

به نام خدا

توجه : این برنامه دارای ده درس و سه محل می باشد که هر کدام از درس ها دو روز برگزار می گردد به این معنی که در هر روز چهار درس و سه محل در دسترس می باشد که این برنامه محل هر درس با توجه به نیاز های آن را چه به صورت دستی و چه به صورت اتوماتیک چیدمان می کند.

می دانیم که تعداد دروس و تعداد محل ها نسبت به یک دانشگاه بسیار کم است اما با همین تعداد کم هم قابلیت های این برنامه مشخص می شود .

لازم به ذکر است که از برنامه کنترل ورژن گیت برای توسعه ی این برنامه استفاده شده است در واقع استراتژی توسعه برنامه ما به سه شکل واقع شد ، دو شکل آن به صورت جدا جدا که برای عمل پوشاندن به آن از قابلیت برنچ گیت استفاده شد و روش سوم هم با هم فکری هم زمان کد زدن بود .

لینک گیت: <https://github.com/kok-sadra/University-program-system-simulation>

همچنین شایان ذکر است که برای توسعه این برنامه از دو IDE که یکی **clion** و دیگری **vs code** استفاده شده است .

این نکته لازم به ذکر است که ما در حال توسعه قسمت **QT5** هستیم اما به دلیل کمبود زمان کافی و همچنین گسترده بودن کتابخانه های **QT** و یادگیری آن متأسفانه نتوانستیم **Qt** را به صورت حرفه ای و قابل قبول برای این برنامه پیاده سازی کنیم .

این برنامه شامل چهار کلاس و هفت تابع گلوبال می باشد . که در حالت کلی وظیفه ی چیدمان برنامه دروس دانشگاهی برای استادان و دانشجویان را دارا می باشد به این صورت که با توجه به 5 ویژگی (ظرفیت و استاد و دانشجو و زمان و ویدئو پروژکتور) مکان درست را به هر درسی (به صورت دستی و اتوماتیک) تخصیص می دهد.

حال به سراغ توضیح تفسیر کد برای سهولت برداشت از برنامه و محاسبات و الگوریتم های استفاده شده می پردازیم :

دو کلاس **courseLocation** , **course** از کلاس **commonInformation** به دلیل وجود اطلاعات مشترک ارث بری کرده است ضمناً کلاس **commonInformation** یک کلاس انتزاعی است.

کلاس **commonInformation** :

هدف از ایجاد این کلاس ارث بری آشکار و چندریختی میان دو کلاس **courseLocation** , **course** می باشد .

در این کلاس به دلیل دسترسی یکسری توابع به متغیرهای **protected** از **friend** کردن توابع استفاده شده است.

id : شناسه هر درس و هر محل برای دسترسی به آن

capacity : ظرفیت هر درس و همچنین ظرفیت هر محل (تعداد دانشجویان هر درس و هر محل)

videoprojector : نیاز به داشتن ویدئو پروژکتور برای هر درس و همچنین دارا بودن ویدیوپروژکتور هر محل

inputInfo() : این تابع به صورت **pure virtual** اعلام شده است و در کلاس های فرزند بازتعریف خواهد شد (چند ریختی) و از این تابع برای خواندن اطلاعات از روی فایل استفاده می شود

کلاس **course** :

رویکرد کلی به این صورت است که در این کلاس اطلاعات مربوط به هر درس در متغیر هایی که به صورت عادی و ترکیب شی از کلاس های دیگر تعریف شده اند به کمک تابعی ذخیره می شوند

