12/24/2022

Abdo Gamal Ahmed

no campany

Design pattern

Creation Design pattern



Table of Contents

[0 – topic in this slide : 2](#_Toc123069108)

[1-singleton: 2](#_Toc123069109)

[1.1- Implementation for singleton: 3](#_Toc123069110)

[1.2 – singlton strcture : 5](#_Toc123069111)

[1.3-applicability : 6](#_Toc123069112)

[1.4-problem in singleton. 6](#_Toc123069113)

[2-factory method : 6](#_Toc123069114)

[2.1 factory method step: 7](#_Toc123069115)

[2.2 example on factory method: 8](#_Toc123069116)

[2.3 –example2 on factory method : 9](#_Toc123069117)

[2.4-structure of factory method: 10](#_Toc123069118)

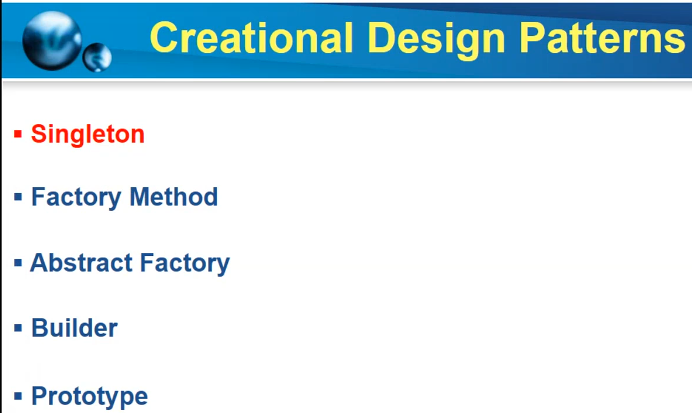
[2.5-applicability : 12](#_Toc123069119)

[2.6- pros and cons : 12](#_Toc123069120)

[2.7 notes : 13](#_Toc123069121)

# 0 – topic in this slide :

هنبداء دلوقتى ف شرح الcreation DP ودة ال DP الى بتسعدنا ف حل المشاكل الى هتوجهنا واحنا بنعمل create ل object .

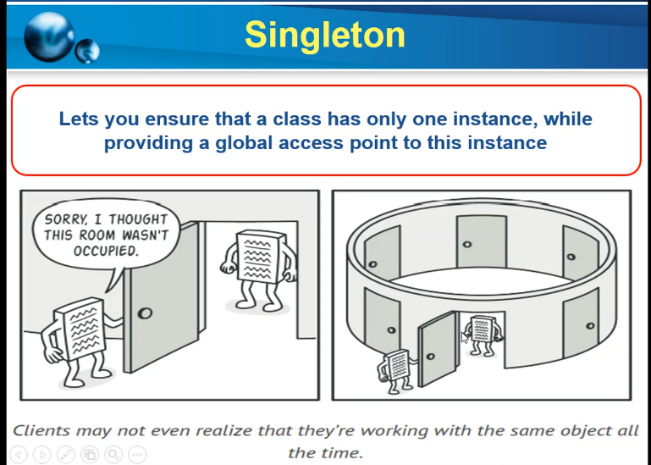


ف كل DP احنا هنوصفه بحمله واحدة ايه المشكله الى هيحلها وممكن يحلها ازاى وبعد كدة نفصل المشكله بالتفصيل وبعد كدة نشوف ال structure بتاع ال DP .

# 1-singleton:

عن طريق ال singleton احنا بنقدر نطمن ان مفيش غير object واحد بس هو الى بيحصله create من class معين , وان ال object دة متاح لاى حد يقدر يستخدامه .

* بعنى ايه الكلام الى فات دة : اوقات بنبقا محتاحين object واحد بس على مستوى ال app من الكلاس معين ولو حد حاول يعمل object جيديد ف انحنا مش بنسمح بدة وبنخليه يتعامل بنفس الobject الى موجود معانا كلنا . وسبب دة ان عوزين يكون ف entry point واحدة للحاجه الى بنشتغل عليها .
* مثلا وانا شغال مع ال database اكون عاوز واحد بس يكون هو الى مسئول عن ال DB يعنى مسئول عن فتح وقفل ال connection وكدة . ف هحط الكلام دة ف كلاس بس عاوز اتاكد ان هيكون ف منه object واحد موجود واى حد هيتعامل مع ال database يتعامل مع ال object االى كلنا بنتعامل بيه .



