Because there are three public classes, so separate them in three files

CarDoor.java////

```
/**
 * CarDoor is a class extends from carPart class and implements LockableDoor and
 * Alarm interfaces
 *
 * @author Peng Gao (pgaooscar@gmail.com)
 * @version 0.1 Novenmber 14 2020
 */
public class CarDoor extends CarPart implements LockableDoor, Alarm {

	// Car door has a lock
	/**
	 * lockFlag is a boolean, check is car door lock or not, initialized false
	 */
	private boolean lockFlag = false;

	// Car door has an alarm switch for you to turn on/off alarm
	/**
	 * alarmFlag is a boolean, check is the car door alarm on or not, initialized
	 * false
	 */
	private boolean alarmFlag = false;

	// Implement this method
	/**
	 * check is the car door alarm on or not
	 *
	 * @return alarm flag
	 */
	public boolean isAlarmOn() {
		return alarmFlag;
	}

	// Implement this method
	/**
	 * turn car door alarm on function
	 */
	public void turnOnAlarm() {
		alarmFlag = true;
	}

	// Implement this method
```

```
/**
 * turn car door alarm off function
 */
public void turnOffAlarm() {
	alarmFlag = false;
}

// To simulate the situation that you open a car while
// sitting in the car
//
// Step 1: Turn on the alarm
// Step 2: Unlock the car
// Step 3: Display "Look out, opening the door"
/**
 * open car door
 */
public void open() {
	turnOnAlarm();
	unlock();
	System.out.print("Look out, opening the door");
}

// To simulate the situation that you close a car while
// sitting in the car
//
// Step 1: Display "Look out, closing the door"
// Step 2: lock the car
// Step 3: Turn off the alarm
/**
 * close car door
 */
public void close() {
	System.out.print("Look out, closing the door");
	lock();
	turnOffAlarm();
}

// Implement this method
/**
 * lock car door
 */
public void lock() {
	lockFlag = true;
}
```

```
        // Implement this method
        /**
          * unlock car door
          */
        public void unlock() {
                lockFlag = false;
        }
}
```

CarPart.java //////

```
/**
  * CarPart is a car part class
  *
  * <pre>
  * CarPart door = new CarPart();
  * door.getWeight();
  * door.getPartID();
  * door.getCost();
  * </pre>
  *
  *
  * @author Peng Gao (pgaooscar@gmail.com)
  * @version 0.1 Novenmber 14 2020
  */
public class CarPart {

        /**
          * partID is the id of car part.
          */
        private int partID;

        /**
          * weight is the weight of car part.
          */
        private float weight;

        /**
          * cost is the cost of car part.
          */
        private float cost;
```

```
	/**
	 * aMethod prints "This is a car part method".
	 */
	public void aMethod() {
		System.out.println("This is a car part method");
	}

	// Implement this method
	/**
	 * get car part's weight
	 *
	 * @return car Part weight
	 */
	public float getWeight() {
		return weight;

	}

	// Implement this method
	/**
	 * get car part's ID
	 *
	 * @return car part ID
	 */
	public int getPartID() {
		return partID;
	}

	// Implement this method
	/**
	 * get car part's cost
	 *
	 * @return car part cost
	 */
	public float getCost() {
		return cost;
	}
}
```

