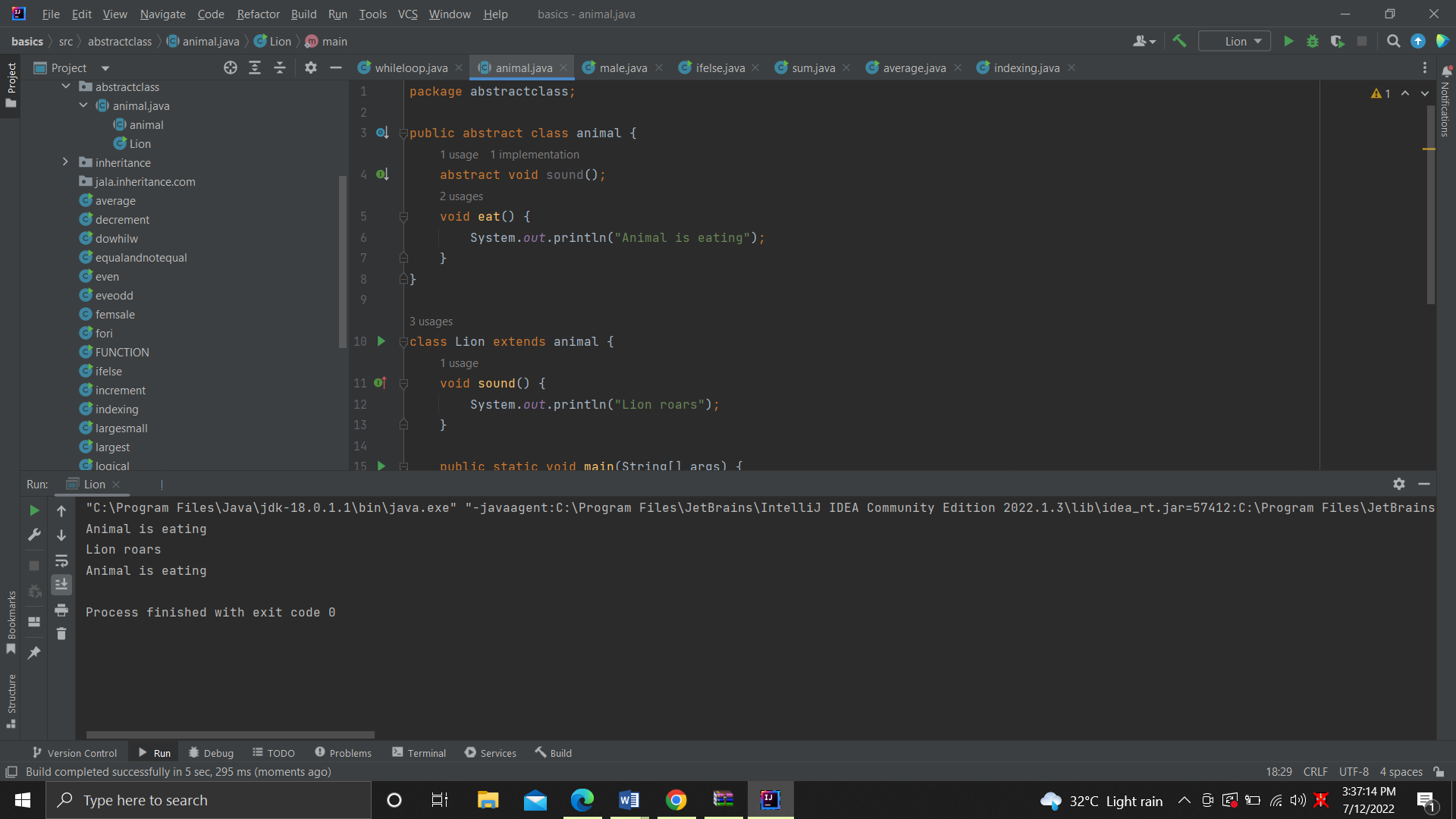
1. package jala.inheritance;  
  
public class A {  
 int n = 11;  
  
 void methodA1() {  
 System.*out*.println("This is Class A Method 1");  
 }  
  
 void methodA2() {  
 System.*out*.println("This is Class A Method 2");  
 }  
  
 void method3() {  
 System.*out*.println("This is override method - Class A");  
 }  
}  
class B extends A {  
 int n = 22;  
  
 void methodB1() {  
 System.*out*.println("This is Class B Method 1");  
 }  
  
 void methodB2() {  
 System.*out*.println("This is Class B Method 2");  
 }  
  
 @Override //override method  
 void method3() {  
 System.*out*.println("This is override method - Class B");  
 }  
}  
class C extends B {  
 int n = 33;  
  
 void methodC1() {  
 System.*out*.println("This is Class C Method 1");  
 }  
  
 void methodC2() {  
 System.*out*.println("This is Class C Method 2");  
 }  
  
 @Override  
 void method3() {  
 System.*out*.println("This is override method - Class C");  
 }  
}  
  
public class Inheritance {  
 public static void main(String[] args) {  
 A a = new A();  
 a.methodA1();  
 a.methodA2();  
 a.method3();  
  
 B b = new B();  
 b.methodB1();  
 b.methodB2();  
 b.method3();  
  
 C c = new C();  
 c.methodC1();  
 c.methodC2();  
 c.method3();  
  
 A orm;  
  
 orm = new B();  
 orm.method3();  
 //upcasting  
 orm = new C();  
 orm.method3();  
 A rtp;  
 rtp = new A();  
 System.*out*.println(rtp.n);  
 rtp = new B();  
 System.*out*.println(rtp.n);  
 rtp = new C();  
 System.*out*.println(rtp.n);  
 }

2.

Abstract class

package abstractclass;  
  
public abstract class animal {  
 abstract void sound();  
 void eat() {  
 System.*out*.println("Animal is eating");  
 }  
}  
  
class Lion extends animal {  
 void sound() {  
 System.*out*.println("Lion roars");  
 }  
  
 public static void main(String[] args) {  
 animal A = new Lion();  
 A.eat();  
 Lion L = new Lion();  
  
 L.sound();  
  