در این کلاس به دلیل دسترسی یکسری توابع به متغیر های **private** از **friend** کردن توابع استفاده شده است

name : نام هر درس

teachername : نام استاد هر درس

temporary : موقت یا دائم بودن آن درس را نشان می دهد (در اصل مربوط به درس های فوق برنامه می باشد)

studentList : برای ذخیره سازی شماره دانشجویی دانشجویان استفاده می شود که به صورت وکتور تعریف گشته است

location : شی ای از کلاس **courseLocation** می باشد که به صورت ترکیب در این کلاس استفاده شده و در اصل برای ذخیره سازی مشخصات محلی که آن درس در آن برگزار می گردد استفاده می شود

inputInformation(): این تابع اطلاعات مربوط به هر درس را از روی فایل میخواند

رویکرد این تابع به این گونه است که **id** درس مورد نظر را که قرار است اطلاعات آن را از روی فایل بخواند دریافت می کند سپس در فایلی که اطلاعات دروس در آن قرار داده شده **id** را با استفاده از یک حلقه **for** سرچ می کند و شماره خطی که آن **id** در آن قرار دارد را در متغیری ذخیره می کند در مرحله بعد با توجه به الگوی قراردادن اطلاعات در فایل، آن ها را دریافت کرده و در هر شی متناظر با متغیر های قرارداده شده در کلاس **course** ذخیره می کند

این نکته لازم به ذکر است که ما برای دریافت اطلاعات و ذخیره سازی آن ها در متغیر از توابعی مانند **getline()** برای خواندن خط به خط از فایل و همچنین از تابع **atoi()** برای تبدیل نوع داده و از **istream** نیز برای ذخیره سازی شماره دانشجویی دانشجویان که با یک فاصله از هم جدا گشته اند استفاده کرده ایم.

الگوی قرار گرفتن اطلاعات در فایل مربوط به دروس (**proj.txt**)

- 1- **id** درس
- 2- نام درس
- 3- نام استاد
- 4- نیاز به ویدئو پروژکتور
- 5- موقتی بودن درس
- 6- ظرفیت درس
- 7- ساعت شروع
- 8- دقیقه شروع
- 9- ساعت مدت برگزاری درس
- 10- دقیقه مدت برگزاری درس
- 11- روزهای برگزاری درس (هر درسی در دو روز از هفته برگزار می شود)
- 12- شماره دانشجویی دانشجویان

کلاس `courseLocation` :

رویکرد کلی به این صورت است که در این کلاس اطلاعات مربوط به هر محل در متغیر هایی به کمک تابعی ذخیره می شوند

در این کلاس به دلیل دسترسی یکسری توابع به متغیر های `private` از `friend` کردن توابع استفاده شده است

`inputInformation()` : این تابع اطلاعات مربوط به هر کلاس (محل) را از روی فایل میخواند

رویکرد این تابع به این گونه است که `id` محل مورد نظر را که قرار است اطلاعات آن را از روی فایل بخواند دریافت می کند سپس در فایلی که اطلاعات کلاس ها در آن قرار داده شده `id` را با استفاده از یک حلقه سرچ میکند و شماره خطی که آن `id` در آن قرار دارد را در متغیری ذخیره می کند در مرحله بعد با توجه به الگوی قرار دادن اطلاعات در فایل، آن ها را دریافت کرده و در هر شی متناظر با متغیر های قرارداده شده در کلاس `courseLocation` ذخیره می کند

در این تابع از یکسری توابع کمکی درون کتابخانه ها و همچنین تابعی مانند `stringToBool()` برای تبدیل نوع داده ها استفاده شده است

الگوی قرار گرفتن اطلاعات در فایل مربوط به کلاس ها (محل)

- 1- `id` کلاس
- 2- ظرفیت کلاس
- 3- دارای ویدئو پروژکتور

کلاس date :

رویکرد کلی این کلاس به این صورت است که اطلاعات مربوط به زمان برگزاری هر درس را در خود ذخیره می کند

در این کلاس به دلیل دسترسی یکسری توابع به متغیرهای private از friend کردن توابع استفاده شده است

HourStart : ساعت شروع

MinStart : دقیقه شروع

HourDuration : مدت زمان ساعت (مدت زمان برگزاری)

MinDuration : مدت زمان دقیقه (مدت زمان برگزاری)