* ودة مثال تانى لفكرة ليه عاوز يكون ف object واحد : انا بعمل simulation لبرنامج os ف على مستوى ال os مفروض معايه object واحد بس من cpu وال network card ف انا عاوز اى حد يتعامل مع البرنامج بتاعى ميقدرش يعمل create لاكتر من object . يعنى كلنا نتعامل مع object واحد من ال cpu . واى حد تانى يحاول يعمل create لobject جديد نخليه يتعامل مع نفس ال object القديم الى كلنا بنتعامل معاه .

ف ممكن تتخيل ان كل باب من دول هو process بتطلب ال cpu

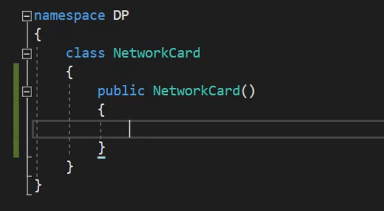
ف كل مرة ال process تطلب ال cpu تلاقى نفس ال object من ال cpu عشان مفيش غير object واحد بس من ال cpu على مستورى

ال app .

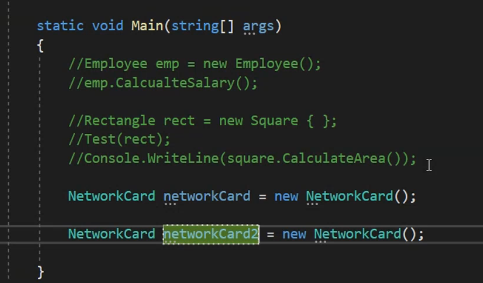
* ممكن واحد يقلى ان مش محتاج اصلا singleton عشان اقدر اضمن ان هعمل object واحد من الكلام ّ!!!!! وان هو هيعمل كل حاجة جوة الكلاس static ف بكدة ابقا حققت الى عاوزة ؟ هقلك صح جدااا ف حاله ان انا الى بعمل ال class و ف احيان كتير الكلاس مش بتكون انت الى عامله انت مرجرد بتستخدامه .

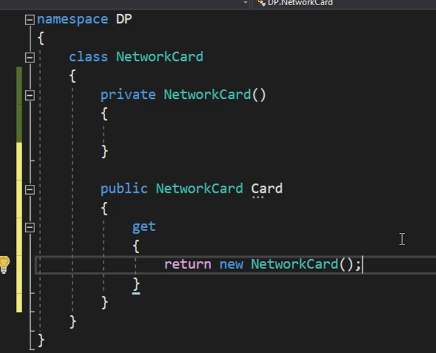
## Implementation for singleton:

المثال دة هيكون على نفس حوار, network card الى شرحته من شويه .

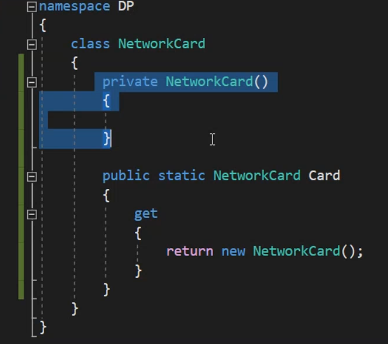
1. هعمل class اسمه network cart بغض النظر هو بيعمل ايه .

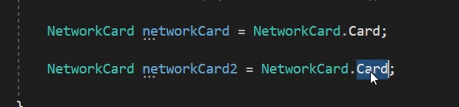
* احنا بقا عوزين نمنع اى حد يعمل create لاكتر من object . يعنى عوزين نمنع المنظر دة .



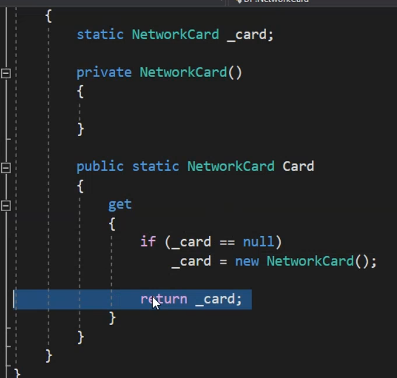
1. ف عوزينن طريقه لما نيجى نحاول نعمل create لاكتر من object من ال network card يرجعلنا نفس الobject القديم الى كنا شغلين بيه .
2. ف هنخلى ال constructor يبقا private وكدة محدض هيقدر يعمل object من برة ال class بس ينفع نعمل object من جوة الكلاس
3. ف تانى حاجة بعد موضع ال private constructor ان اعمل property او function ترجعلى object من الكلاس
4. ف كدة انا نفزت الخطوتين الى فاته . بس ف مشكله ان كدة برضو

محتاح object عشان انادى اداله card .