 L.eat();  
 }  
}



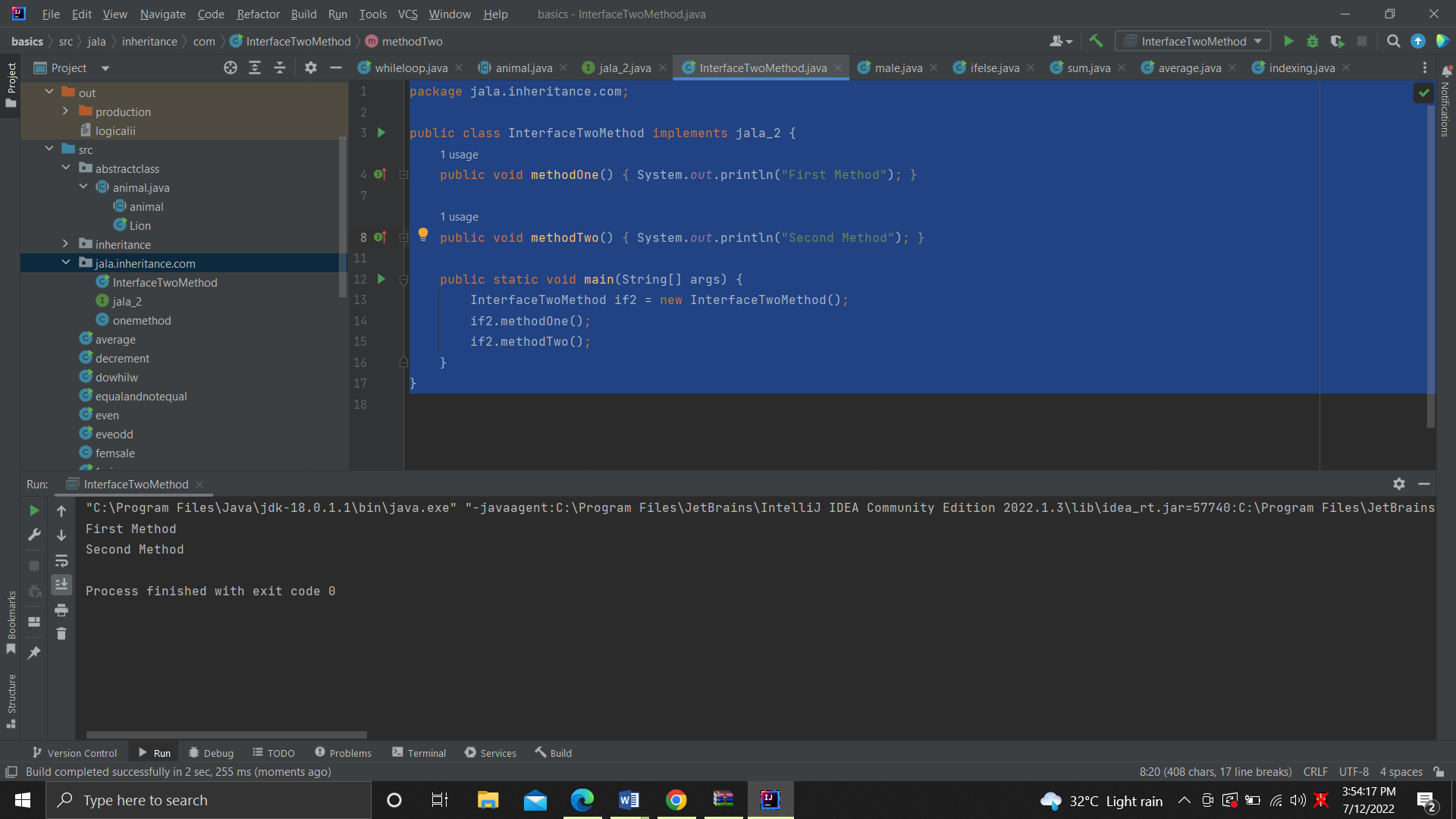
1.

package jala.inheritance.com;  
  
public interface java\_01 {  
 static void oneMethod() {  
  
 }  
}  
public class interfaceOneMethod implements java\_01 {  
  
 public static void main(String[] args) {  
 new interface if.1 = new interfaceOneMethod();  
 }

}

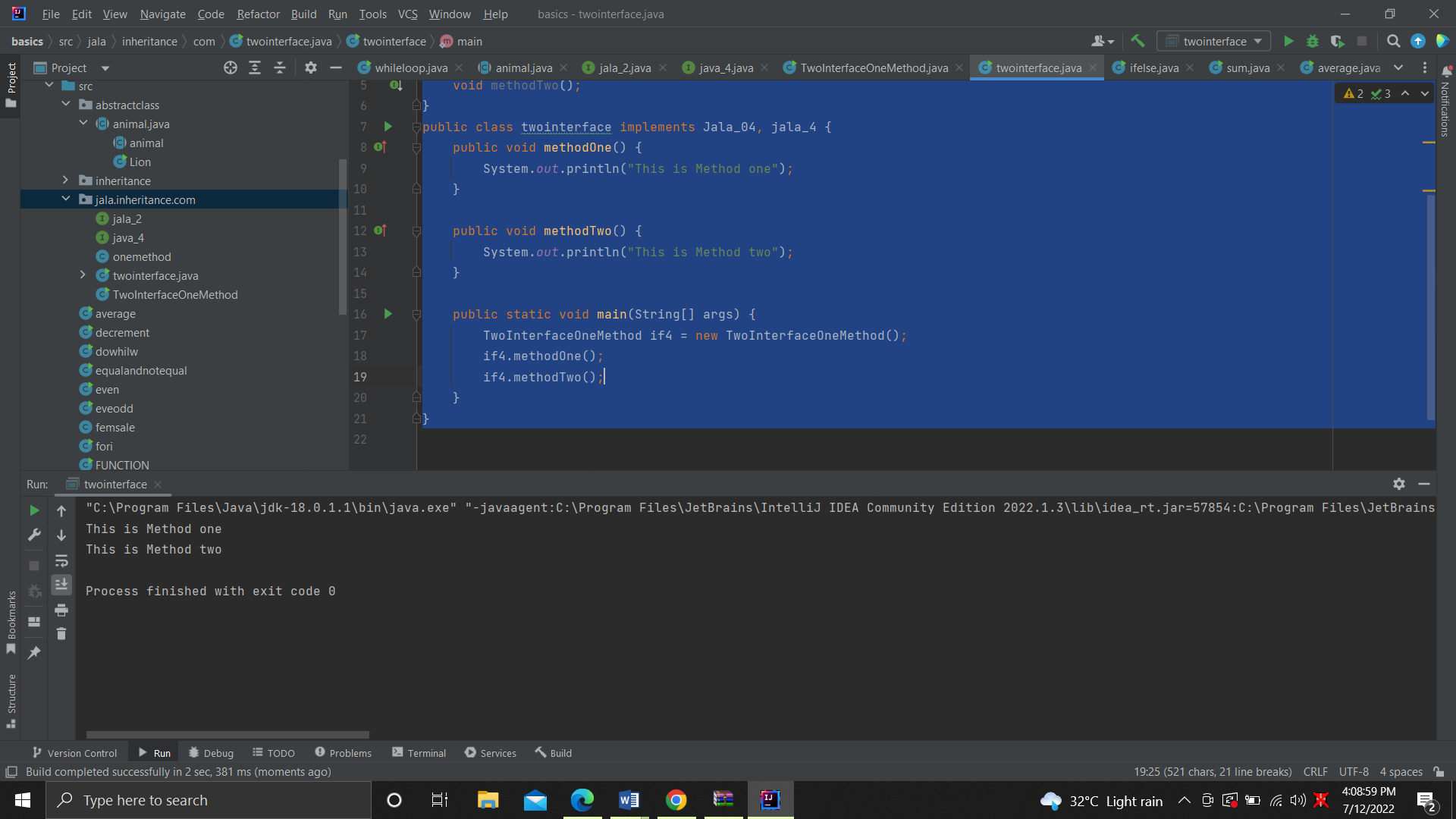
2.

package jala.inheritance.com;  
  
public class InterfaceTwoMethod implements jala\_2 {  
 public void methodOne() {  
 System.*out*.println("First Method");  
 }  
  
 public void methodTwo() {  
 System.*out*.println("Second Method");  
 }  
  
 public static void main(String[] args) {  
 InterfaceTwoMethod if2 = new InterfaceTwoMethod();  
 if2.methodOne();  
 if2.methodTwo();  
 }  
}



