

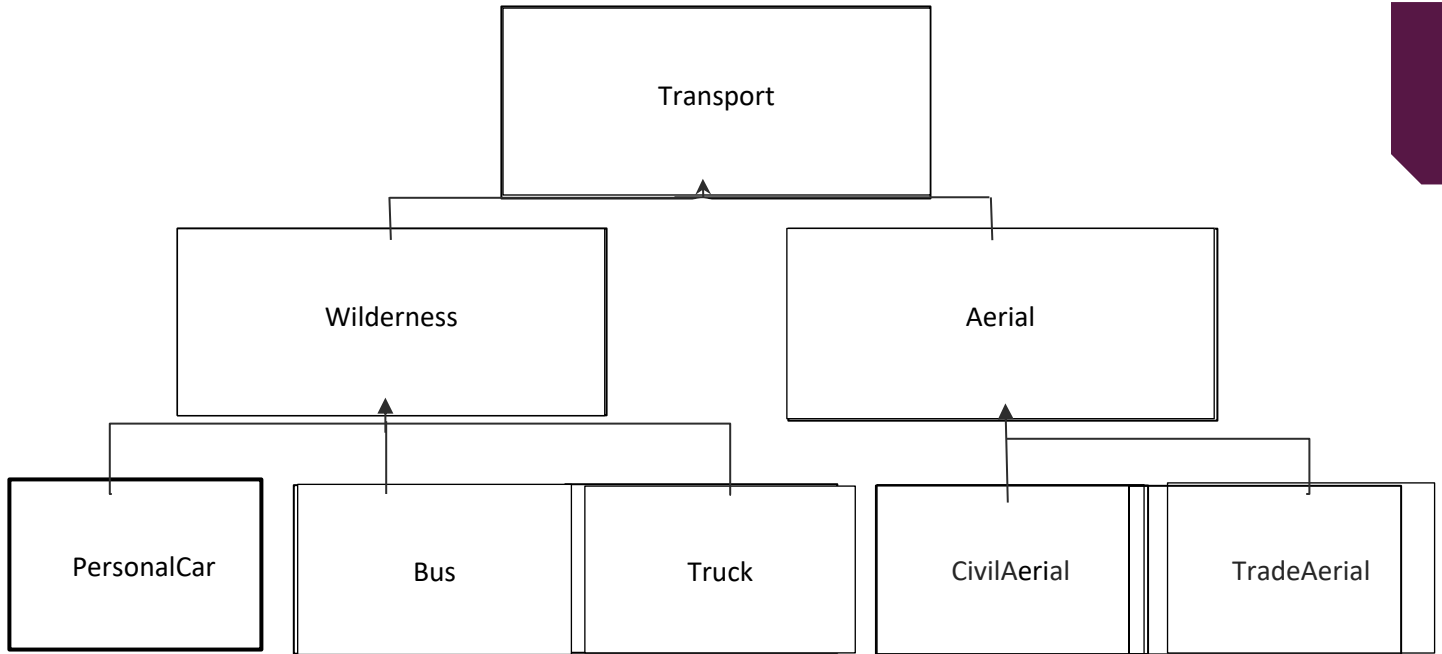
Damascus University
F.I.T.E

PROGRAMMING 3
PROJECT REPORT

Done By:
Houmam Al-Bazzal
Bker Kazmoz
Bishr Al-Khoja

ENG. MANAR AL-KASEM

Classes Diagram & Inheritance



الصف Transport : هو الصف الأب للصفين Wilderness و Aerial وهو يعبر عن أي وسيلة نقل محتملة

الصف Wilderness : هو الصف الأب لصفوف وسائل النقل البرية PersonalCar و Bus و Truck

الصف Transport : هو الصف الأب لصفوف وسائل النقل الجوية CivilAerial و TradeAerial

-1 Transport Class :

- المتحولات:

- SN: من نوع int يعبر عن رقم الوسيلة الخاص
- Speed: من نوع int يعبر عن السرعة القصوى للوسيلة
- minWight: من نوع Double يعبر عن وزن الوسيلة الفارغ
- maxWight: من نوع Double يعبر عن وزن الوسيلة القائم (الفارغ + أقصى حمولة للوسيلة)
- Bker: يأخذ قيمه من 1 لـ 5 حيث كل قيمة تعبر عن وسيلة معينة مثال (1 يعبر عن السيارة و 2 يعبر عن الحافلة و 5 عن الطائرة التجارية و 4 عن الطائرة المدنية و 3 عن الشاحنة)
- Length: من نوع Double يعبر عن طول الوسيلة
- Brand: من نوع char[15] يعبر عن موديل الوسيلة أو الشركة الصانعة لها
- Type: من نوع Char ولها قيمتين (C لوسائل النقل المدنية و T التجارية)
- Category: من نوع Char ولها ثلاث قيم (I لوسائل النقل الدولية G الحكومية و)
- Mdate: من نوع Date يعبر عن تاريخ تسجيل السيارة

- التوابع والاجرائيات المستخدمة:

باني افتراضي وباني بالقيم وتوابع جالبة وتوابع تهيئة وتابع طباعة

-2 Wilderness: صف المركبات البرية

- المتحولات:
- engineVolume: من نوع int يعبر عن سعة المحرك
- الإجراءيات:
- إجرائية تهيئة وجلب سعة المحرك وإجرائية طباعة لكافة المعلومات

-3 Aerial: صف المركبات الجوية:

- المتحولات:
- numOfEngines: من نوع int يعبر عن عدد محركات الطائرة
- الإجراءيات:
- إجرائية تهيئة وجلب عدد محركات الطائرة وإجرائية طباعة لكافة المعلومات

-4 CivilAerial: صف الطائرات المدنية

- المتحولات:
- Sets: من نوع int يدل على عدد الكراسي في الطائرة
- الإجراءيات:
- إجرائيات جالبة ومهيئة وإجرائية طباعة لبيانات الطائرة

-5 TradeAerial: صف الطائرات التجارية

- المتحولات:
- نفس متحولات الصف الأب Aerial
- الإجراءيات:
- نفس إجرائيات الصف الأب

-6 PersonalCar: صف السيارات المدنية

- المتحولات:
- numofdoors: من نوع int يدل على عدد الأبواب في السيارة
- الإجراءيات:
- إجرائيات جالبة ومهيئة وإجرائية جالبة لعدد الكراسي وإجرائية طباعة لبيانات السيارة

-7 Bus: صف الحافلات

- المتحولات:
- Sets: من نوع int يدل على عدد الكراسي في الحافلة
- الإجراءيات:
- إجرائيات جالبة ومهيئة وإجرائية طباعة لبيانات الحافلة

-8 Truck: صف الشاحنات التجارية

- المتحولات:
- high: من نوع double يدل على ارتفاع قسم البضائع
- الإجراءيات:
- إجرائيات جالبة ومهيئة وإجرائية طباعة لبيانات الشاحنة

ويوجد أيضا الصف **Journey** المعبر عن الرحلات وهو مستقل عن باقي الصفوف بالإضافة للصف Date المعبر عن بنية التاريخ والصف Seat المعبر عن مقعد محجوز في رحلة

9-Journey: صف الرحلات

- المتحولات:

- SN: من نوع int يعبر عن رقم الوسيلة الخاص
 - journeyNumber: من نوع int يعبر عن رقم الرحلة الخاص
 - seatnum: من نوع int يعبر عن عدد المقاعد المحجوزة لهذه الرحلة
 - fullseat: من نوع int يعبر عن عدد مقاعد الوسيلة الكاملة
 - transportType: من نوع char يعبر عن نوع وسيلة النقل (حكومية/خاصة/هيئات دولية)
 - date: من نوع Date يعبر عن تاريخ الرحلة
 - wightWrning: من نوع bool يعبر عن وجود تحذير للوزن ام لا
 - from: من نوع مصفوفة char يعبر عن نقطة الانطلاق
 - to: من نوع مصفوفة char يعبر عن نقطة الوصول
 - curWight: من نوع double يعبر عن وزن البضائفة المحملة بالإضافة الى وزن الوسيلة الفارغ
 - maxWight: من نوع double يعبر عن الوزن الأقصى للوسيلة
 - minWight: من نوع double يعبر عن الوزن الفارغ للوسيلة
 - seats: من نوع مصفوفة Seat ديناميكية يعبر عن المقاعد المحجوزة ووزن امتعة كل مقعد
- الإجراءيات:
- اجرائيات جالبة ومهيئة لجميع المتحولات السابقة حسب أسمائها بالإضافة الى إجرائية طباعة لمعلومات الرحلة كاملة