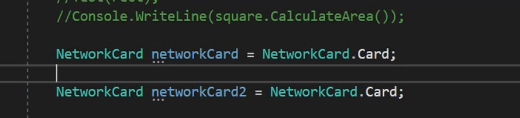
1. ف الحل اخلى داله card تكون static .
2. بعد خطوه رقم اربعه انا كدة اقدر اعمل object من ال network عادى . بس لسه موضوع ان اخلى نفس object يرجع كل ما حد يحاول يعمل create ل new object .

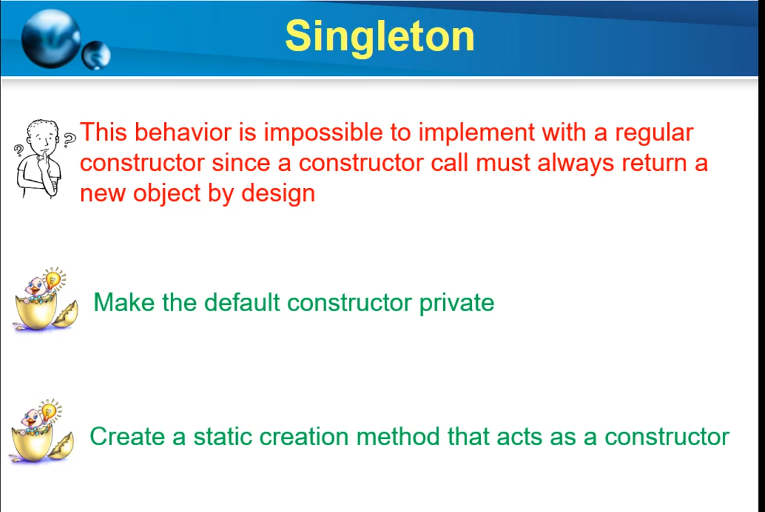


1. ف مفروض ان اعمل reference جوة ال class واول مرة بس حد اعمل new وبعد كدة ارجع الobject الى عملته .

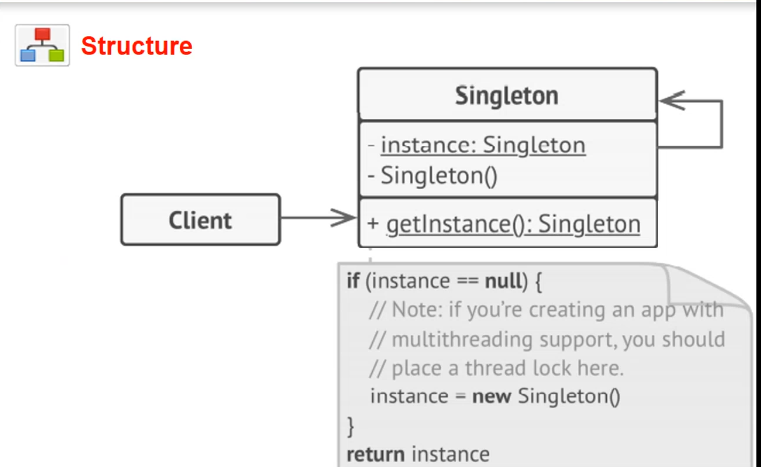


ف كدة انا ضمنت دايما ان object واحد بس هيكون موجود على مستوى ال app .

* ف كدة الاتنين reference دول بيشاورة على نفس ال object .

ودة نفس الكلام الى شرحته .

## 1.2 – singlton strcture :

دة ال strecture الى عملناه .