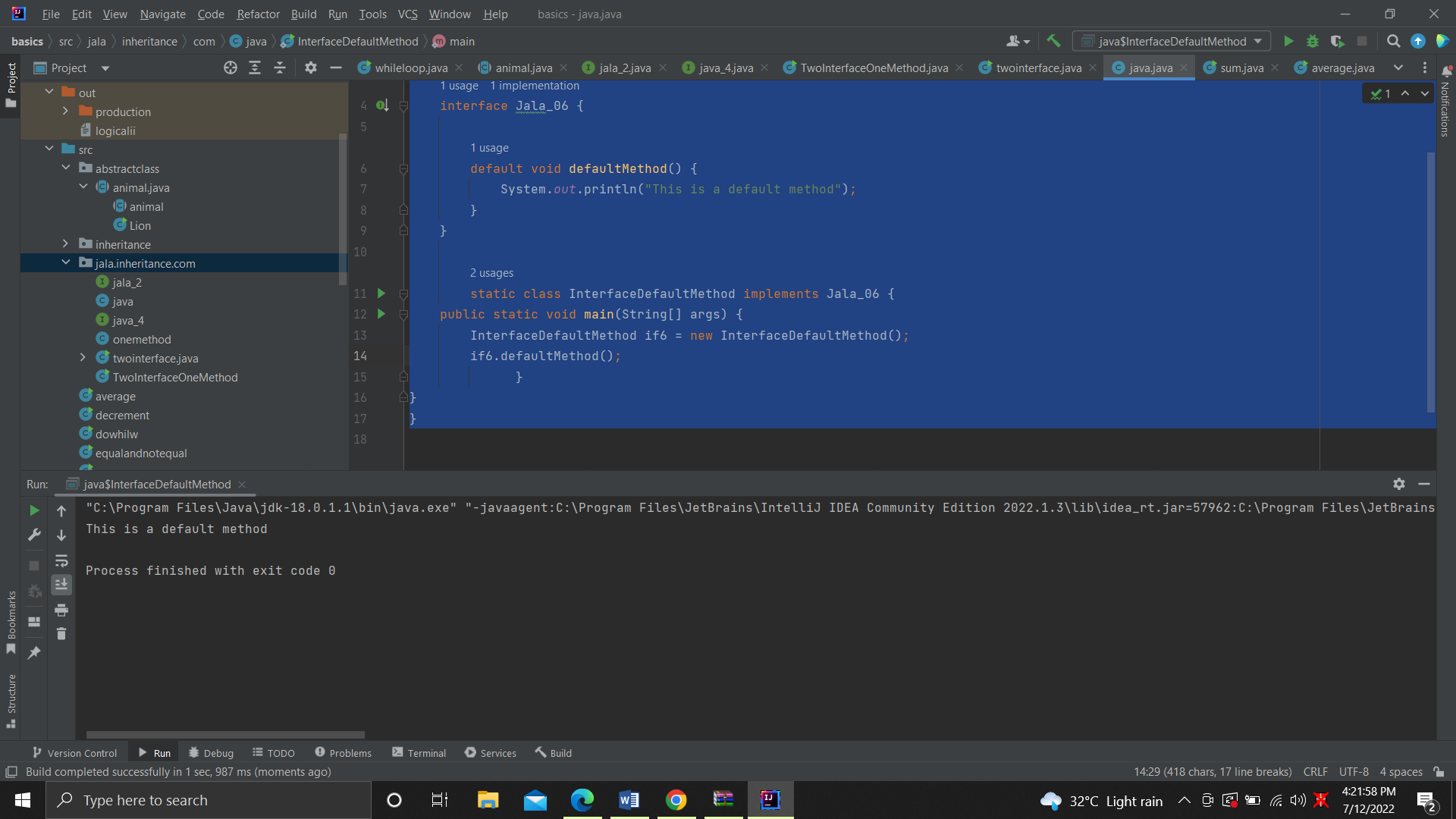
4. package jala.inheritance.com;  
interface Jala\_04 {  
 void methodOne();}  
 interface jala\_4 {  
 void methodTwo();  
}  
public class twointerface implements Jala\_04, jala\_4 {  
 public void methodOne() {  
 System.*out*.println("This is Method one");  
 }  
  
 public void methodTwo() {  
 System.*out*.println("This is Method two");  
 }  
  
 public static void main(String[] args) {  
 TwoInterfaceOneMethod if4 = new TwoInterfaceOneMethod();  
 if4.methodOne();  
 if4.methodTwo();  
 }  
}

Output



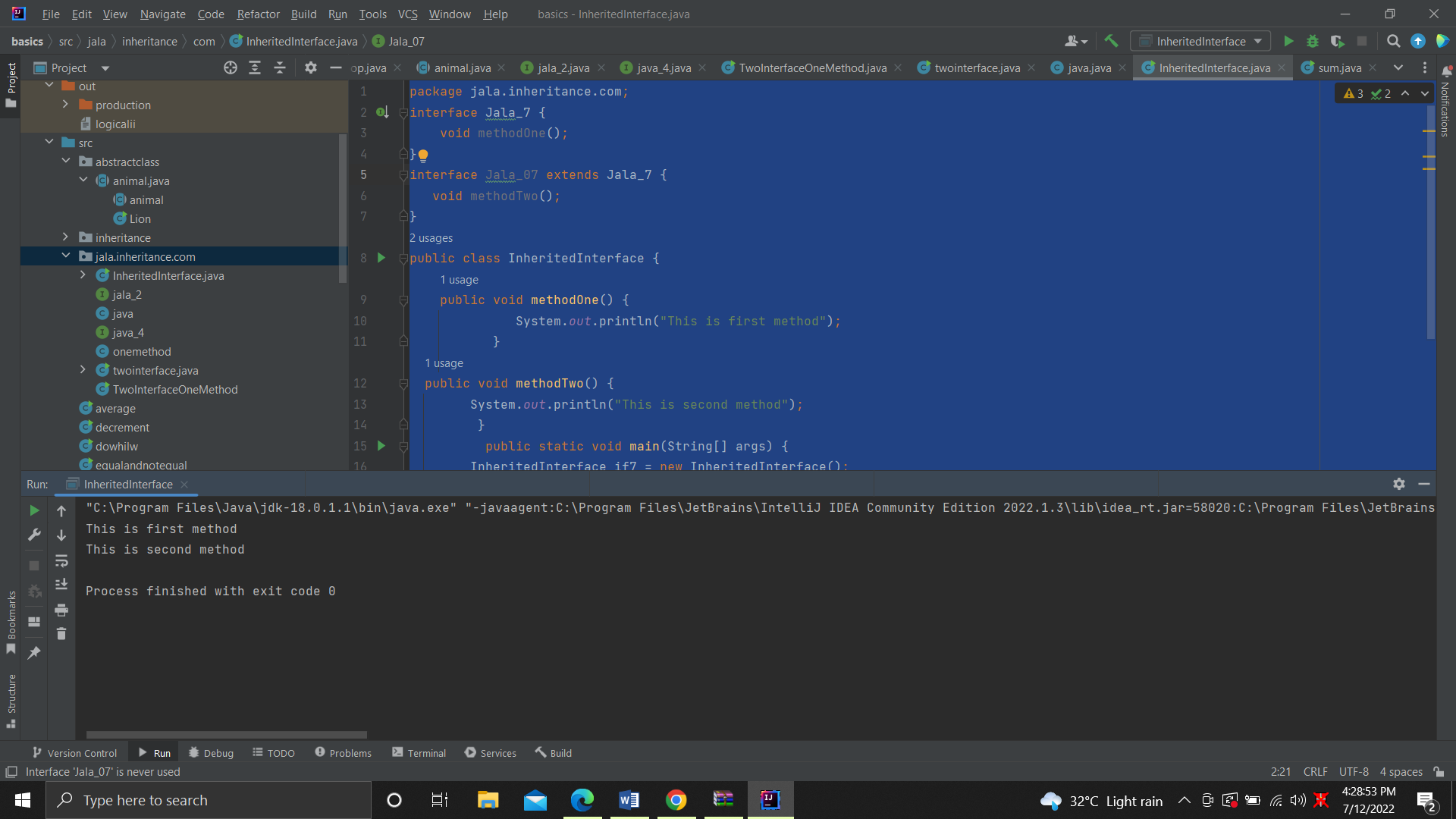
5. package jala.inheritance.com;  
interface Jala\_5 {  
 void sameMethod();  
}  
interface Jala\_05 {  
 void samemethod();  
}  
 class TwoInterfaceSameMethod implements Jala\_5, Jala\_05 {  
public class samemethod {  
System.*out*.println("this is same method");  
 }  
 public static void main(String[] args) {  
 TwoInterfaceSameMethod if5 = new TwoInterfaceSameMethod();  
 if5.sameMethod();  
}  
}