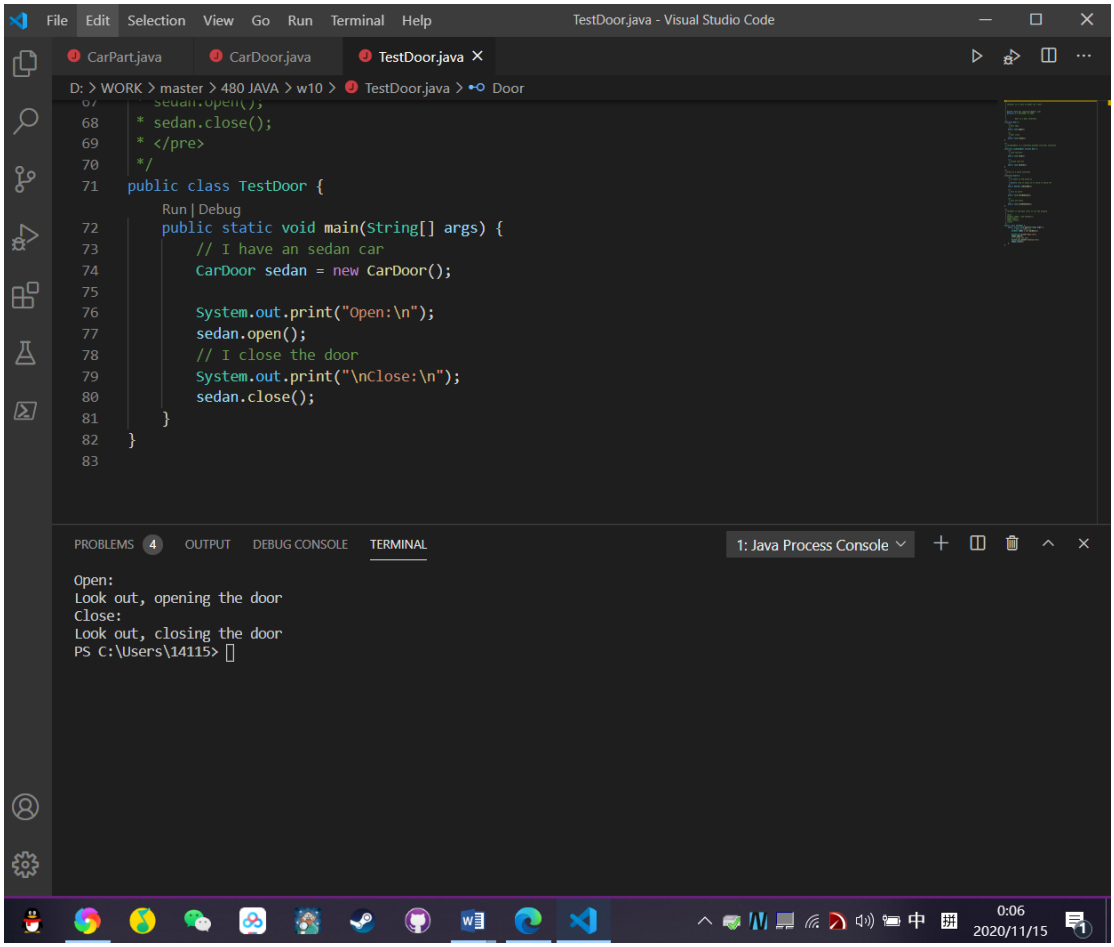
TestDoor.java//////

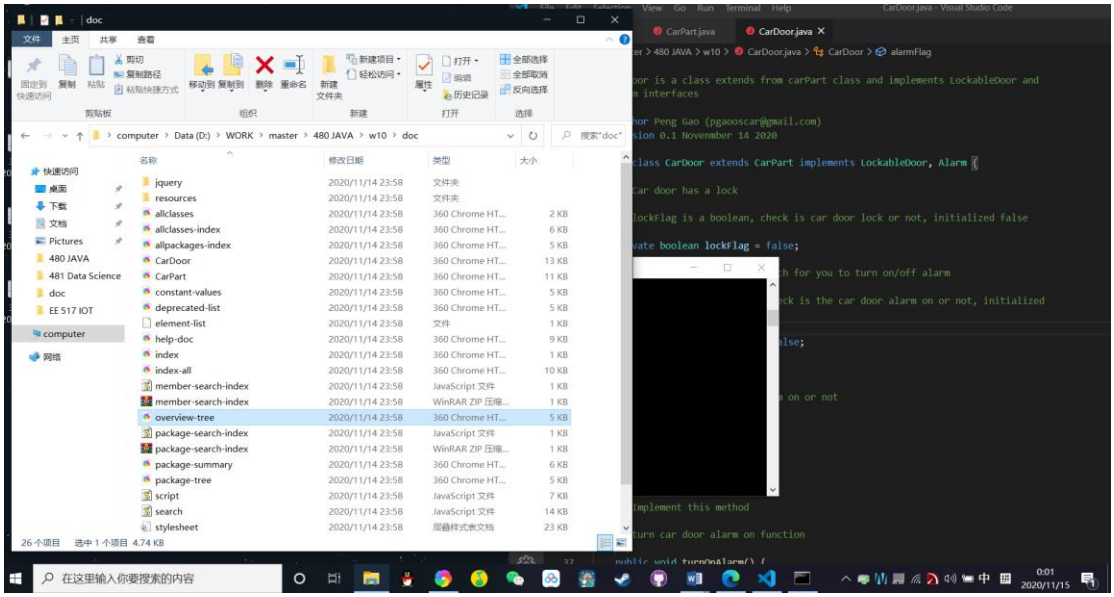
```
/**
 *
```

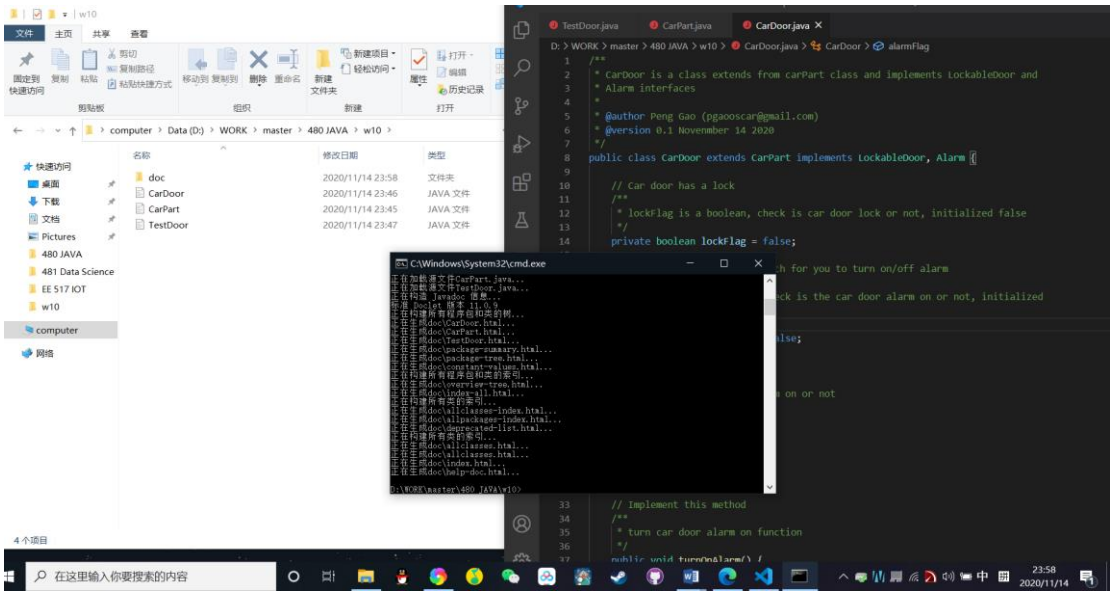
```
 * testdoor is a class to model car's door
 *
 *
 *
 * @author Peng Gao (pgaooscar@gmail.com)
 * @version 0.1 Novenmber 14 2020
 *
 *
 *                  Door is a door interface
 */
interface Door {
	/**
	 * door open
	 */
	public void open();

	/**
	 * door close
	 */
	public void close();
}

/**
 * LockableDoor is a interface extends from Door interface
 */
interface LockableDoor extends Door {
	/**
	 * lock function
	 */
	public void lock();

	/**
	 * unlock function
	 */
	public void unlock();
}

/**
 * Alarm is a alarm interface
 */
interface Alarm {
	/**
	 * to chech is the alarm on
	 *
```

