و jbutton برای وارد شدن به پنجره بعدی استفاده شده است.

View.MainForm: دومین پنجره که نمایش داده میشود در این کلاس طراحی شده است. در این پنجره 5 دکمه وجود دارد که با انتخاب هر کدام طبق نیازمان به پنجره بعدی متعلق به آن انتقال داده میشویم.

View.HotelForm: در این کلاس پنجره مربوط به امور هتل ها نمایش داده میشود که دارای 3 پنل ثبت و ویرایش و حذف هتل است. و با گرفتن اطلاعات مورد نظر توسط ادمین و با استفاده از کلاسهای service و entity این کار را انجام میدهد. این کلاس دارای کامپوننت های همچون jbutton , jtabbedpane , jpanel , jlabel میباشد.

View.PeopleForm: در این کلاس پنجره مربوط به امور مسافران نمایش داده میشود که دارای 3 پنل ثبت و ویرایش و حذف هتل است. و با گرفتن اطلاعات مورد نظر توسط ادمین و با استفاده از کلاسهای service و entity این کار را انجام میدهد. این کلاس دارای کامپوننت های همچون jbutton , jtabbedpane , jpanel میباشد.

View.PlaceForm: در این کلاس پنجره مربوط به امور مکان های دیدنی نمایش داده میشود که دارای 3 پنل ثبت و ویرایش و حذف هتل است. و با گرفتن اطلاعات مورد نظر توسط ادمین و با استفاده از کلاسهای service و entity این کار را انجام میدهد. این کلاس دارای کامپوننت های همچون jbutton , jtabbedpane , jpanel میباشد.

View.TourForm: در این کلاس پنجره مربوط به امور تورها نمایش داده میشود که دارای 3 پنل ثبت و ویرایش و حذف هتل است. و با گرفتن اطلاعات مورد نظر توسط ادمین و با استفاده از کلاسهای service و

entity این کار را انجام میدهد . این کلاس دارای کامپوننت های همچون , jtabbedPane , jButton , jlabel , jpanel , JComboBox می باشد . کامپوننت JComboBox به منظور انتخاب انواع تور از بین داخلی، خارجی، زیارتی، سیاحتی و بهاره، تابستانه، پاییزه، زمستانه ایجاد شده است.

View.VehicleForm: در این کلاس پنجره مربوط به امور وسایل نقلیه نمایش داده میشود که دارای 3 پنل ثبت و ویرایش و حذف هتل است. و با گرفتن اطلاعات مورد نظر توسط ادمین و با استفاده از کلاسهای service و entity این کار را انجام میدهد . این کلاس دارای کامپوننت های همچون , jButton , jtabbedPane , jpanel میباشد.

Controller.LoginForm: در این کلاس متد main را ایجاد کرده و با شی ساختن و فراخوانی متد هم نامش در پکیج View آن را قابل اجرا کرده ام. و به وسیله کلاس setVisible() پنجره مورد نظر قابل رویت شده است.

Controller.MainForm: در این کلاس متد main را ایجاد کرده و با شی ساختن و فراخوانی متد هم نامش در پکیج View آن را قابل اجرا کرده ام. و به وسیله کلاس setVisible() پنجره مورد نظر قابل رویت شده است.