10-Seat: صف مقاعد الرحلات

- المتحولات:

- numberOfSeat: من نوع int عن رقم الكرسي
 - Book: من نوع bool يعبر عن وجود حجز أم لا في هذا المقعد
 - w_bag: من نوع double يعبر عن وزن امتعة هذا الكرسي
- الاجرائيات:
- اجرائيات جالبة ومهيئة لمتحولات الصف واجرائية طباعة لمعلومات لعنصر هذا الصف

11-Date: صف التاريخ

- المتحولات:

- Day: من نوع int يعبر عن اليوم
 - Month: من نوع int يعبر عن الشهر
 - Year: من نوع int يعبر عن السنة
- الاجرائيات:
- تحميل زائد لعمليات المقارنة بين تاريخين من نفس الصف

الإجراءات المستخدمة:

1- Search transport:

Return type: Transport*

وهو تابع يقوم بالبحث عن وسيلة نقل معينة حسب رقمها و إرجاع بياناتها في حال وجودها

نقوم بإرسال رقم الوسيلة كبارامتر

نبدأ بالبحث في linkedlist الخاص بوسائل النقل بدءاً من ال head بمقارنة رقم الوسيلة المدخل مع رقم الوسيلة الخاص بعنصر السلسلة المعين وفي حال عدم وجوده نكمل البحث مروراً على كل السلسلة وعند وجوده نقوم بإرجاع بيانات الوسيلة المرادة

2- Search journey:

Return type: int

وهو تابع يقوم بالبحث عن رحلة معينة وإرجاع رقم الرحلة في حالة وجودها

نبدأ بالبحث في linkedlist الخاص بالرحلة بدءاً من ال head بمقارنة رقم الرحلة المدخل مع رقم الرحلة الخاص بعنصر السلسلة المعين وفي حال عدم وجوده نكمل البحث مروراً على كل السلسلة وعند وجوده نقوم بإرجاع رقم الرحلة

3- CheackAvilabilite:

Return type: bool

وهو تابع يقوم بالتأكد من عدم وجود رحلة لوسيلة ما في تاريخ معين

نقوم بإرسال رقم الوسيلة والتاريخ المدخل كبارامترات

نبدأ من ال head في linkedlist الخاص بوسائل النقل بمقارنة رقم الرحلة وتاريخ الرحلة مع الرقم والتاريخ المرسلين كبارامتر وفي حال عدم وجوده نكمل البحث مروراً على كل السلسلة وعند وجوده نقوم بإرجاع true

4- Printinformation:

Return type: void

وهو تابع يقوم بطباعة بيانات جميع الرحلات لوسيلة نقل معينة

نبدأ بإدخال رقم الوسيلة المراد طباعة بيانات رحلاتها ثم نبدأ بالبحث في linkedlist الرحلة بدءاً من ال head بمقارنة رقم الرحلة المدخل مع رقم الوسيلة لعنصر السلسلة وفي حال عدم وجوده نكمل البحث مروراً على كل السلسلة وعند وجوده نقوم باستدعاء تابع طباعة البيانات print() الخاص ب linkedlist الرحلة

5- Printnumber:

Return type: void

وهو تابع يقوم بطباعة أرقام الرحلات لوسيلة نقل معينة بين مدينتين محددتين

نقوم بإدخال رقم الوسيلة المرادة ثم نقوم باستدعاء تابع Search transport للبحث عن وجود هذه الوسيلة ثم نقوم بإدخال مكان بداية الرحلة والوجهة ومقارنة رقم الرحلة و بداية الرحلة والوجهة بعناصر linkedlist الخاص بالرحلة وفي حال وجوده يتم طباعة رقم الرحلات لهذه الوسيلة

6- calculatePassengerNumber:

return type: void

وهو تابع يقوم بحساب عدد الركاب الذين سافروا على متن وسيلة ما خلال فترة زمنية محددة

بداية نقوم بإدخال رقم الوسيلة ثم نبحث عنها عن طريق تابع search transport وبعدها ندخل تاريخ البداية وتاريخ النهاية للفترة الزمنية المراد الحساب عندها مع معالجة كافة الإدخالات الخاطئة ثم نقوم بالبحث داخل linkedlist الرحلة عن رقم الوسيلة و التأكد من وجود رحل في الفترة المحددة بمقارنة تاريخ الرحل مع تاريخ البداية والنهاية

وعند وجود اكثر من رحلة في هذا الوقت نقوم بحساب عدد الركاب الذين كانوا على متن الوسيلة في هذه الفترة الزمنية

7- search_journey_for_add:

return type: int

وهو تابع يقوم بالتأكد من عدم وجود رحلتين تحملان نفس الرقم

يعتمد هذا التابع على شرط أساسي وهو مقارنة الرقم المدخل مع أرقام الرحلات الموجودة لدينا ويتم عبر linkedlist الخاص بالرحلة فعند عدم وجود رحلة بهذا الرقم يقوم بإرجاع هذا الرقم وإلا يقوم بإرجاع 1-

8- newTransport():

return type: void

و هو تابع يقوم بإضافة وسيلة نقل جديدة الى بنية المعطيات الموجودة وذلك وفق خطوات معينة على الشكل :

يتم ادخال نوع الوسيلة المراد إدخالها وفق قائمة يختار منها المستخدم نوع الوسيلة لياخذ معامل ضمن الصف (يدعى bker) قيمة الخيار المدخل ويحدد نوع الوسيلة ليطلب من المستخدم ادخال معاملاتها المناسبة وفق

- bker=1 : أي نوع الوسيلة المدخلة هي سيارة ركاب

- bker=2 : أي نوع الوسيلة المدخلة هي حافلة ركاب

- bker=3 : أي نوع الوسيلة المدخلة هي شاحنة بضائع

- bker=4 : أي نوع الوسيلة المدخلة هي طائرة ركاب

- bker=5 : أي نوع الوسيلة المدخلة هي طائرة بضائع

يتم ادخال رقم الوسيلة (SN) والذي يميز الوسيلة عن غيرها فهو رقم وحيد لا يتكرر

يتم ادخال تاريخ صنع الوسيلة (mdate من كلاس date) مع مراعاة الدخال الصحيح (فالايام بين 0 و 30 و الأشهر بين 1 و 12 والسنوات بين 1950 و 2019) حيث يحقق ذلك ضمان عدم ادخال ارقام عشوائية او تواريخ غير منطقية

رايعا يتم ادخال السرعة العظمى للوسيلة (speed) و طولها (leanght) و نوعها (brand) و ملكيتها (category) وفق قائمة "حكومية او دولية او خاصة " وهي معاملات عامة تشترك فيها جميع الوسائل

هنا يتم التفصيل في ادخال المعاملات حسب نوع الوسيلة

من اجل السيارة(bker=1)

يتم ادخال عدد الأبواب (doors) و الذي من خلاله نعرف عدد مقاعد السيارة وفق المعادلة (Doors-1)

ندخل الوزن الفارغ (minweight) و الوزن القائم (maxweight) يجب ان يحقق على الأقل ((minWight + (25*(Doors-1)))

ندخل سعة المحرك (engineVolume)

ننشأ اوبجيكت بالمعاملات السابقة من الكلاس personalcar و نضيفه الى linckedlist من نوع transportlist