6. package jala.inheritance.com;  
  
public class java {  
 interface Jala\_06 {  
  
 default void defaultMethod() {  
 System.*out*.println("This is a default method");  
 }  
 }  
  
 static class InterfaceDefaultMethod implements Jala\_06 {  
 public static void main(String[] args) {  
 InterfaceDefaultMethod if6 = new InterfaceDefaultMethod();  
 if6.defaultMethod();  
 }  
}  
}



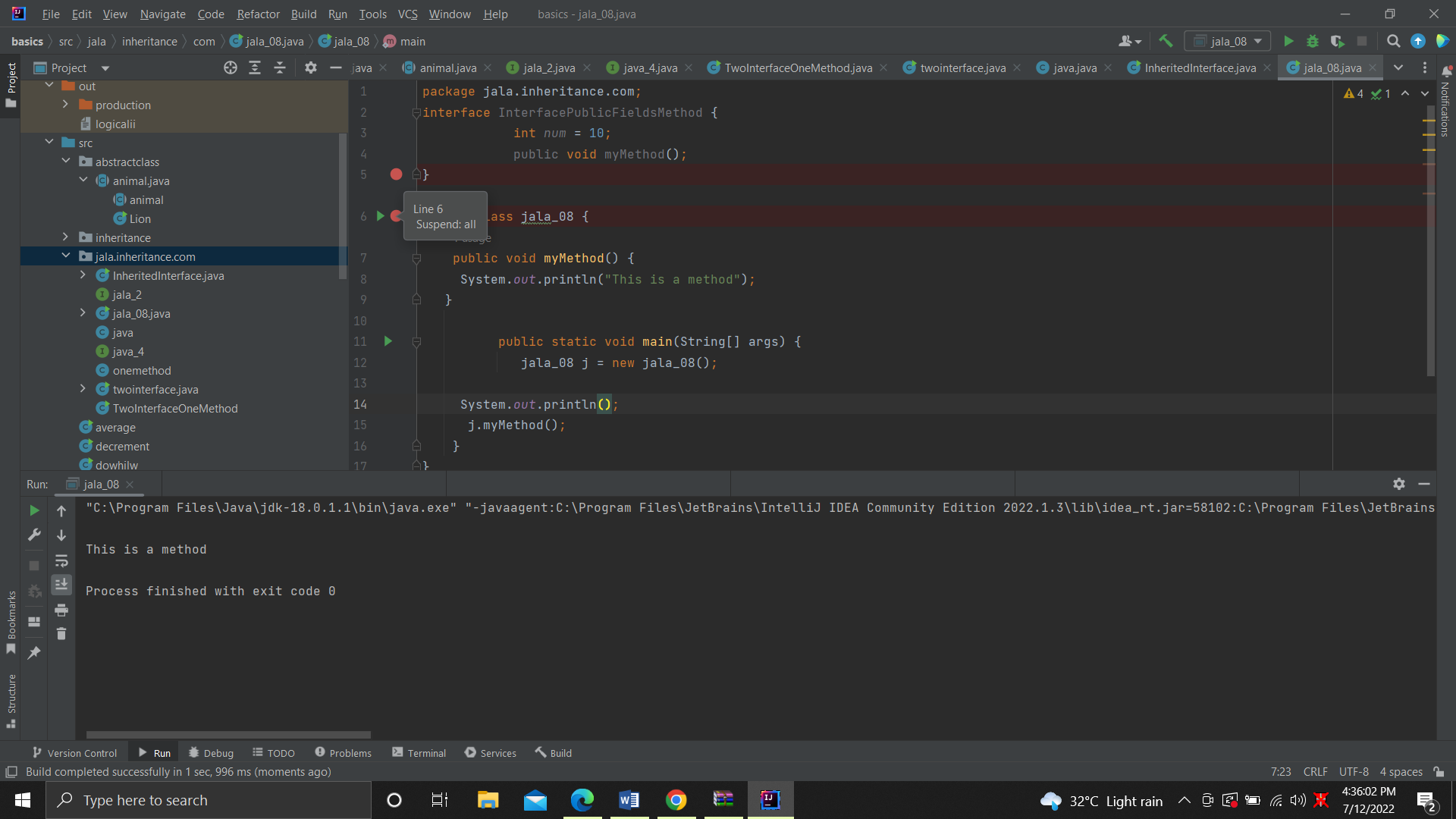
7. package jala.inheritance.com;  
interface Jala\_7 {  
 void methodOne();  
}  
interface Jala\_07 extends Jala\_7 {  
 void methodTwo();  
}  
public class InheritedInterface {  
 public void methodOne() {  
 System.*out*.println("This is first method");  
 }  
 public void methodTwo() {  
 System.*out*.println("This is second method");  
 }  
 public static void main(String[] args) {  
 InheritedInterface if7 = new InheritedInterface();  
 if7.methodOne();  
 if7.methodTwo();  
 }  
}