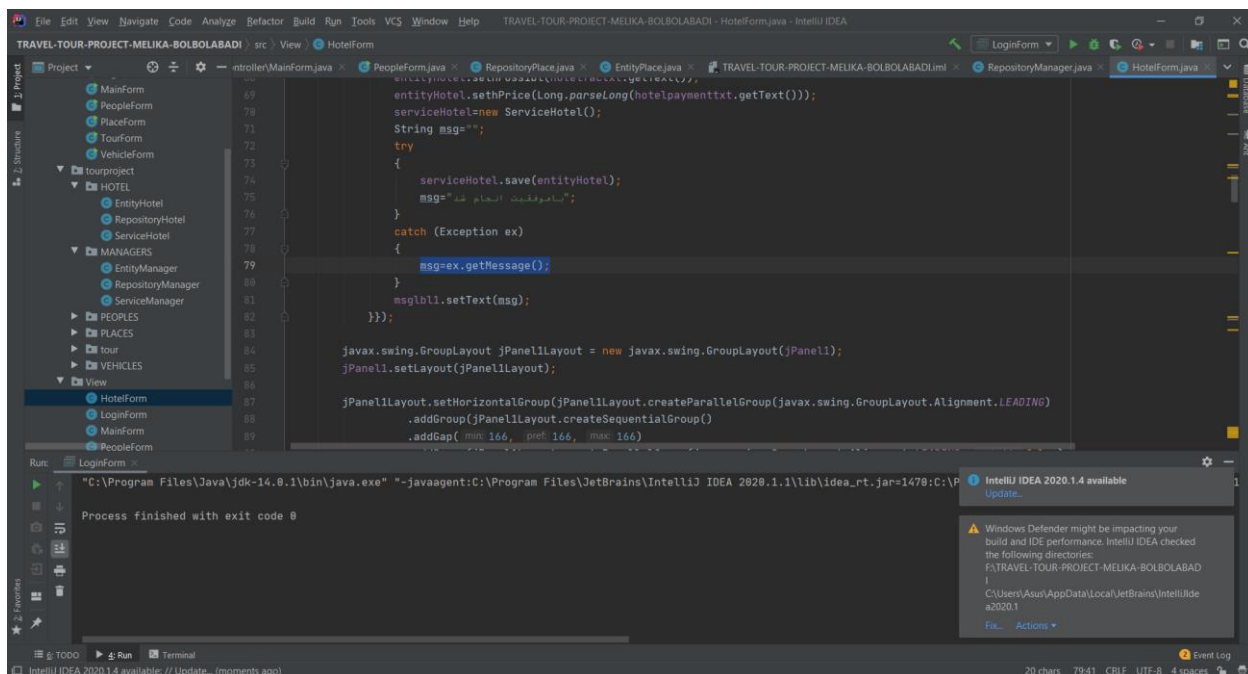
Controller.HotelForm: در این کلاس متد main را ایجاد کرده و با شی ساختن و فراخوانی متد هم نامش در پکیج View آن را قابل اجرا کرده ام. و به وسیله کلاس setVisible() پنجره مورد نظر قابل رویت شده است.

Controller.PeopleForm: در این کلاس متد main را ایجاد کرده و با شی ساختن و فراخوانی متد هم نامش در پکیج View آن را قابل اجرا کرده ام. و به وسیله کلاس setVisible() پنجره مورد نظر قابل رویت شده است.

Controller.PlaceForm: در این کلاس متد **main** را ایجاد کرده و با شی ساختن و فراخوانی متد هم نامش در پکیج **View** آن را قابل اجرا کرده ام. و به وسیله کلاس **setVisible()** پنجره مورد نظر قابل رویت شده است.

Controller.TourForm: در این کلاس متد **main** را ایجاد کرده و با شی ساختن و فراخوانی متد هم نامش در پکیج **View** آن را قابل اجرا کرده ام. و به وسیله کلاس **setVisible()** پنجره مورد نظر قابل رویت شده است.

Controller.VehicleForm: در این کلاس متد **main** را ایجاد کرده و با شی ساختن و فراخوانی متد هم نامش در پکیج **View** آن را قابل اجرا کرده ام. و به وسیله کلاس **setVisible()** پنجره مورد نظر قابل رویت شده است.



admin

admin

ورود

امور هتل

امور مسافران

امور مکان های دیدنی

امور وسایل نقلیه

امور تور

افزودن تور

حذف تور

ویرایش تور

کد تور را وارد کنید

نام تور را وارد کنید

بازه زمانی تور را وارد کنید

وسیله نقلیه تور را وارد کنید

کشافراد ثبت نامی در تور را وارد کنید

مبلغ کل تور را وارد کنید

مدیر تور را وارد کنید

نام هتل را وارد کنید

نام مکان های دیدنی را وارد کنید

نوع تور را انتخاب کنید

دسته بندی تور را انتخاب کنید

دسته بندی تور را انتخاب کنید

بهاره

تابستانه

پاییزه

زمستانه

اطلاعات را با علامت جدا کنید