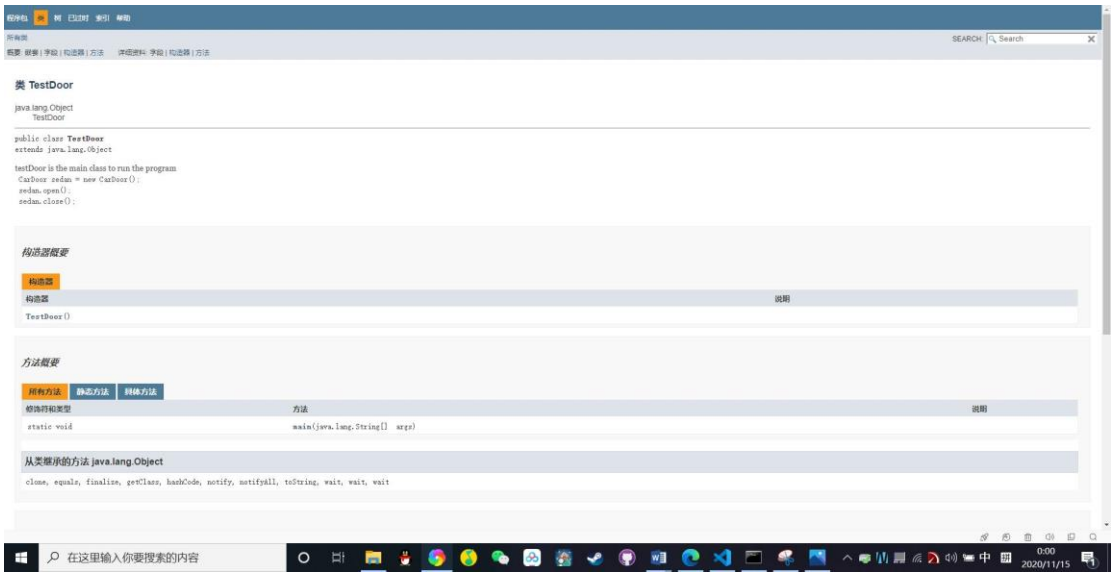
```
     * @return true if alarm on or false if alarm off
     */
    public boolean isAlarmOn();

    /**
     * turn on alarm
     */
    public void turnOnAlarm();

    /**
     * turn off alarm
     */
    public void turnOffAlarm();
}

/**
 * testDoor is the main class to run the program
 *
 * <pre>
 * CarDoor sedan = new CarDoor();
 * sedan.open();
 * sedan.close();
 * </pre>
 */
public class TestDoor {
    public static void main(String[] args) {
        // I have an sedan car
        CarDoor sedan = new CarDoor();

        System.out.print("Open:\n");
        sedan.open();
        // I close the door
        System.out.print("\nClose:\n");
        sedan.close();
    }
}
```

