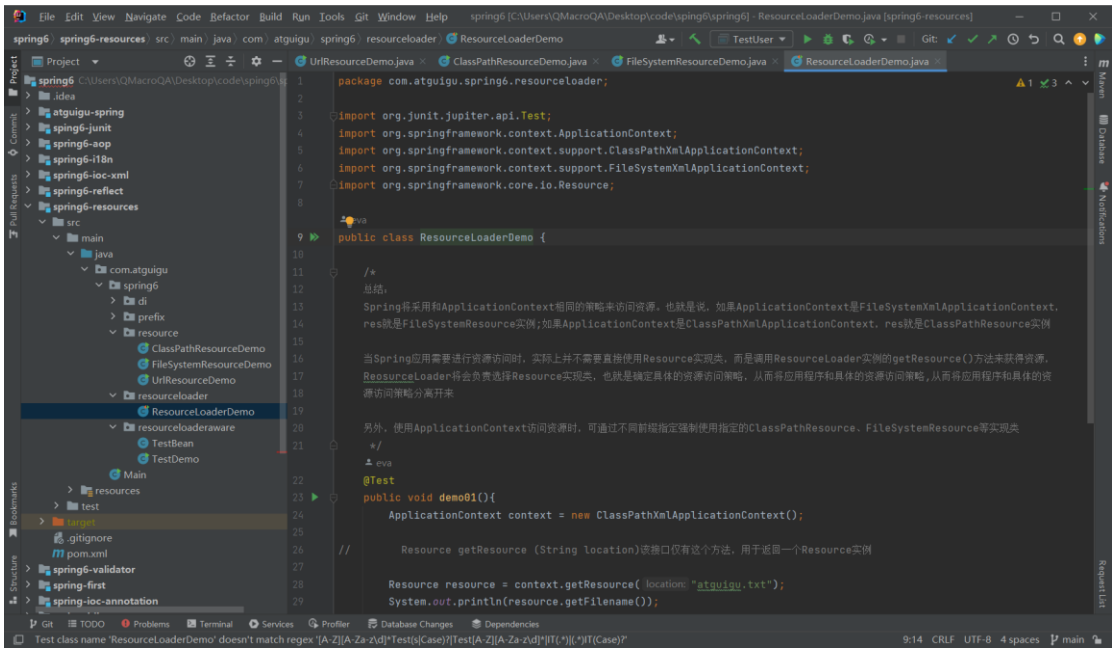
8.5.3、ResourceLoader 总结

Spring将采用和ApplicationContext相同的策略来访问资源。也就是说，如果ApplicationContext是FileSystemXmlApplicationContext，res就是FileSystemResource实例；如果ApplicationContext是ClassPathXmlApplicationContext，res就是ClassPathResource实例

当Spring应用需要进行资源访问时，实际上并不需要直接使用Resource实现类，而是调用ResourceLoader实例的getResource()方法来获得资源，ReosurceLoader将会负责选择Reosurce实现类，也就是确定具体的资源访问策略，从而将应用程序和具体的资源访问策略分离开来

另外，使用ApplicationContext访问资源时，可通过不同前缀指定强制使用指定的ClassPathResource、FileSystemResource等实现类

```
Resource res = ctx.getResource("calsspath:bean.xml");
Resrouce res = ctx.getResource("file:bean.xml");
Resource res = ctx.getResource("http://localhost:8080/beans.xml");
```



```java
    @Test
    public void demo01(){
        ApplicationContext context = new ClassPathXmlApplicationContext();

//        Resource getResource (String location)该接口仅有这个方法，用于返回一个Resource实例

        Resource resource = context.getResource( location: "atguigu.txt");
        System.out.println(resource.getFilename());

    }

    eva
    @Test
    public void demo02(){
        ApplicationContext context = new FileSystemXmlApplicationContext();
        Resource resource = context.getResource( location: "atguigu.txt");
        System.out.println(resource.getFilename());

    }
}
```