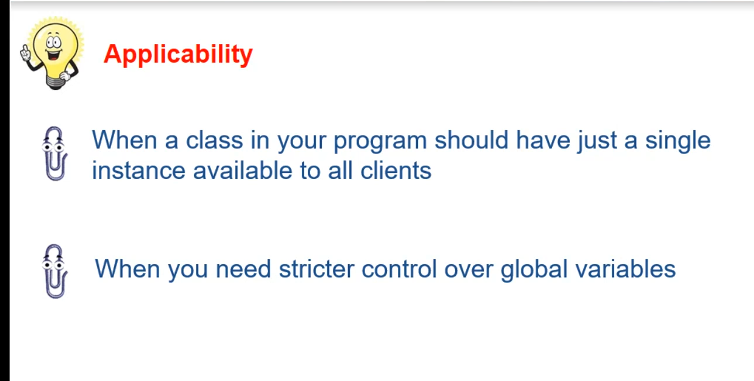
* خيت ال constractor يبقا private
* وعلمت refrance من ال class جواه
* وعملت داله ترجع object من الكلاس والكود بتاع

الداله دى ظاهر ف الاسكرين .

## 1.3-applicability :

امتى استخدام ال singleton DP .

1. لما نكون عوزين نخلى object واحد بس موجود على مستوى ال app او على مستوى الclient .
2. لما تعوز تعمل control على ال global variable . .



## 1.4-problem in singleton.

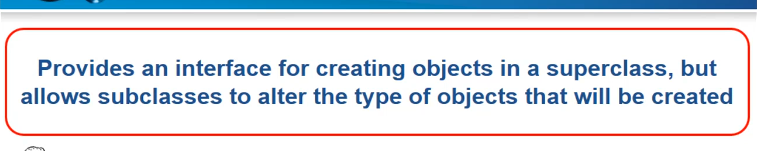
لو انت كنت شغال multithread ف هتكون ف مشكله افرض مثلا ان عندك 2 thread الاوله نفزت ال if وقبل متعمل new طلعت ف التانيه دخلت ونفزت الif ولقت ال card=null ف عملت new وبعد كدة حات ال thread الاول وكملت من بعد ال if ف هتعمل new ف كدة بقى ف 2 object .

. ف لازم تعمل lock للif condition .



# 2-factory method :



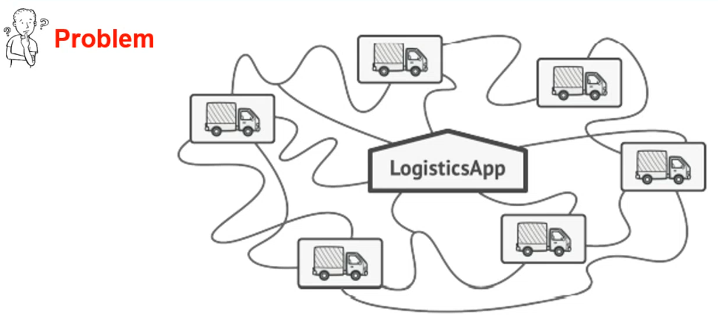


الfactory method بتوفرى طريقه اعمل بيها object جوة parent class وف نفس الوقت بتيح ليا اعمل object منclass تانيه ف ال child class .

يعنى انا عندى class design جواه function بتعمل create for object من كلاس اسمه x ف انا عملت child كلاس بيورث من ال design class ف للاسف كدة انا اجبرت يكون عندى function جوة ال child class تكون بتعمل object من كلاس x وانا مش عاوز كدة انا عاوز object من كلاس y جوة ال child . لو مفهمتش ف تابع معايه . وعلى فكرة احنا عملنا ال factory method من وهقلك انتى بس تابع معايه .

تعاله ناخد مشكله ونشوف ازاى ال factory method هتسعدنا ف حلها :

افرض انك دلوقتى بتعمل تصميم لجزء من system بتاع سوق مثلا ف اول حاجة اطبت منك انك تهندل الموضع بتاع الshaping طريقه نقل البضاعه يعنى .

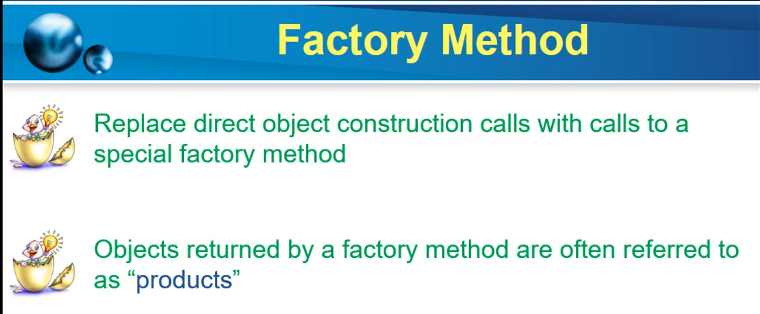
ف انت ف الاول عملت التصميم بتاعك معتمد على عربيات النقل بس trucks .ف انت كدة ف الاول التصميم بتاعك هوtautly couple مع عربيات النقل .