Day1 : یکی از روزهای برگزاری درس

Day2 : روز دیگر برگزاری درس

st : این متغیر از نوع اعشاری است که در اصل زمان شروع را به صورت عدد اعشار در خود ذخیره می کند مثلاً: ساعت 5:45 را به صورت 5.45 درون خود ذخیره می کند

et : این متغیر از نوع اعشاری است که در اصل زمان پایان را به صورت عدد اعشار در خود ذخیره می کند مثلاً: ساعت 6:45 را به صورت 6.45 درون خود ذخیره می کند

توابع set_Day(), set_DM(), set_DH(), set_MS(), set_HS() هرکدام برای مقداردهی متغیرهای پرایوت این کلاس به کار می رود از این توابع در تابع inputInformation() در کلاس های course , courseLocation استفاده شده است

CalculateTime() : این تابع برای محاسبه ساعت پایانی هر درس با توجه به زمان شروع و مدت زمان برگزاری ایجاد شده که در نهایت زمان شروع و زمان پایان را در دو متغیر et, st ذخیره می کند

آرایه دو بعدی `week[7][4]` از نوع `course` به صورت گلوبال تعریف شده که در آن دروس طبق روز برگزاری ذخیره می شود که 7 نشان دهنده روز های هفته و 4 تعداد دروس در هر روز می باشد

تابع `sortDay()`: این تابع به عنوان ورودی تمامی دروس را به صورت آرایه گرفته و دروس را بر اساس روز، چیدمان و در آرایه دو بعدی `week` ذخیره می کند

تابع `weekShow()`: این تابع دروس را در همراه با یکسری از اطلاعاتشان که در آرایه `week` براساس روز مرتب شده اند را نمایش می دهد

تابع `writingTheFile()`: این تابع دروس را در همراه با یکسری از اطلاعاتشان که در آرایه `week` براساس روز مرتب شده اند را در فایلی به نام `plan` می نویسد و ذخیره می کند

تابع `specify()`: این تابع زمانی مورد استفاده قرار می گیرد که کاربر قصد دارد محل برگزاری هر درس را به صورت دستی انتخاب و وارد نماید

تابع `mothercheck()`: در این تابع موارد زیر بررسی می شود:

- 1- ظرفیت کلاس از ظرفیت درس بیشتر باشد در غیر اینصورت پیغام مناسب را چاپ می کند
- 2- درسی که نیاز به ویدیو پروژکتور دارد در کلاس دارای ویدئو پروژکتور باشد در غیر اینصورت پیغام مناسب را چاپ میکند و همچنین درسی که ویدیو پروژکتور ندارد مهم نیست در کلاسی باشد که ویدیو پروژکتور دارد
- 3- بررسی می کند که دو درس یا چند درس با `id` های یکسانی تعریف نشده باشند در صورت وجود دو یا چند درس با `id` های یکسان پیغام مناسب را چاپ می کند

همچنین در این تابع ، تابع `babycheck()` نیز فراخوانی می شود که در بند بعد نحوه عملکرد آن را توضیح می دهم

تابع babycheck(): در این تابع موارد زیر بررسی می شود:

- 1- در محل یکسان و در زمان مشخصی دو درس در حال برگزاری نباشد در غیر اینصورت پیغام مناسب را چاپ میکند
- 2- در زمان مشخصی یک استاد برای دو درس قرار داده نشده باشد در غیر اینصورت پیغام مناسب را چاپ میکند
- 3- با استفاده از تابع studentcheck() بررسی می کند که در زمان مشخصی یک دانشجو در دو کلاس حضور نداشته باشد در غیر اینصورت پیغام مناسب را چاپ می کند

تابع studentcheck(): در این تابع مورد زیر بررسی میشود:

- 1- در زمان مشخصی یک دانشجو در دو کلاس حضور نداشته باشد در غیر اینصورت پیغام مناسب را چاپ می کند

تابع automaticLocationDetermination(): این تابع زمانی مورد استفاده قرار میگیرد که برنامه به صورت خودکار محل برگزاری(کلاس) هر درس را مشخص نماید