من اجل الحافلة (bker=2)

ادخال عدد المقاعد للحافلة (seats) ثم نقوم بمعالجتها لتكون منطقية (160<-8)

ندخل الوزن الفارغ (minweight)

و الوزن القائم (maxweight) يجب ان يحقق على الأقل ((minWight + (25*Seats)

ندخل سعة المحرك (engineVolume)

ننشأ اوبجيكت بالمعاملات السابقة من الكلاس bus و نضيفه الى linckedlist من نوع transportlist

من اجل الشاحنة (bker=3)

ندخل ارتفاع البضائع (High من 1 الى 5)

ندخل الوزن الفارغ (minweight) و الوزن القائم (maxweight) بشكل منطقي

ننشأ اوبجيكت بالمعاملات السابقة من الكلاس truck و نضيفه الى linckedlist من نوع transportlist

من اجل طائرة الركاب (bker=4)

ندخل عدد المقاعد (seats) بشكل منطقي (200<-20)

ندخل الوزن الفارغ (minweight)

و الوزن القائم (maxweight) يجب ان يحقق ((minWight + (25*Seats)

وندخل عدد المحركات (numberOfEngine)

ثم ننشأ اوبجيكت بالمعاملات السابقة من الكلاس civilaerial و نضيفه الى linckedlist من نوع transportlist

من اجل طائرة تجارية (bker=5)

ندخل عدد المحركات (numberOfEngine)

والوزن الفارغ (minweight) و الوزن القائم (maxweight) بشكل منطقي

ننشأ اوبجيكت وفق المعاملات السابقة من الكلاس tradeaerial و نضيفه الى linckedlist من نوع transportlist

9- newJourney()

return type: void

و هو تابع يقوم بإضافة وسيلة نقل جديدة الى بنية المعطيات الموجودة وذلك وفق خطوات معينة على الشكل :

نقوم بإدخال رقم الرحلة (JourneyNumber) و التحقق فيما اذا ما كان هذا الرقم مستخدماً من قبل من اجل رحلة أخرى ام لا

نقوم بإدخال رقم الوسيلة المطلوبة (SN) من اجل الرحلة و التحقق فيما اذا كانت موجودة ام لا

نحدد نوع الوسيلة وفق المعامل الخاص بهذه المهمة (bker) ونطبع نوعها للمستخدم

نقوم بإدخال مكان انطلاق (from) و وجهة الرحلة (to) والمقارنة بينهما من اجل ضمان ان الوجهة ليست نقطة الانطلاق نفسها (أي من مكان الى مكان اخر)

ندخل تاريخ الرحلة (date من كلاس date) ونضمن صحة الادخال ثم نقارن تاريخ الرحلة مع اليوم (today) حيث لا يمكن انشاء رحلة حدثت في الماضي و انتهت بل الرحلة يجب ان تنطلق على الأقل من اليوم التالي لانشائها وبذلك نكون قد ضمنا واقعية الرحلة و أيضا انها لا يمكن ان تكون بتاريخ يسبق تاريخ صنع الوسيلة (mdate) التي ستقوم بهذه الرحلة

بعدها نتحقق من الوسيلة ليس لديها أي رحلات في التاريخ المدخل من خلال التابع CheackAvilabilite والذي يأخذ رقم الوسيلة و تاريخ الرحلة كمعاملات و يرجع قيمة بوليانية توافق حالة الوسيلة

نحضر معلومات الوسيلة (Type , minWight , maxWight) وفق توابع get لنملاً ببيانات الرحلة وفقها

هنا تتم عمليات ادخال المعاملات وفق نوع الوسيلة :

من اجل السيارة (bker=1)

ندخل عدد المقاعد المراد حجزها (seatnum) و ذلك ضمن عدد المقاعد الكلية المتاحة (fullseat)

