آزمایش هشتم

آزمایشگاه مهندسی نرم افزار

نام اعضا و شماره دانشجویی:
محمدصالح سعیدی ۹۶۱۰۵۸۴۲
امیرحسین محسن‌نژاد ۹۶۱۰۵۳۹۴

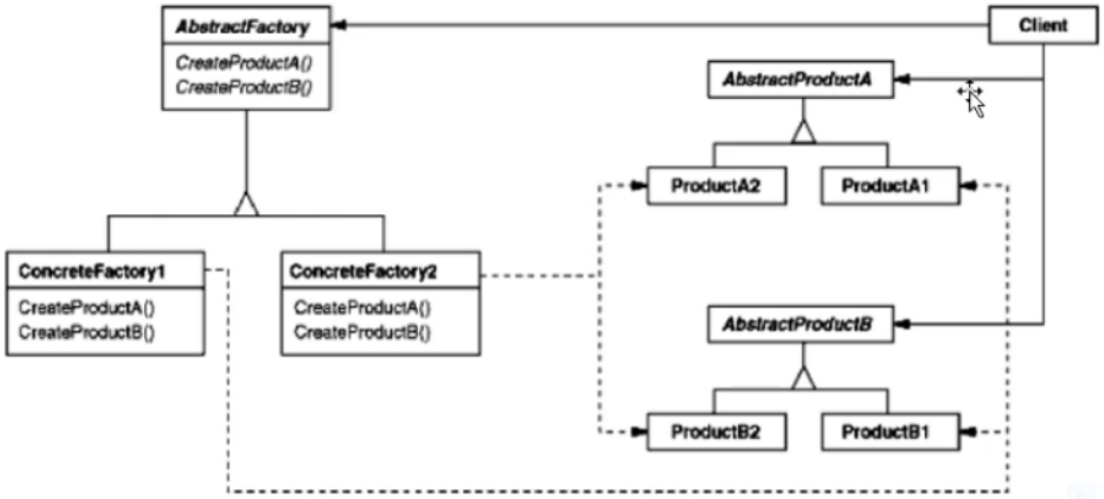
ترم بهار ۱۴۰۱

مقدمه

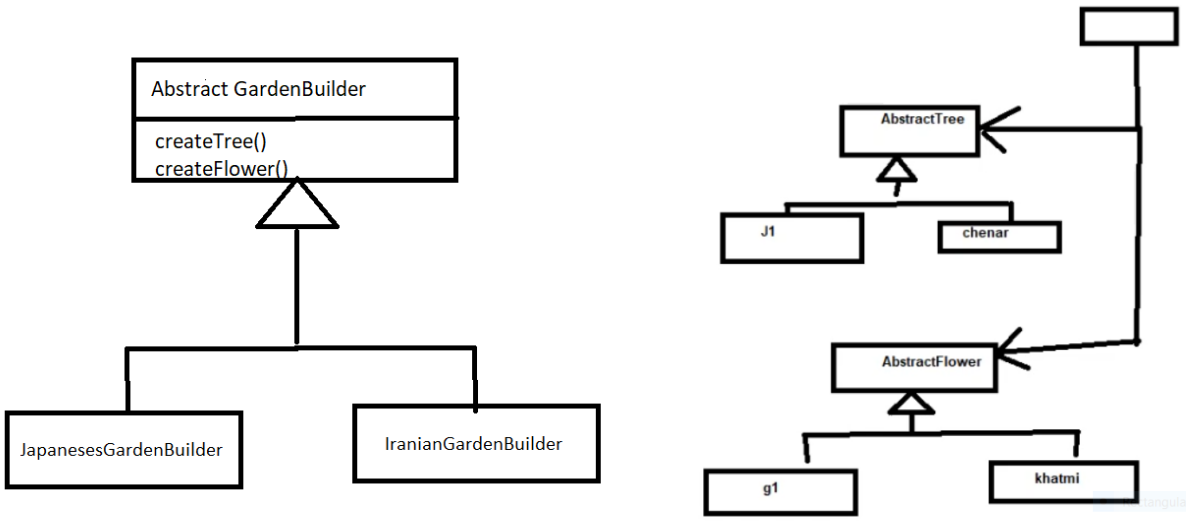
در این آزمایش سعی کردیم شبیه سازی از باغی را با استفاده از دو الگوی طراحی Abstract factory و prototype انجام بدیم. مانند یکی دیگر از آزمایش‌های گذشته فرایند توسعه TDD را پیش بردیم به این صورت که ابتدا تست ها و سپس پیاده سازی اصلی و نیازمندی‌ها را انجام دادیم. برای مشاهده جزییات بیشتر، می توانید به اینجا مراجعه کنید.