Output



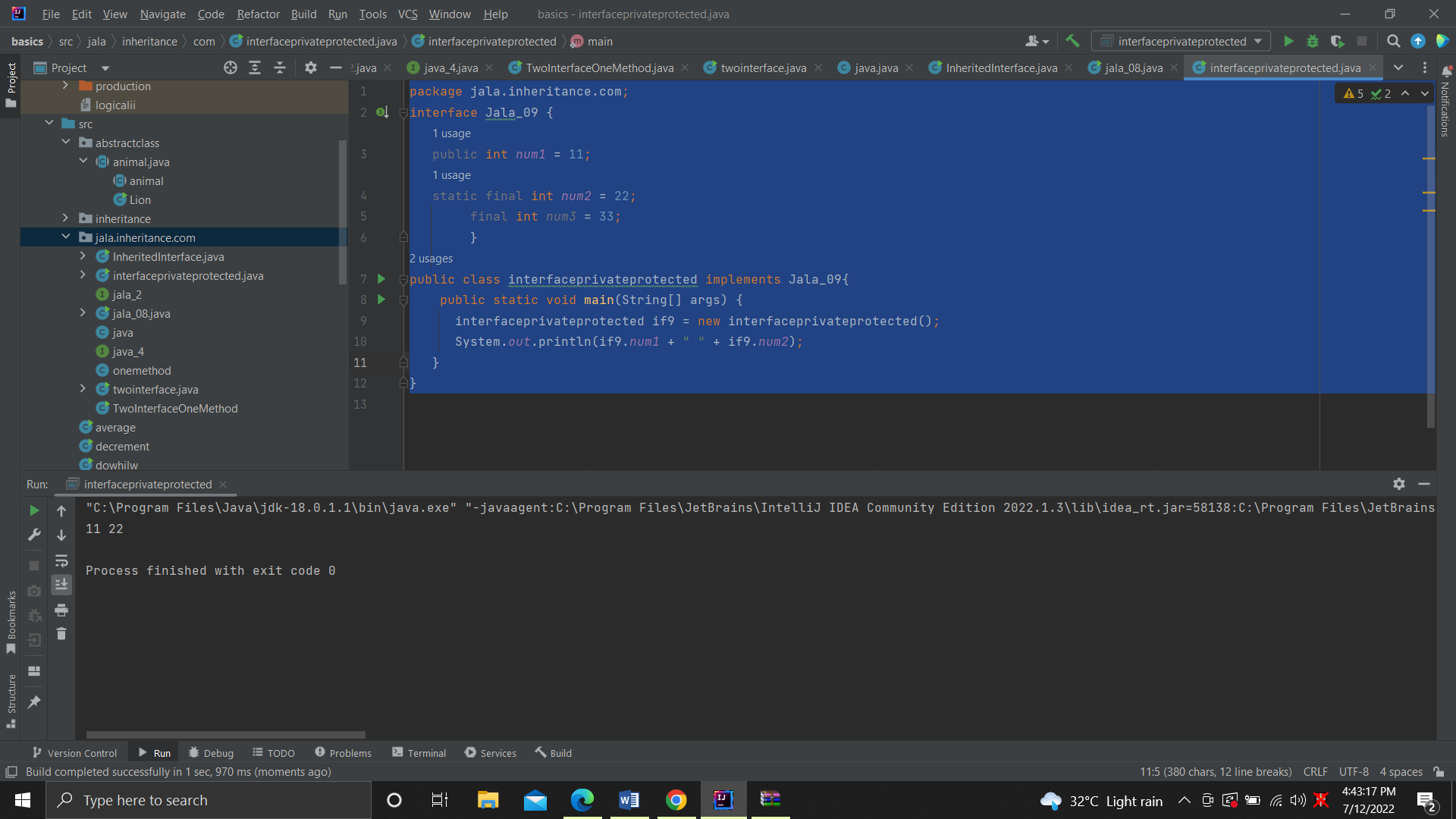
8. package jala.inheritance.com;  
interface InterfacePublicFieldsMethod {  
 int *num* = 110;  
 public void myMethod();  
}  
public class jala\_08 {  
 public void myMethod() {  
 System.*out*.println("This is a method");  
 }  
  
 public static void main(String[] args) {  
 jala\_08 j = new jala\_08();  
  
 System.*out*.println(num);  
 j.myMethod();  
 }  
}

Output



9,10,11.

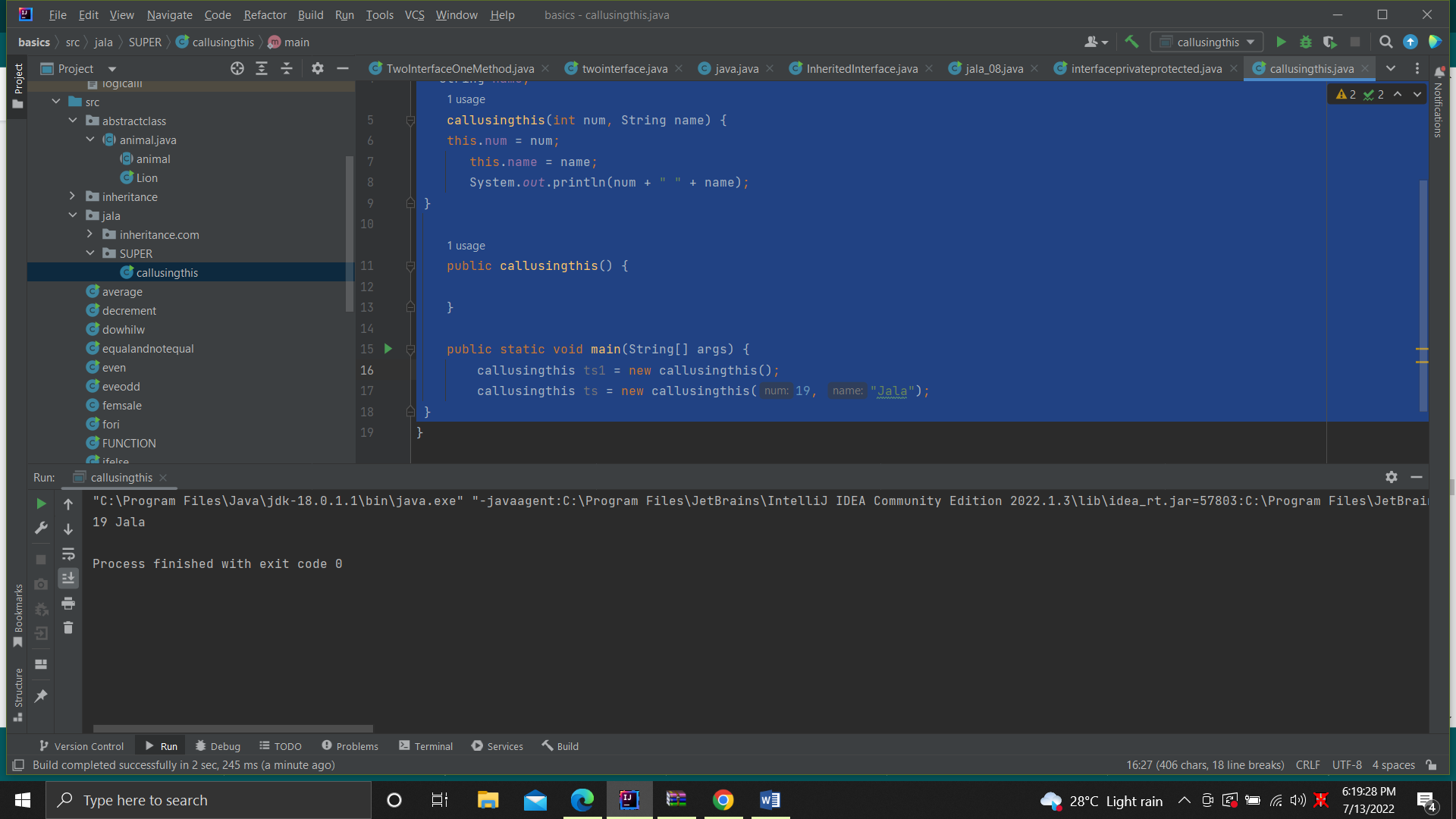
package jala.inheritance.com;  
interface Jala\_09 {  
 public int *num1* = 11;  
 static final int *num2* = 22;  
 final int *num3* = 33;  
 }  
public class interfaceprivateprotected implements Jala\_09{  
 public static void main(String[] args) {  
 interfaceprivateprotected if9 = new interfaceprivateprotected();  
 System.*out*.println(if9.*num1* + " " + if9.*num2*);  
 }  
}



