

به نام خدا

آزمایش چهارم آزمایشگاه مهندسی نرم افزار

کاربرد عملی الگوهای طراحی شی گرای Abstract Factory و Prototype با استفاده از روش  
TDD

حمیدرضا هدایتی ۹۶۱۰۹۹۳۹

علیرضا دیزجی ۹۶۱۰۵۷۴۵

دانشگاه صنعتی شریف

پاییز ۱۴۰۰

## Abstract Factory

برای پیاده‌سازی این الگو از مسئله "ساخت باغ های ایرانی و ژاپنی" که در کلاس مطرح شد استفاده کردیم. این بخش شامل ۴ قسمت است: Creator, Flower, Tree, Provider

### :Flower

شامل کلاس های مربوط به گل می باشد. فایل Flower.java شامل یک کلاس abstract می باشد که "نحوه رشد گلبرگ گل" علت اصلی ابسترتک بودن آن می باشد چرا که فرض شده است نحوه رشد گلبرگ یک گل ایرانی (رز) با یک گل ژاپنی (ساکورا) می تواند از لحاظ تعداد و رنگ آن متفاوت باشد.

### :Tree

شامل کلاس های مربوط به درخت می باشد. فایل Tree.java شامل یک کلاس abstract می باشد که "نحوه رشد درخت" علت اصلی ابسترتک بودن آن می باشد چرا که فرض شده است نحوه رشد یک درخت ایرانی (چنار) با یک درخت ژاپنی (ماپل ژاپنی) می تواند از لحاظ ارتفاع، تعداد برگ و طول متوسط آنها متفاوت باشد.

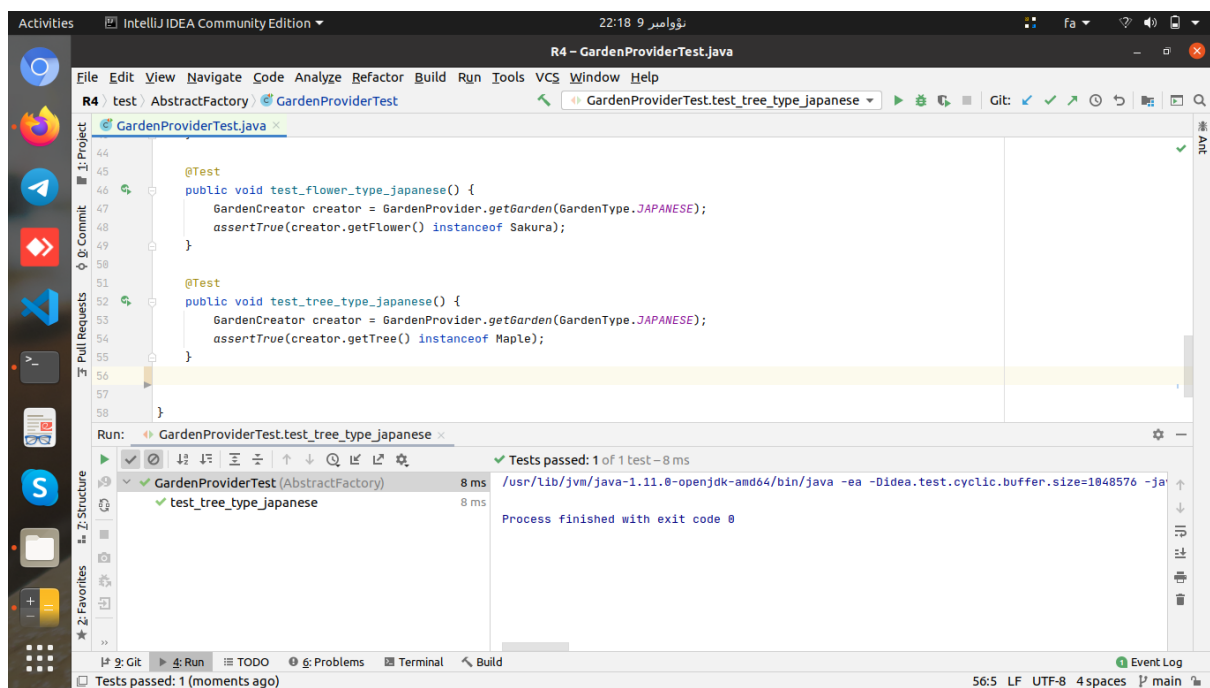
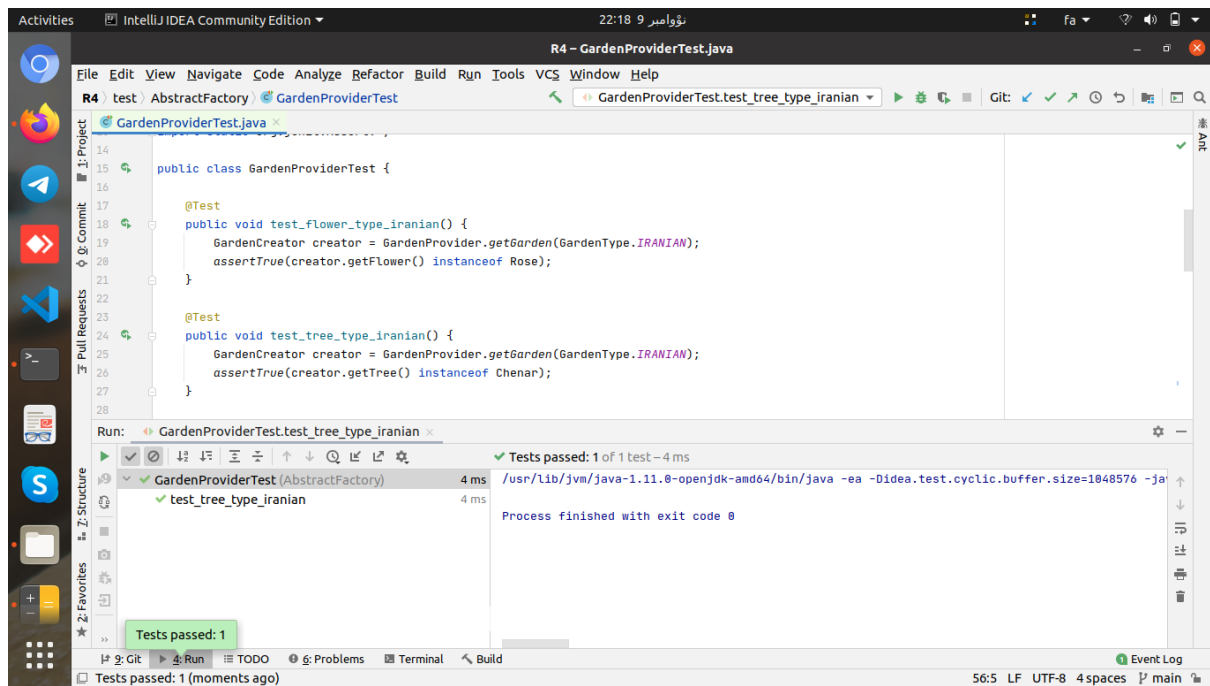
### :Creator