پیاده سازی Abstract factory

با توجه به توضیحات داده شده در فیلم، قرار داده شده در cw فلو این الگوی طراحی به این صورت است.



در ادامه معرفی این الگوی طراحی مثال باغ مطرح شده و نمودار پیاده سازی آن با کمی تغییر به صورت زیر معرفی شد:

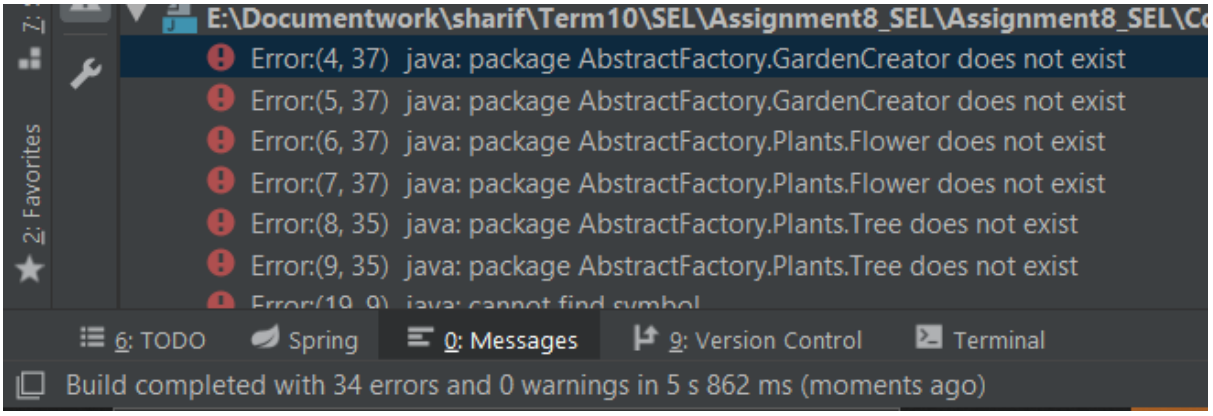


که این نمودار به این موضوع اشاره دارد که ما به دو نوع باغ نیاز داریم، باغ ایرانی و ژاپنی که هرکدام از آن ها دارای یک درخت و یک گل است. ولی نوع این دو فرق دارد و بطور مثال باغ ایرانی باید دارای درخت چنار و گل ختمی باشد.

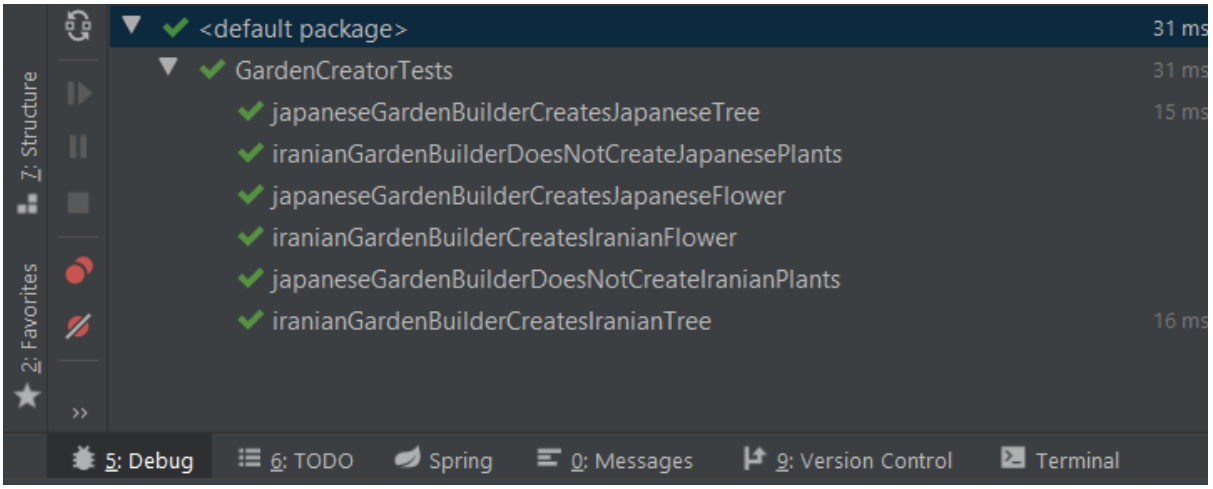
پیاده سازی انجام شده نهایی نیز بر همین اساس جلو رفت و تمام موجودیت های پیاده سازی شده، مطابق با نمودار بالا پیش رفتند.