ثم نقوم بالاستعانة بحلقة for تدور بعدد المقاعد المراد حجزها

ننشأ مصفوفة ديناميكية (seats) لحجز المقاعد حيث ندخل رقم المقعد (sear يجب ان يكون ضمن العدد الكلي للمقاعد) و وزن حمولة صاحب المقعد (wightcar يجب ان تكون اقل من 25) ونتحقق اذا كان المقعد متاحاً ام محجوزاً فاذا كان محجوزاً نعيد الادخال والا نخزنه و نعطي المقعد قيمة بوليانية تساوي الواحد تبين انه قد حجز

ثم نجمع الوزن الفارغ مع وزن الحمولة و نخزنه بالوزن الحال (curWight) الذي يحقق انه اصغر او يساوي الوزن الاعظمي للسيارة (maxWight)

من اجل الحافلة (bker=2)

ندخل عدد المقاعد المراد حجزها (seatnum) و ذلك ضمن عدد المقاعد الكلية المتاحة (fullseat)

ثم نقوم بالاستعانة بحلقة for تدور بعدد المقاعد المراد حجزها

ننشأ مصفوفة ديناميكية (seats) لحجز المقاعد حيث ندخل رقم المقعد (se يجب ان يكون ضمن العدد الكلي للمقاعد) و وزن حمولة صاحب المقعد (wight يجب ان تكون اقل من 25) ونتحقق اذا كان المقعد متاحاً ام محجوزاً فاذا كان محجوزاً نعيد الادخال والا نخزنه و نعطي المقعد قيمة بوليانية تساوي الواحد تبين انه قد حجز

ثم نجمع الوزن الفارغ مع وزن الحمولة و نخزنه بالوزن الحال (curWight) الذي يحقق انه اصغر او يساوي الوزن الاعظمي للحافلة (maxWight)

ونتحقق اذا كان اصغر من الوزن الاعظمي للحافلة نتيج له ادخال حمولة تجارية زائدة قدرها (maxWight - curWight) فاذا زادت الحمولة عن الوزن المتاح نرسل warning ونعيد الادخال

من اجل الشاحنة (bker=3)

ندخل وزن الحمولة المراد نقلها في هذه الرحلة (wightload) ونتحقق اذا كانت ضمن مجال الحمولة المنطقية ام لا (- maxWight minWight) فاذا زادت الحمولة عن الوزن المتاح نرسل warning ونعيد الادخال من اجل طائرة ركاب (bker=4)

ندخل عدد المقاعد المراد حجزها (seatnum) و ذلك ضمن عدد المقاعد

الكلية المتاحة (fullseat)

نقوم بالاستعانة بحلقة for بعدد المقاعد المراد حجزها

ننشأ مصفوفة ديناميكية (seats) لحجز المقاعد حيث ندخل رقم المقعد (se) يجب ان يكون ضمن العدد الكلي للمقاعد) و وزن حمولة صاحب المقعد (wight) يجب ان تكون اقل من 25) ونتحقق اذا كان المقعد متاحا ام محجوزا فاذا كان محجوزا نعيد الادخال والا نخزنه و نعطي المقعد قيمة بوليانية تساوي الواحد تبين انه قد حجز

نجمع الوزن الفارغ مع وزن الحمولة و نخزنه بالوزن الحال (curWight) الذي يحقق انه اصغر او يساوي الوزن الاعظمي للحافلة (maxWight)

ونتحقق اذا كان اصغر من الوزن الاعظمي للحافلة نتيج له ادخال حمولة تجارية زائدة قدرها (maxWight - curWight) فاذا زادت الحمولة عن الوزن المتاح نرسل warning ونعيد الادخال

من اجل طائرة تجارية (bker=5)

ندخل وزن الحمولة المراد نقلها في هذه الرحلة (wightload) ونتحقق اذا كانت ضمن مجال الحمولة المنطقية ام لا (- maxWight minWight) فاذا زادت الحمولة عن الوزن المتاح نرسل warning ونعيد الادخال

نقوم بانشاء اوبجيكت وفق المعاملات المدخلة للرحلة و نقوم بتخزينه في linckrdlist من نوع journeyList