شامل کلاس های مربوط به ساخت باغ می باشد. فایل GardenCreator.java شامل یک کلاس abstract می باشد که "ساخت گل" و "ساخت درخت" علت اصلی ابسترتک بودن آن می باشد چرا که طبیعتا باغ ایرانی از گل و درخت ایرانی (و همینطور برای باغ ژاپنی) تشکیل شده است (در اینجا فرض شده است هر باغ تنها از یک درخت و گل تشکیل شده است).

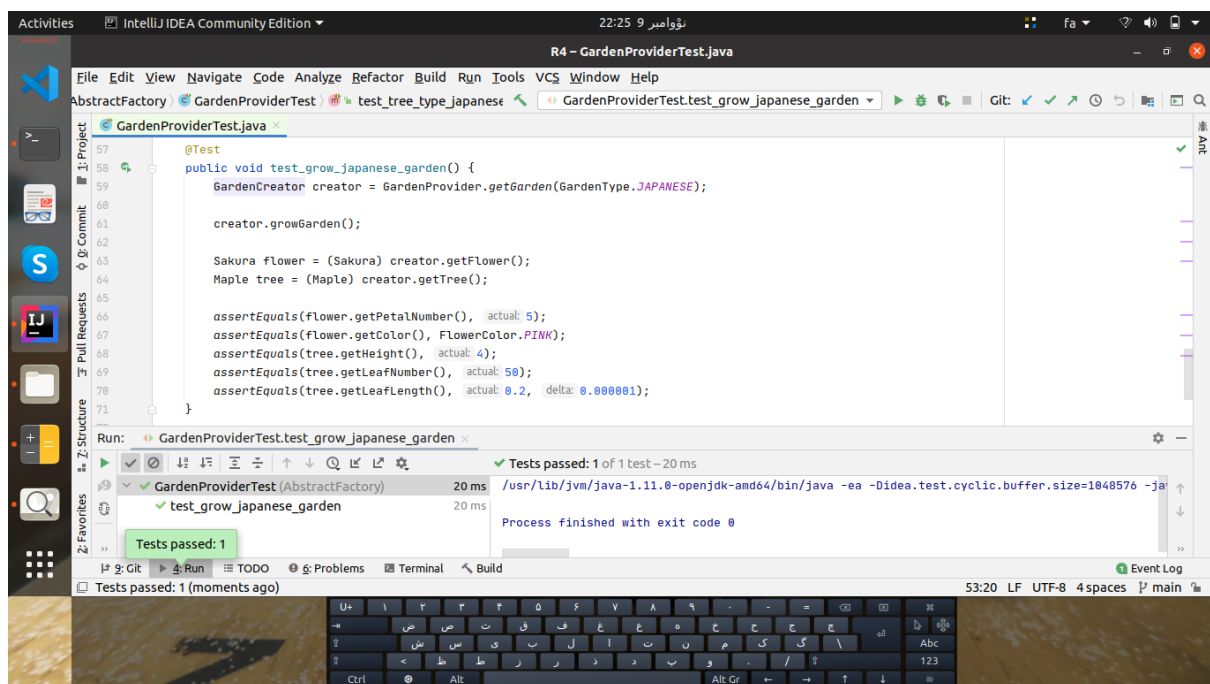
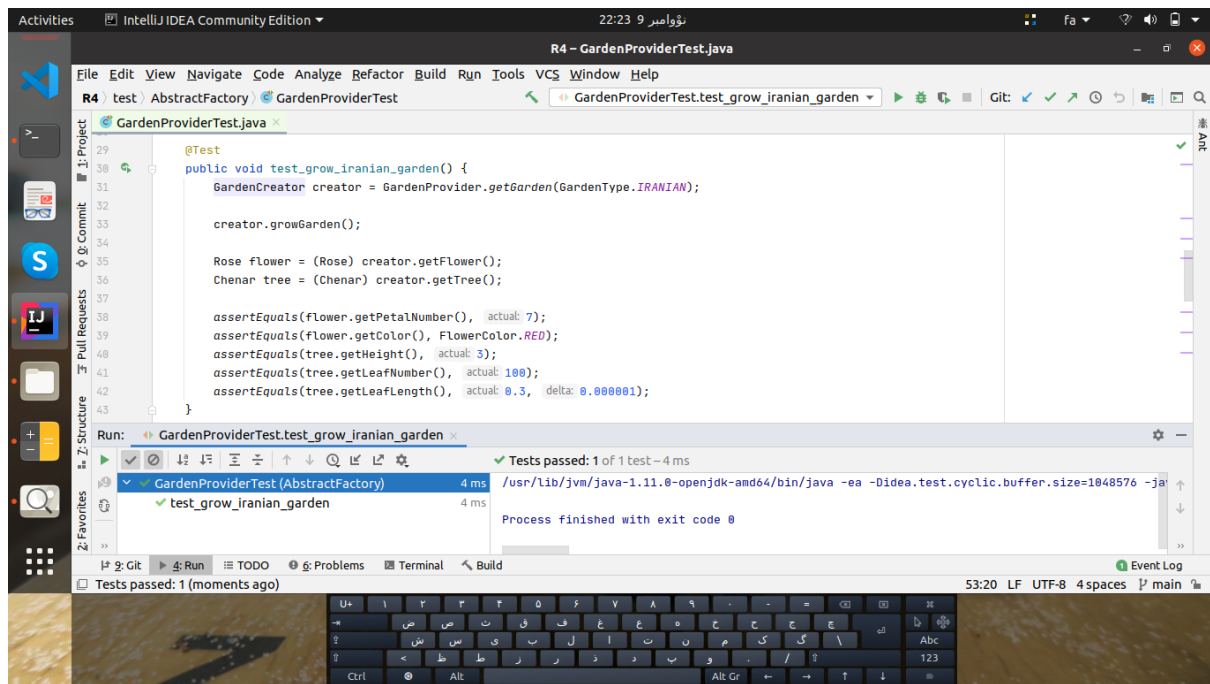
### :Provider

شامل یک کلاس است که درخواست ها را براساس خواسته مشتری ثبت می کند. خواسته مشتری تنها مشخص می کند که چه نوع باغی را دوست دارد.

حال از آنجایی که باید به صورت TDD دولوپ می کردیم، ابتدا تست های مربوط به ساخت باغ ها را ایجاد کردیم. یعنی اگر فرضا باغ ایرانی داریم آیا گل مان از نوع رز است یا خیر؛ یا درخت مان از نوع چنار است یا خیر. به همین ترتیب برای باغ ژاپنی.