مراحل انجام گرفتن کار نیز بر اساس TDD بود. به این صورت که در ابتدا تست های مربوط به این بخش

نوشته شد و کامپایل شد.

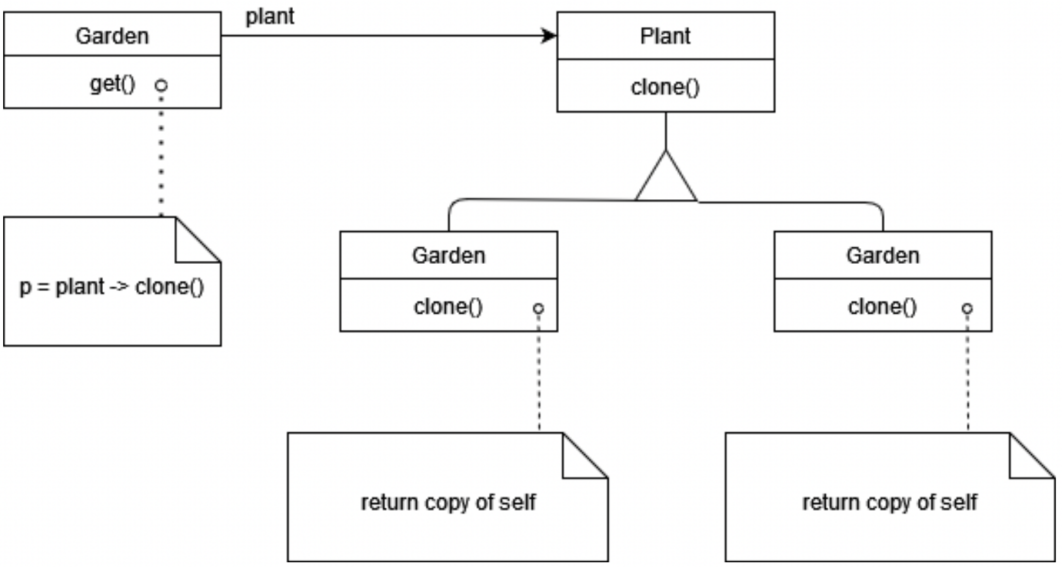


که همانطور که دیده می شود، نیاز است که کلاس های اصلی پیاده شوند تا تست ها پاس شوند.
بعد از پیاده سازی کلاس های اصلی، در تصویر زیر می توان مشاهده کرد که تست ها به درستی پاس شدند.



پیاده‌سازی با prototype:

مسئله باغ و گل و درخت را به اینگونه مدل کردیم که می‌خواهیم از روی یک درخت نمونه که همان prototype است، درخت‌های دیگری را clone کنیم و همین کار را برای گل نیز انجام دهیم.



اگر به روش TDD، در قدم اول تستها را می‌نویسیم و به خطای زمان کامپایل بر می‌خوریم.

پس کلاسهای زیر را پیاده‌سازی می‌کنیم تا خطاهای کامپایل برطرف شود.

موجودیت Garden:

این موجودیت یک موجودیت بالاتر است که ابتدا یک گل و درخت اولیه را می‌سازد و سپس دارای یک تابع get است که بسته به نوع ورودی، یک شیء بسته به چیزی که ورودی داده‌ایم clone می‌کند.

موجودیت Plant:

یک abstract class به نام Plant ساخته شد و کلاسهای Flower و Tree از آن ارث میبرند. در این کلاس علاوه بر fieldهای کلی مانند color وisLiveInApartment، یک abstract method به نام clone وجود دارد که توسط بچه‌های این کلاس override میشود. کاری که این تابع قرار است انجام دهد این است که یک object را کپی کرده و برگرداند. همچنین تابع equal پیاده سازی شد تا تست کنیم که آیا واقعا fieldهای شیء برگردانده شده و شیء اصلی یکی هستند یا خیر. حال به سراغ یکی از بچه‌های Plant میرویم.

موجودیت Flower:

تنها ویژگی خصوصی این کلاس، فیلد lifeTime است. هنگامی که یک شیء از این کلاس از روی شیء از پیش موجودی ساخته شود، عملکرد آن به اینگونه است که ابتدا super را صدا می کند و سپس این فیلد را از روی آن شیء مقدار می‌دهد.

```
public Flower(Flower target) {
    super(target);
    this.lifeTime = target.lifeTime;
}
```

و تابع Oclone نیز به صورت زیر پیاده‌سازی شده است:

```
@Override
public Plant clone() {
    return new Flower(this);
}
```

حال پس از انجام تمامی این مراحل به تست برنامه می‌پردازیم و همانطور که مشاهده می‌کنید تست‌ها با موفقیت انجام شدند.