This and super

package jala.SUPER;  
public class callusingthis {  
 int num;  
 String name;  
 callusingthis(int num, String name) {  
 this.num = num;  
 this.name = name;  
 System.*out*.println(num + " " + name);  
 }  
  
 public callusingthis() {  
  
 }  
  
 public static void main(String[] args) {  
  
 }

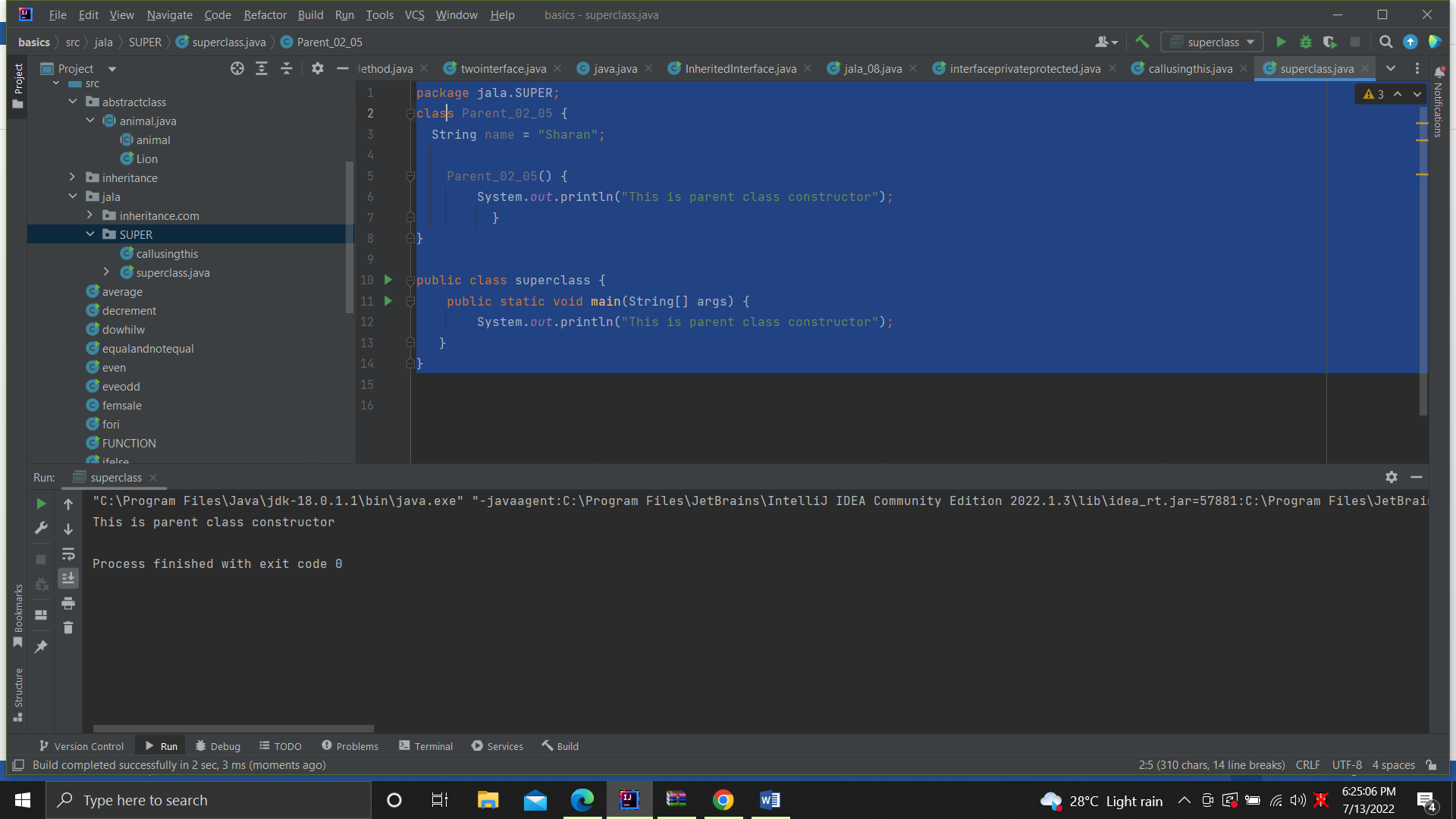
output



2.

package jala.SUPER;  
class Parent\_02\_05 {  
 String name = "Sharan";  
  
 Parent\_02\_05() {  
 System.*out*.println("This is parent class constructor");  
 }  
}  
  
public class superclass {  
 public static void main(String[] args) {  
 System.*out*.println("This is parent class constructor");  
 }  
}

output



3.

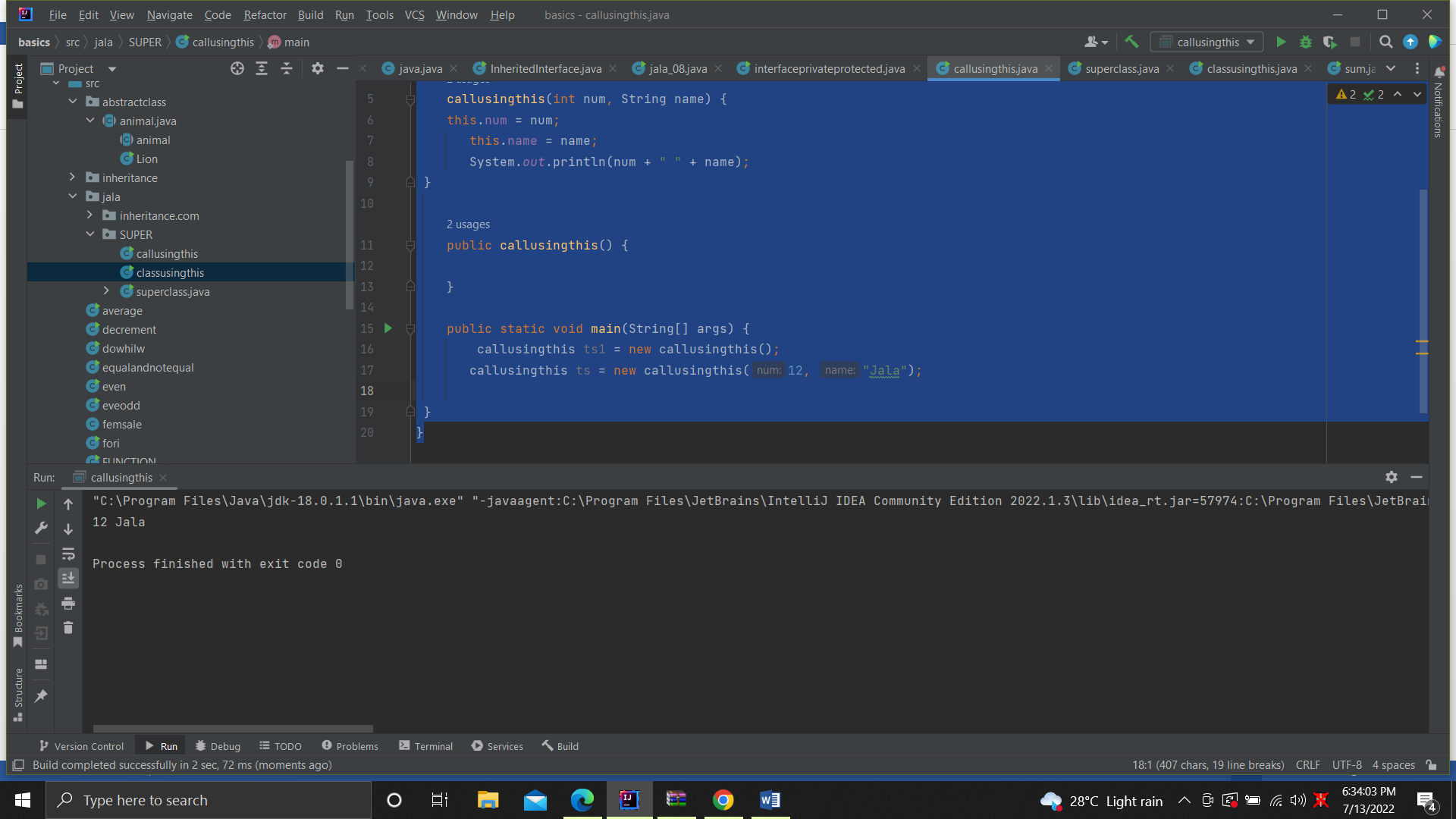
package SUPER;  
  
public class callusingthis {  
 int *num*;  
 String *name*;  
  
 callusingthis(int i, String sharan) {  
 this(11, "Sharan");  
 }  
  
 public static void main(String[] args) {  
 System.*out*.println("the num is"*num*);  
 System.*out*.println("the age is"*name*);  
 }  
}

output

4.