پس از آن به دنبال تست و توسعه "رشد باغ ها" بودیم که پاس شدن تست های آن را می توانید در ذیل مشاهده کنید.

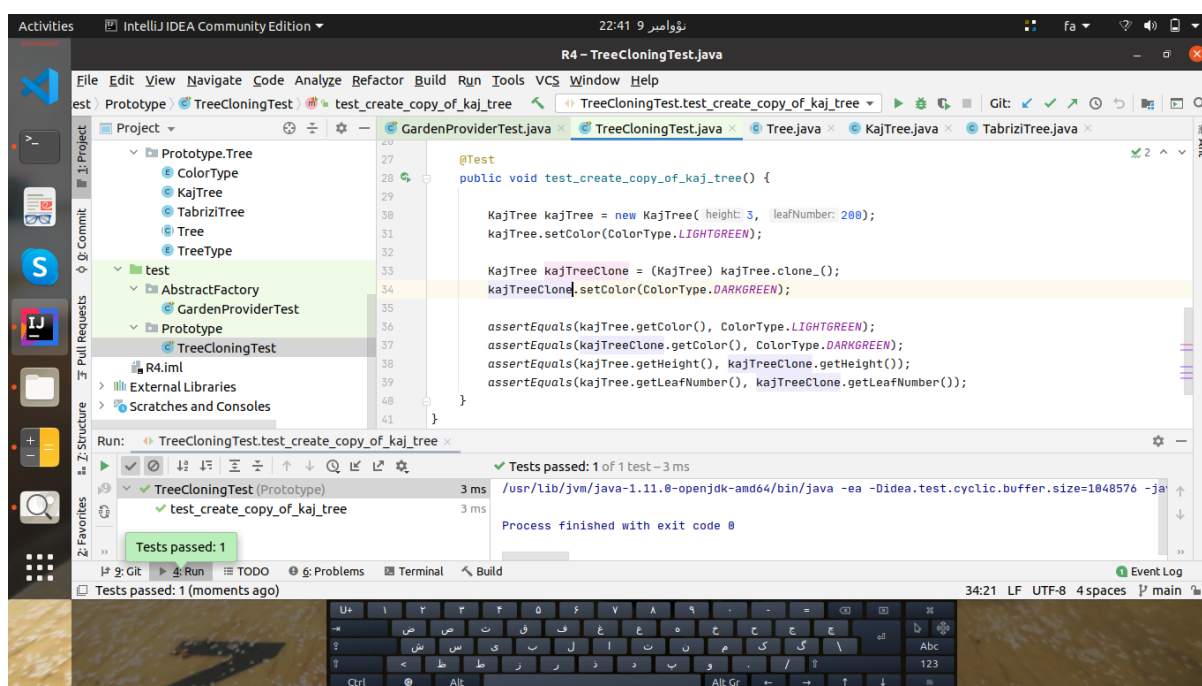


## Prototype

برای پیاده‌سازی این الگو از مسئله کپی سازی درخت های کاج و تبریزی استفاده شده است. فرض را بر این اساس گرفتیم که هر درخت به صورت کلی دارای ۳ ویژگی ارتفاع، تعداد برگ و رنگ می باشد منتها درخت های مختلف کاج از لحاظ رنگ و درخت های مختلف تبریزی از لحاظ ارتفاع با هم تفاوت دارند. حال کلاس Tree در فایل Tree.java از نوع انتزاعی است که نحوه پیاده‌سازی کپی برداری را به عهده کلاس های ارث بر گذاشته است. برای ساخت یک درخت تبریزی

جدید بایستی است ویژگی های رنگ و تعداد برگ ثابت بمانند و برای ساخت یک درخت جدید کاج بایستی ویژگی های تعداد برگ و ارتفاع ثابت بمانند.

در نهایت از آنجایی که می بایست به صورت TDD دولوپ می کردیم، در حین توسعه نحوه عملکرد کپی برداری این دو نوع درخت را بررسی می کردیم که پاس شدن تست های آنها را می توانید در ذیل مشاهده کنید.