این تابع به این گونه عمل میکند:

در ابتدا محل را با توجه به پیش نیاز های درس که همان ظرفیت کلاس و داشتن ویدئو پروژکتور است پیدا میکند سپس با استفاده از تابع checkroom () در بین تمام درس های موجود در آن روز به دنبال درسی میگردد که محل انتخاب شده توسط برنامه با محل برگزاری آن درس یکسان باشد حال اگر پیدا کرد ایندکس مربوط به آن درس را که در ارایه week ذخیره شده بود برمیگرداند در غیر اینصورت یعنی اگر درسی پیدا نکرد که محل برگزاری اش با محل برگزاری انتخاب شده توسط برنامه یکسان باشد مقدار پیش فرضی که همان 10 است را برمیگرداند

در ادامه تابع automaticLocationDetermination () بررسی میشود اگر درسی پیدا شد که محل برگزاری اش با محل پیشنهادی برنامه یکسان بود تداخل زمانی را بین دو درس پیدا شده و درسی که برنامه قصد انتخاب محل را برای آن دارد چک میکند که اگر تداخل زمانی وجود نداشت آن محل را به درس مورد نظر اختصاص دهد و اگر تداخل زمانی وجود داشت نمی توان محل را به درس اختصاص داد چون که در زمانی که درس مورد نظر برگزار میشود درس دیگری نیز در آن محل در حال برگزاری است و در این صورت برنامه پیغام مناسب را چاپ خواهد کرد

برای اجرا گرفتن از برنامه روند زیر را پیش بگیرید

```
run program :  
  
you want to determine the location of each course yourself (enter 1)| the program specifies the location of each course (enter 0)  
1
```

زمانی که روی دکمه ی اجرا می زنیم این پیام برای ما نمایش داده می شود

برای وارد کردن محل هر درسی به صورت دستی عدد 1 و به صورت اتوماتیک عدد 0 را وارد کنید

روند پیش رو برای وارد کردن محل هر درس به صورت دستی که توسط کاربر انجام می شود :

```
you want to determine the location of each course yourself (enter 1)| the program specifies the location of each course (enter 0)  
1  
Courses : AdvanceProgramming , Math , Physics , Workshop , Differential , Statistics , History , DataStructure , LogicalCircuit , DiscreteMath  
Locations : 401 , 402 , 501  
1 : to determine the classroom , enter the desired course : Math  
enter the desired classroom for the Math course : 402  
2 : to determine the classroom , enter the desired course : History  
enter the desired classroom for the History course : 401  
3 : to determine the classroom , enter the desired course : AdvanceProgramming  
enter the desired classroom for the Advance Programming course : 501  
4 : to determine the classroom , enter the desired course : DataStructure  
enter the desired classroom for the DataStructure course : 401  
5 : to determine the classroom , enter the desired course : Differential  
enter the desired classroom for the Differential course : 501  
6 : to determine the classroom , enter the desired course : Statistics  
enter the desired classroom for the Statistics course : 402  
7 : to determine the classroom , enter the desired course : DiscreteMath  
enter the desired classroom for the DiscreteMath course : 501  
8 : to determine the classroom , enter the desired course : Physics  
enter the desired classroom for the Physics course : 402  
9 : to determine the classroom , enter the desired course : Workshop  
enter the desired classroom for the Workshop course : 401  
10 : to determine the classroom , enter the desired course : LogicalCircuit  
enter the desired classroom for the LogicalCircuit course : 501
```

همانطور که در تصویر بالا مشاهده می کنید برنامه از شما میخواهد که درس مورد نظر را وارد و محل دلخواه خود را برای آن درس وارد کنید

توجه :در وارد کردن نام دروس به کوچک و بزرگ بودن حروف و همچنین بدون فاصله بودن آن توجه کنید