طيب بعد شويه الشركه فتحت فروع تانى وطرق ال shipment اتغيرت ممكن يبقا فيه طرق نقل باطيارة او السفينه . ف كدة هحتاج اعيد تصميم ال app .

ف الfactory method هتسعدنا من الاول وانا بعمل التصميم بتاعى يكون extensible . يعنى اقدر ازود عليه من غير ماثر على التصميم الى موجود . ف تعاله نشوف خطوات ال factory method .

## 2.1 factory method step:

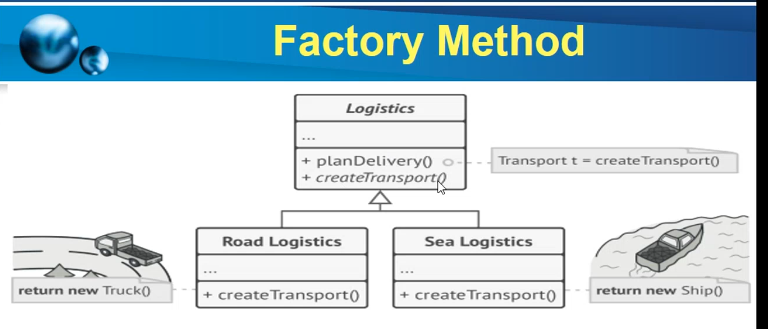
1. بدال متعمل object من class معين جوة الكلاس اعتمد على method اعمل جوها ال create لل object .الmethod الى هتعمل جوها ال object دى اسمها factory method .
2. الobject الى راجعه من ال factory method اسمها products .
3. ومعلومه عموما product هو المنتج النهائى الى هيرجعلك من الfunction سواء كانت factory method او لا

دى اسكرينا بنفس الخطوات الى فات خطوات ال factory method .

## 2.2 example on factory method:

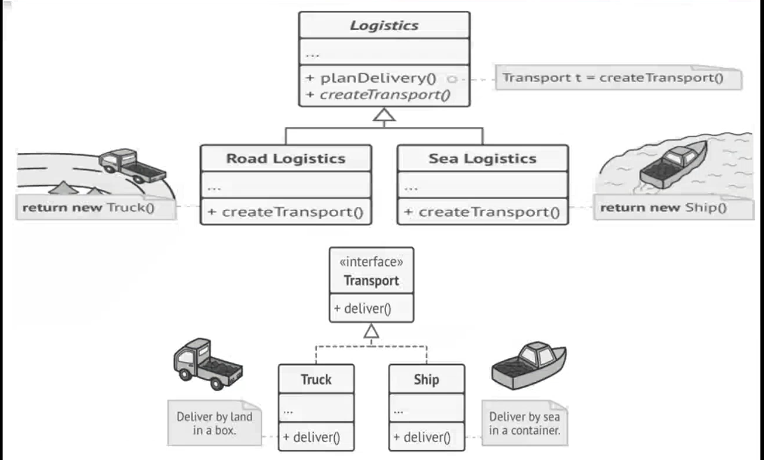
هنا هحل المشكله الى شفتها من شويه بتاعه عربيات الشحن truck .

زمان ف الاسكرين بتاعه ال problem كان كلاس logistic بيعمل create ل truck object وقلنا ان دة اسمه tautly couple . وقلنا factory method هتحل المشكله تعاله ناخد خطوات الحل يبو الصحاب.



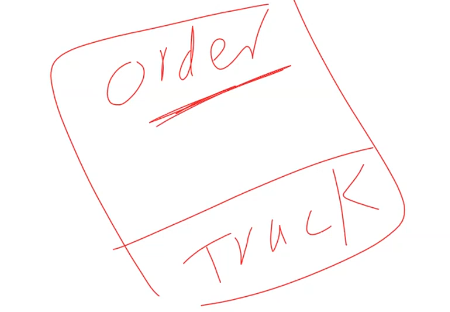
1. اول حاجة بدال مكان بيعمل object form truck عملت function اسمها create transport والداله دى هى abstract function جوة abstract class ف حالتنا هنا وممكن تكون عادى جواها implementation ف حلات تانى او ممكن تكون جوة interface مش abstract class .
2. Transport هنا دة interface او abstract class ولسه هظهرهولك ف الاسكرين الى جايه. ف كدة ال logistic بيقول اى حد عاوز يشتغل معايه يعمل implement للinterface الى اسمه transport .
3. كدة بقا انا اقدر اضيف اى logic للنقل يعنى لو عاوز يكون عندى طريقه للنقل البرى هعمل class اسمه road واخليه يورث من ال logistic ويعمل override للداله الى اسمها create transport وجوها يرجع object من truck بس كدة لازم ال truck يكون وارث من ال transport interface عشان الفنكشن الى اسمهاcreate transport بترح حاجة من نوع transport .