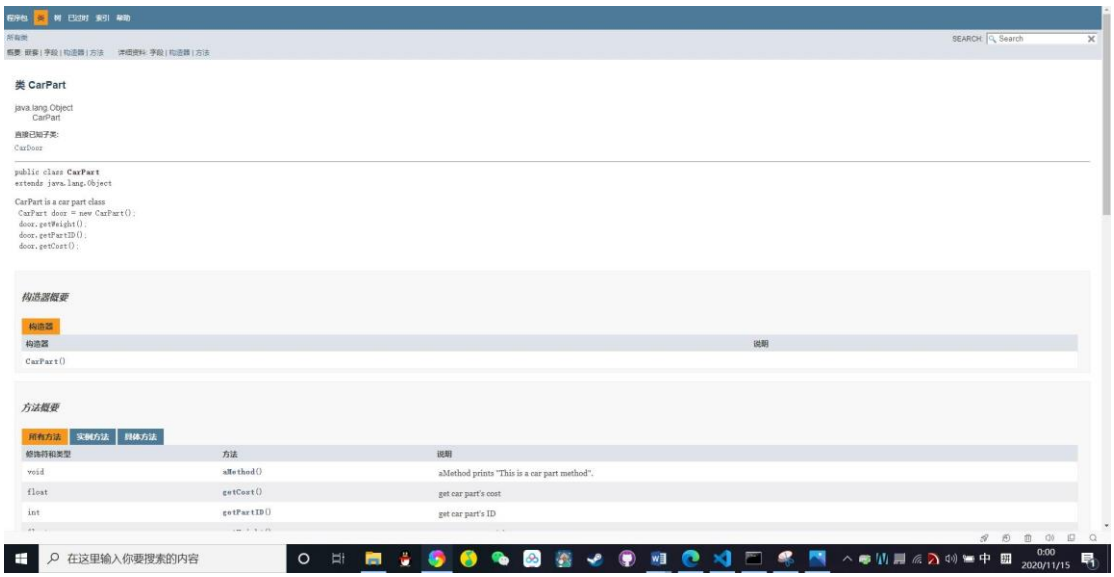
Compile the program and run the Javadoc

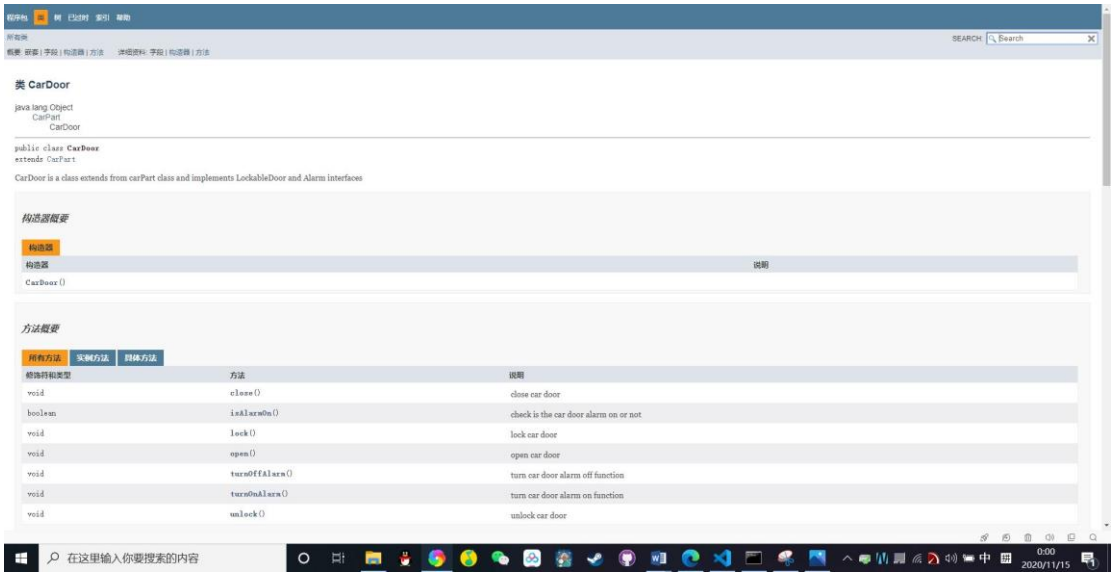

File Edit Selection View Go Run Terminal Help
TestDoor.java - Visual Studio Code
CarPart.java
CarDoor.java
TestDoor.java
D: > WORK > master > 480 JAVA > w10 > TestDoor.java > Door
* sedan.close();
* </pre>
*/
public class TestDoor {
Run | Debug
public static void main(String[] args) {
// I have an sedan car
CarDoor sedan = new CarDoor();
System.out.print("Open:\n");
sedan.open();
// I close the door
System.out.print("\nClose:\n");
sedan.close();
}
}
PROBLEMS 4
OUTPUT
DEBUG CONSOLE
TERMINAL
1: Java Process Console
Open:
Look out, opening the door
Close:
Look out, closing the door
PS C:\Users\14115>
0:06
2020/11/15


doc
computer > Data (D:) > WORK > master > 480 JAVA > w10 > doc
jquery
resources
allclasses
allclasses-index
allpackages-index
CarDoor
CarPart
constant-values
deprecated-list
element-list
help-doc
index
index-all
member-search-index
overview-tree
package-search-index
package-summary
package-tree
script
search
stylesheet
26 个项目 选中 1 个项目 4.74 KB
在这里输入你要搜索的内容
0:01
2020/11/15