package jala.SUPER;  
public class callusingthis {  
 int num;  
 String name;  
 callusingthis(int num, String name) {  
 this.num = num;  
 this.name = name;  
 System.*out*.println(num + " " + name);  
 }  
  
 public callusingthis() {  
  
 }  
  
 public static void main(String[] args) {  
 callusingthis ts1 = new callusingthis();  
 callusingthis ts = new callusingthis(12, "Jala");  
  
 }  
}

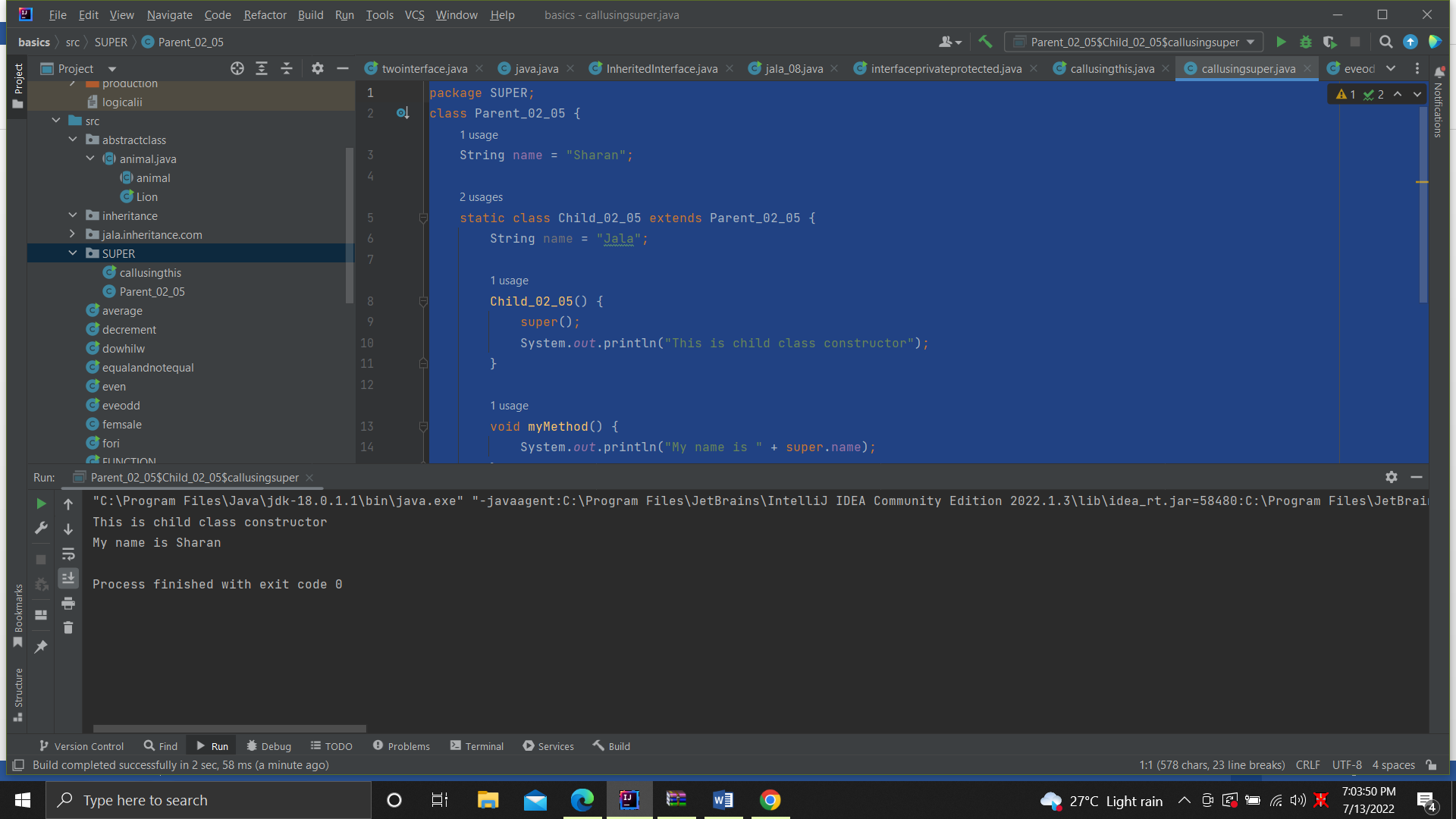
output



5.

package SUPER;  
class Parent\_02\_05 {  
 String name = "Sharan";  
  
 static class Child\_02\_05 extends Parent\_02\_05 {  
 String name = "Jala";  
  
 Child\_02\_05() {  
 super();  
 System.*out*.println("This is child class constructor");  
 }  
  
 void myMethod() {  
 System.*out*.println("My name is " + super.name);  
 }  
  
 public static class callusingsuper {  
 public static void main(String[] args) {  
 Child\_02\_05 c = new Child\_02\_05();  
 c.myMethod();  
 }  
 }  
 }  
}

output



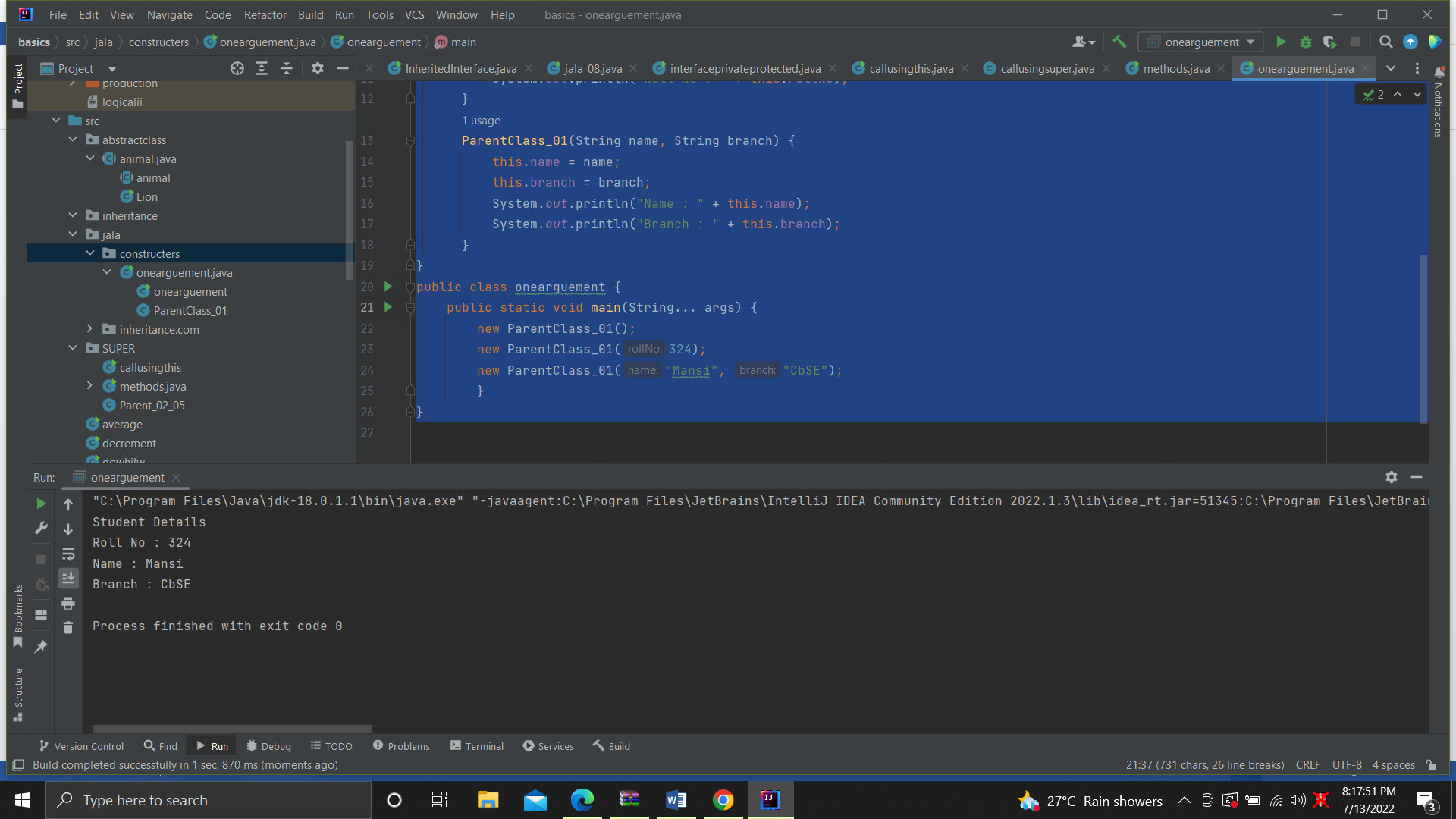
6. package SUPER;  
abstract class Parent\_06 {  
 void method1() {  
 System.*out*.println("Used this() and super() in methods");  
 }  
  