بعد از این مرحله برنامه وجود تداخل ها و پیش نیاز ها برای انتخاب محل را به شما گزارش می دهد

```
The capacity of AdvanceProgramming is greater than the capacity of the location
AdvanceProgramming needs a video projector and location have not a video projector
Workshop needs a video projector and location have not a video projector
The capacity of Differential is greater than the capacity of the location
DataStructure needs a video projector and location have not a video projector
LogicalCircuit needs a video projector and location have not a video projector
The capacity of DiscreteMath is greater than the capacity of the location
DiscreteMath needs a video projector and location have not a video projector
-----
AdvanceProgramming and LogicalCircuit have time overlap on 2of the week
AdvanceProgramming and LogicalCircuit have time teacher interference on 2of the week
Math and DataStructure have time teacher interference on 5of the week
student number1is present in both AdvanceProgramming and Workshop courses and there is a time overlap
-----
```

طبق تصویر بالا برنامه :

برای ظرفیت و همچنین ویدیوپروژکتور درس **AdvanceProgramming** که با محل تخصیص داده شده به آن دچار مشکل می شود پیغام نمایش داده است

برای ویدیوپروژکتور درس **Workshop** که با محل تخصیص داده شده به آن دچار مشکل می شود پیغامی نمایش داده است

همینطور نیز پیغام هایی برای درس های **DataStructure , LogicalCircuit , DiscreteMath** نمایش داده است

در ادامه برنامه برای تداخل زمان میان دو درس **AdvanceProgramming , LogicalCircuit** که هر دو در کلاس 501 در روز دوم هفته یعنی یکشنبه برگزار می شوند پیغامی نمایش داده است

همچنین برنامه برای استاد درس **AdvanceProgramming , LogicalCircuit** به دلیل حضور در این دو درس که باهم تداخل زمانی دارند پیغامی نمایش داده است

همچنین برنامه برای دانشجویی با شماره دانشجویی 1 به دلیل حضور در دو درس **AdvanceProgramming , Workshop** , که از نظر زمانی با هم در تداخل اند پیغامی نمایش داده است

در آخر برنامه یک نمای کلی از دروس که براساس روز مرتب شده اند همراه با یک سری از اطلاعات دروس ر در ترمینال برنامه ا نمایش میدهد و همچنین این برنامه هفتگی را هم در فایل **plan.txt** می نویسد

روند پیش رو برای تخصیص محل برای هر درس به صورت اتوماتیک است :
در این حالت برنامه محل ها را به دروس اختصاص می دهد اما اگر نتواند به هر دلیلی (در تابع **automaticLocationDetermination()** فرایند به صورت کامل توضیح داده شده) پیامی مناسب آن دلیل چاپ می کند در مرحله ی آخر تابع **babycheck()** که در آن تداخل زمانی میان دو درس و همچنین تداخل برای استاد و تداخل برای دانشجویی می شود را فراخوانی می کند و پیغام های مربوط به آن تابع را چاپ می کند.

```
you want to determine the location of each course yourself (enter 1)| the program specifies the location of each course (enter 0)
0
program could not assign a location to the Workshop course Because the program considers the place of Workshop course in class402, which has a time conflict with the AdvanceProgramming course that is held in class 402.(on tuesday or saturday)
program could not assign a location to the LogicalCircuit course Because the program considers the place of LogicalCircuit course in class402, which has a time conflict with the AdvanceProgramming course that is held in class 402.(on sunday or monday)
program could not assign a location to the Differential course Because the program considers the place of Differential course in class402, which has a time conflict with the Physics course that is held in class 402.(on monday or wednesday)
-----
AdvanceProgramming and LogicalCircuit have time teacher interference on 2of the week
Workshop and DiscreteMath have time overlap on 4of the week
Math and DataStructure have time teacher interference on 5of the week
student number1is present in both AdvanceProgramming and Workshop courses and there is a time overlap
-----
```

در آخر برنامه یک نمای کلی از دروس که براساس روز مرتب شده اند همراه با یک سری از اطلاعات دروس ر در ترمینال برنامه نمایش میدهد و همچنین این برنامه هفتگی را هم در فایل **plan.txt** می نویسد