* و لو عاوز اضيف logic للنقل البحرى هعمل class اسمه sea واخليه يورث من الlogistic class ويعمل override للداله الى اسمها createtransport ويخيها ترجع object من ship .

1. وتانى خلى بالك ship ,road لازم يبقا وارث من transport class وعشان كدة قلتلك ان اى حد حابب يتعامل مع كلاس logic لازم يورث من transport class .
2. دة شكل التصميم النهائى .
3. ف كدة ال function الى اسمها create transport تبقا factory method .
4. Factory method : يعنى function تكون مسئوله عن creation for objects .
5. كدة بستخدام ال factory method قدرت جوة كلاس logistic اقول ان انا بشتغل مع اى حاجة من نوع transport . وكدة كل حد هيورث منى يعمل override للفنكشن الى اسمها createtransport ويخلها ترج object زى مهو عايز مرة يخليها ترج object من truck ومره يخليها ترجع object من ship بس لازم ال ship او ال truck تكون وارث من transport .
6. انا كدة كصاحب كلاس logistic هعمل الclass مرة واحدة وبعد كدة تقدر الناس تشتغل عليه يعنى اى حد يقدر يضيف اى طريقه هو عوزها زى مثلا النقل الجوى كل الى هعمله ان اعمل class اسمه area واخليه يورث من logistic واعمل كلاس اسمه plan واخليه يورث من transport وجوة ال area class اعمل فنكشن createtransport الى بترجع object من كلاس plan بس على شكلreference من transport .
7. **كل الرغى دة احنا شرحنا قبل كدة على فكرة ف مثلا كنت بعمل فيه تصميم لشكره وكان عندى انواع من الشركات وكمان انواع من الprogrammer المثال دة ف part .**

**لو انت كل دة مفهمتش تعاله اديك مثال مختصر شويه**

## 2.3 –example2 on factory method :

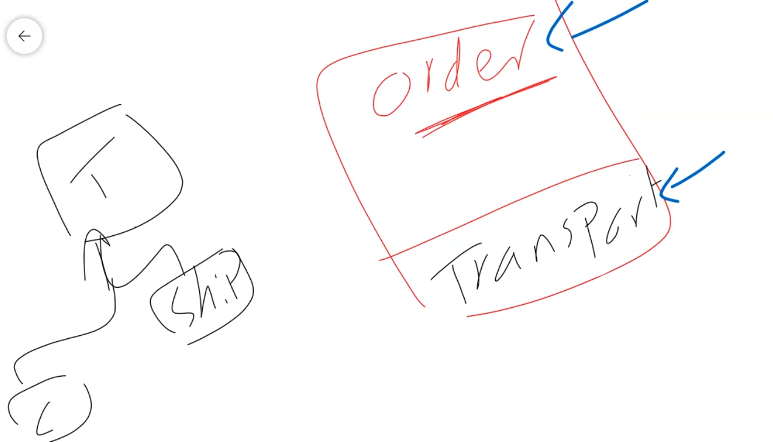
انا عندى كلاس امسه order ف عاوز احدد طريقه نقل ال order . ف عندى طريقه نقل عن طريقه البر فقط ف هحط object form truck جوة كلاس ال order .

بعد شويه بقينا ننقل الorder عن طريق البحر والجو ف كدة التصميم بتاع order بقا

مش صح عشان هو رابط دايما توصيل ال order ب ال truck . ف عشان نحل المشكله

هنخلى ال order مرتبط ب transport وسيله نقل وال transport دة اما abstract class

او interface .

ف دة شكل التصميم ناو .