 protected void method2() {  
 }  
}  
  
 class Child\_06 extends Parent\_06 {  
 void method3() {  
 super.method2();  
 }  
}  
public class methods {  
 public static void main(String[] args) {  
 Child\_06 c = new Child\_06();  
 c.method3();  
 }  
}

Constructers

1.

package jala.constructers;  
class ParentClass\_01 {  
 int rollNo;  
 String name;  
 String branch;  
 ParentClass\_01() {  
 System.*out*.println("Student Details");  
 }  
 ParentClass\_01(int rollNo) {  
 this.rollNo = rollNo;  
 System.*out*.println("Roll No : " + this.rollNo);  
 }  
 ParentClass\_01(String name, String branch) {  
 this.name = name;  
 this.branch = branch;  
 System.*out*.println("Name : " + this.name);  
 System.*out*.println("Branch : " + this.branch);  
 }  
}  
public class onearguement {  
 public static void main(String... args) {  
 new ParentClass\_01();  
 new ParentClass\_01(324);  
 new ParentClass\_01("Mansi", "CbSE");  
 }  
}

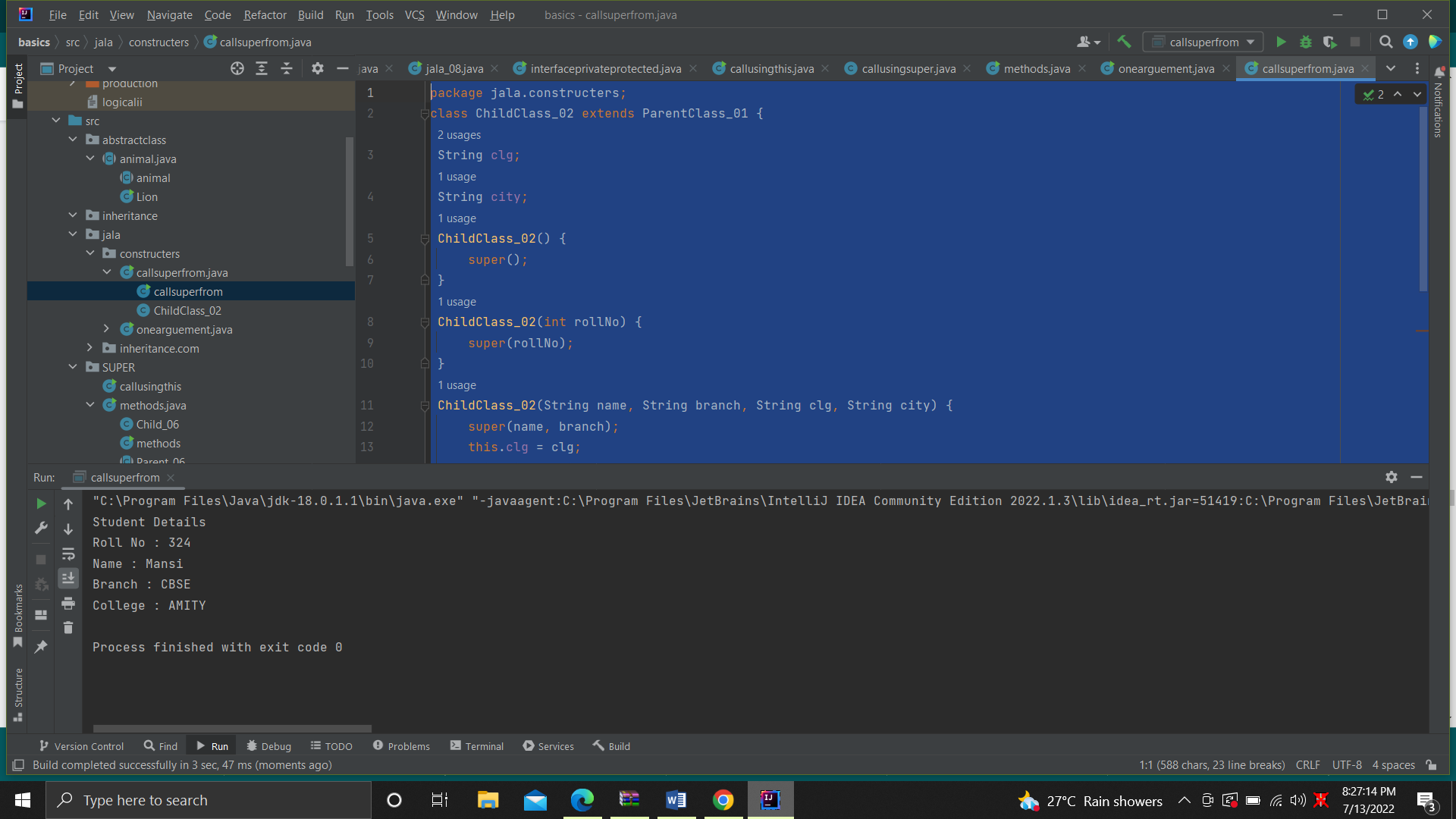
output



2.

package jala.constructers;  
class ChildClass\_02 extends ParentClass\_01 {  
 String clg;  
 String city;  
 ChildClass\_02() {  
 super();  
 }  
 ChildClass\_02(int rollNo) {  
 super(rollNo);  
 }  
 ChildClass\_02(String name, String branch, String clg, String city) {  
 super(name, branch);  
 this.clg = clg;  
 this.city = city;  
 System.*out*.println("College : " + this.clg);  
 }  
}  
public class callsuperfrom {  
 public static void main(String[] args) {  
 new ChildClass\_02();  
 new ChildClass\_02(324);  
 new ChildClass\_02("Mansi", "CBSE", "AMITY", "KOLKATA");  
 }  
}

output



3.

0