w10
文件 主页 共享 查看
computer › Data (D:) › WORK › master › 480 JAVA › w10 ›
名称 修改日期 类型
doc 2020/11/14 23:58 文件夹
CarDoor 2020/11/14 23:46 JAVA 文件
CarPart 2020/11/14 23:45 JAVA 文件
TestDoor 2020/11/14 23:47 JAVA 文件
4 个项目
TestDoor.java
CarPart.java
CarDoor.java
D: > WORK > master > 480 JAVA > w10 > CarDoor.java > CarDoor > alarmFlag
/**
* CarDoor is a class extends from carPart class and implements LockableDoor and
* Alarm interfaces
*
* @author Peng Gao (pgaooscar@gmail.com)
* @version 0.1 Novenmber 14 2020
*/
public class CarDoor extends CarPart implements LockableDoor, Alarm {
// Car door has a lock
/**
* lockFlag is a boolean, check is car door lock or not, initialized false
*/
private boolean lockFlag = false;
// Implement this method
/**
* turn car door alarm on function
*/
C:\Windows\System32\cmd.exe
在加载源文件CarPart.java...
在加载源文件TestDoor.java...
在构造 Javadoc 信息...
标准 Doclet 版本 11.0.9
在构建所有程序包和类的树...
在生成doc\CarDoor.html...
在生成doc\CarPart.html...
在生成doc\TestDoor.html...
在生成doc\package-summary.html...
在生成doc\package-tree.html...
在生成doc\constant-values.html...
在构建所有程序包和类的索引...
在生成doc\overview-tree.html...
在生成doc\index-all.html...
在构建所有类的索引...
在生成doc\allclasses-index.html...
在生成doc\allpackages-index.html...
在生成doc\deprecated-list.html...
在构建所有类的索引...
在生成doc\allclasses.html...
在生成doc\allclasses.html...
在生成doc\index.html...
在生成doc\help-doc.html...
D:\WORK\master\480 JAVA\w10>
在这里输入你要搜索的内容
23:58
2020/11/14


SEARCH:
类 TestDoor
java.lang.Object
TestDoor
public class TestDoor
extends java.lang.Object
testDoor is the main class to run the program
CarDoor sedan = new CarDoor();
sedan.open();
sedan.close();
构造器概要
构造器
构造器 说明
TestDoor()
方法概要
所有方法 静态方法 具体方法
修饰符和类型 方法 说明
static void main(java.lang.String[] args)
从类继承的方法 java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
在这里输入你要搜索的内容
0:00
2020/11/15


SEARCH:
类 CarPart
java.lang.Object
CarPart
直接已知子类:
CarDoor
public class CarPart
extends java.lang.Object
CarPart is a car part class
CarPart door = new CarPart();
door.getWeight();
door.getPartID();
door.getCost();
构造器概要
构造器
构造器 说明
CarPart()
方法概要
所有方法 实例方法 具体方法
修饰符和类型 方法 说明
void aMethod() aMethod prints "This is a car part method".
float getCost() get car part's cost
int getPartID() get car part's ID
在这里输入你要搜索的内容
0:00
2020/11/15


程序包 类 树 已过时 索引 帮助
所有类
概要: 嵌套 | 字段 | 构造器 | 方法 详细资料: 字段 | 构造器 | 方法
SEARCH: Search
类 CarDoor
java.lang.Object
CarPart
CarDoor
public class CarDoor
extends CarPart
CarDoor is a class extends from carPart class and implements LockableDoor and Alarm interfaces
构造器概要
构造器
构造器 说明
CarDoor()
方法概要
所有方法 实例方法 具体方法
修饰符和类型 方法 说明
void close() close car door
boolean isAlarmOn() check is the car door alarm on or not
void lock() lock car door
void open() open car door
void turnOffAlarm() turn car door alarm off function
void turnOnAlarm() turn car door alarm on function
void unlock() unlock car door
在这里输入你要搜索的内容
0:00
2020/11/15