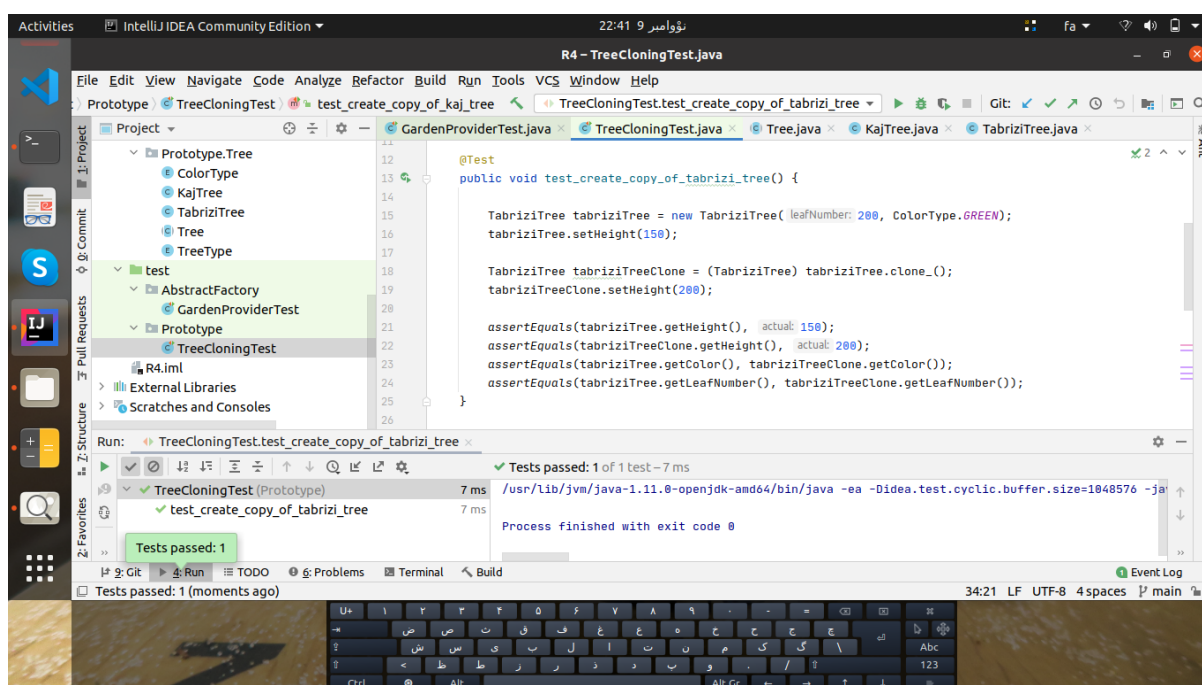


The screenshot shows the IntelliJ IDEA interface with the 'TreeCloningTest.java' file open. The test method 'test\_create\_copy\_of\_kaj\_tree' is highlighted, showing the creation of a 'KajTree' object with height 3 and leaf number 200, setting its color to 'LIGHTGREEN', cloning it, and setting the clone's color to 'DARKGREEN'. The test passes, as indicated by the 'Tests passed: 1 of 1 test - 3 ms' message in the Run window.

```
@Test
public void test_create_copy_of_kaj_tree() {
    KajTree kajTree = new KajTree( height: 3, leafNumber: 200);
    kajTree.setColor(ColorType.LIGHTGREEN);

    KajTree kajTreeClone = (KajTree) kajTree.clone();
    kajTreeClone.setColor(ColorType.DARKGREEN);

    assertEquals(kajTree.getColor(), ColorType.LIGHTGREEN);
    assertEquals(kajTreeClone.getColor(), ColorType.DARKGREEN);
    assertEquals(kajTree.getHeight(), kajTreeClone.getHeight());
    assertEquals(kajTree.getLeafNumber(), kajTreeClone.getLeafNumber());
}
```



The screenshot shows the IntelliJ IDEA interface with the 'TreeCloningTest.java' file open. The test method 'test\_create\_copy\_of\_tabrizi\_tree' is highlighted, showing the creation of a 'TabriziTree' object with leaf number 200 and color 'GREEN', setting its height to 150, cloning it, and setting the clone's height to 200. The test passes, as indicated by the 'Tests passed: 1 of 1 test - 7 ms' message in the Run window.

```
@Test
public void test_create_copy_of_tabrizi_tree() {
    TabriziTree tabriziTree = new TabriziTree( leafNumber: 200, ColorType.GREEN);
    tabriziTree.setHeight(150);

    TabriziTree tabriziTreeClone = (TabriziTree) tabriziTree.clone();
    tabriziTreeClone.setHeight(200);

    assertEquals(tabriziTree.getHeight(), actual: 150);
    assertEquals(tabriziTreeClone.getHeight(), actual: 200);
    assertEquals(tabriziTree.getColor(), tabriziTreeClone.getColor());
    assertEquals(tabriziTree.getLeafNumber(), tabriziTreeClone.getLeafNumber());
}
```

ریازیتوری این پروژه را می توانید از [این لینک](#)، بخش R4، مشاهده کنید.