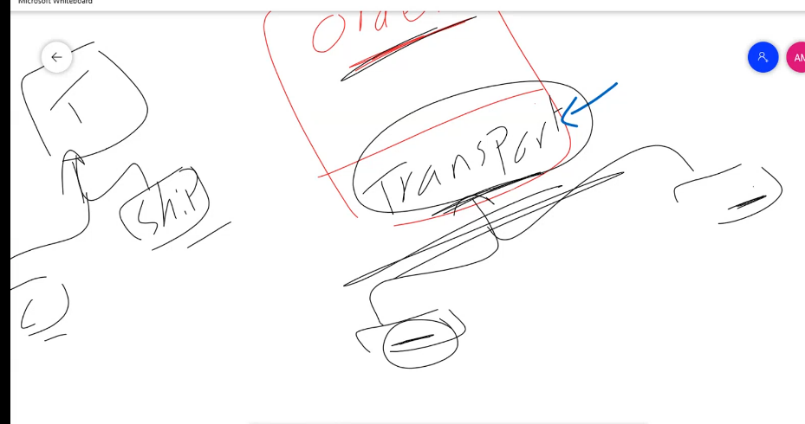
T for transport

C for car

طيب تعاله نزود ف التصميم تانى . هنقول ان ال order الى هتنتقل عن طريق البر هتتعامل بطريقه مختلفه عن الorder الى هتنتقل عن طريق البحر . ف معنى كدة ان عندى انواع من ال order يعنى هعمل class تورث من ال order وبالتالى مينفعش يكون ال creation ل class transport جوه ال order عشان class order ميعرفش انهى نوع ال transportation الى هيركبه .

ف ال creation بتاع ال transport هيكون جوة ال child class الى ورث من ال order وهيكون جوة function .

وعشان اوحد طريقه ال creation الfunction دى هتكون abstract fuction جوة كلاس order . والchild يعملها override .

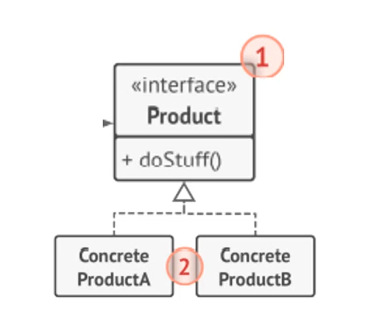


## 2.4-structure of factory method:

1. الproduct دة زى ال transport الى شرحناه . ودة interface وال creation بتاعه هيختلف من كلاس للتانى.

لو انا مثلا موجود جوة road logistic يبقا عمل create لtruck

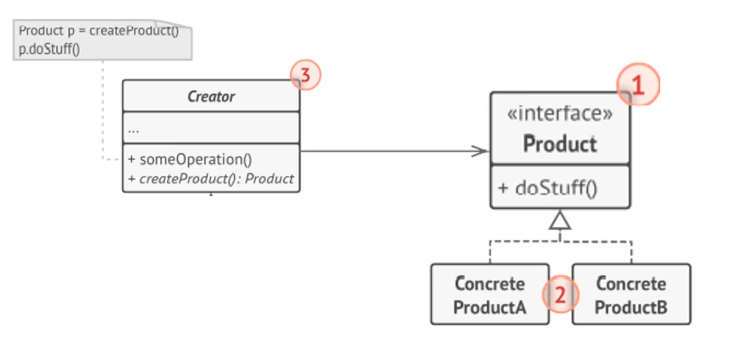
.

1. فلازم الاول يكون عندى concrete class بتورث من product . ف هيكون عندى

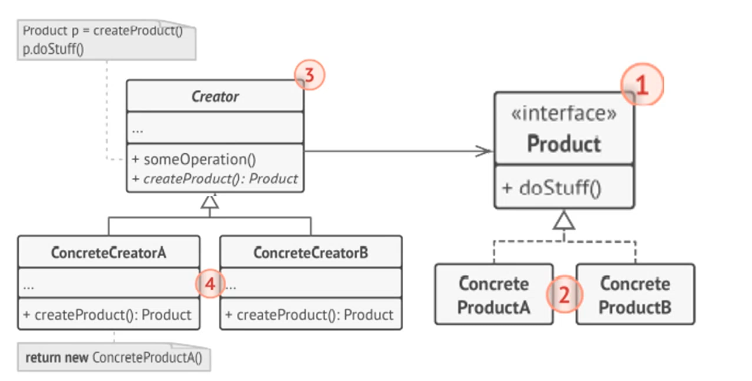
ProductA ,productB

1. الكلاس الاساسى بتاعى الى مسئول عن ال creation . وهو abstraction بروض وهيكون جواه function اسمها createpoduct

بس هو مش هيكون عارف هعمل create لانهى نوع من ال product . ومتنساش بقا ان انا بستخدام ال abstraction عرفت ال what فاضل ال how الى بنعرفها ب ال concrete .



