Requierd part:P27-P48

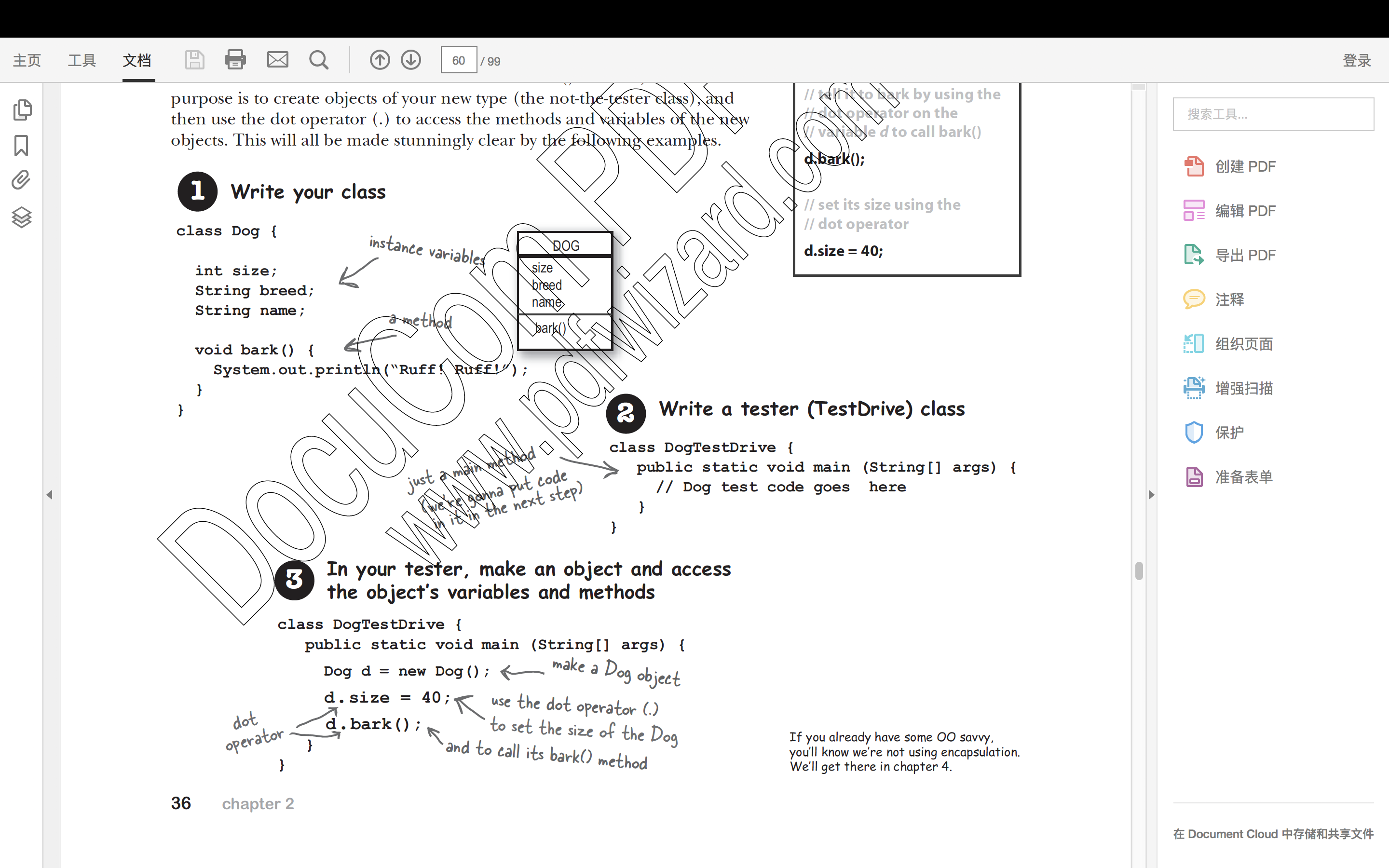
1. use main(): not object-oriented
2. OO: flexibility, extensibility, do not have to touch the code that has been touched and delivered
3. Just change the method in one class, do not have to touch other parts
4. Class includes method, method
5. Classes have in common –abstract out—inheritance

Superclass &subclass

1. Override: redefines inherited methods when want to change/extend
2. Just invoke, not to care how,
3. New object: own behavior
4. Data method in one class,
5. Design an object:

Things the object knows：instance variables: use when invoke the object

Things the object does：methods

1. Instance == object
2. Class != object, class construct object
3. Object has own instance variables
4. Make the object: 2 class : use & test
5. Put **main method** in test class
6. Main method: create/get objects of use class. Get objects --.
7. Job of test class: try out v & met..
8. Real : its objects we want to use
9. Format 
10. In main() not in Objectville , should be object talk to each other
11. Talking: objects calling methods on one another
12. main() :call methods, start app
13. The Heap: object-allocate memory space on the heap 🡪 object never be used: eligible for garbage collection-throw out unreachable objects –space can be reused
14. No concept of ‘global ’ variables and methods in Java OO program
15. public and static 🡪 method behave like a global
16. public, static,final…………..constant…..
17. everything goes in a class , global-like things are exceptions but not rules
18. a java program: pile of classes >= 1 ; main method:start;
19. **question: JVM CD-ROM**
20. files-a.jar file,(manifest indicates main())
21. run time :objects talking to each other
22. p42 A：can not compile reason: not new an object
23. p45 my answer:object,class,method,class&object,?variable instances,object&class,M&I,I,class,?,?,I