1. ف عشان كل الكلام الى قلته ف رقم 3 كدة انا هعمل inherit من creator . عشان زى ما انا معايه abstraction من creator يكون معايه concrete from creator . ف ال concreateCreatorA جواه method بترع object form poductA وهكذا.



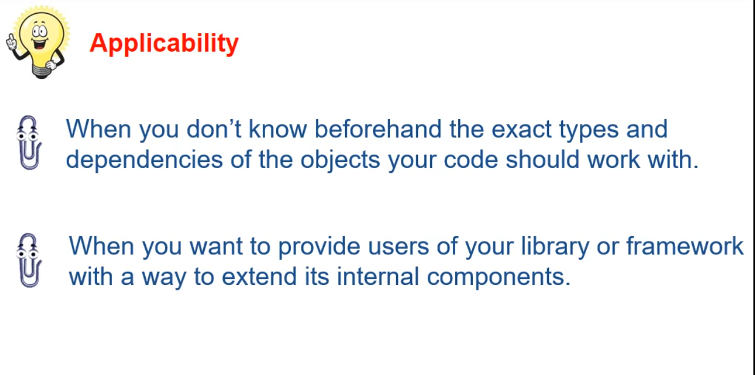
## 2.5-applicability :

امتى استخدام ال factory method :

1. لما اكون مش عارف ايه انواع ال object الى هتعامل معاه مسبقنا . لو فاكر التصميم الى عملته للشركه والمبرمجين . وقلنا ان لازم نعتمد على abstraction مش concrete .

* ف ال product as abstraction هيكون هو الemployee وال concrete class هيكون هو ال designer ,tester الى بيورث من employee .
* وعلى مستوى ال company as abstract كدة هيكون جواه function call getEmployee وهنورث من company ونعمل override للfunction getEmployee ونعمل جوها ال creation بتاع ال object الى عوزها .

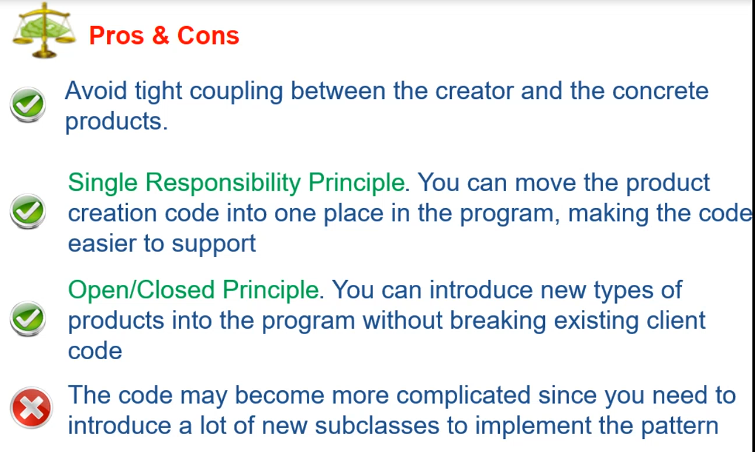
1. انك تكون عاوز تعمل extend للfunction من غير متفتح الكود وتعدل فيه .



## 2.6- pros and cons :

مميزات وعيوب ال factory method .

1. بتشيل ال tight couple وبتخلينا شغال loosely couple . كدة ال creator مش معتمد على product ف كدة لو اتغيرت ال product ال creator مش هيتاثر .
2. Single responsibility principle مطبق بشكل كبير . عشان كدة كل كلاس مسئول عن حاجته ف ال creator مي بيعمل create لل product انا بخلى ال child class هى الى تعمل الشغل بتاعها .
3. Open –close principle : انا بقدر اعمل extend لل function من غير ملعب ف الكود.
4. العيب الاساسى الى هيكون ف كل ال pattern ان الكود هيكون complicate حبيتين .

دة نفس الكلام الى فوق منه بس بشكل مرتب.

## 2.7 notes :

مش شرط ال pattern يكون مطبق كل مبادئ ال solid principle وكمان ممكن زى مهنشوف كمان شويه يكون ف pattern بيتعارض مع مبداء من بتوع ال principle solid .