Main.java

*Local Inner Class*

**public** **class** Main {

**int** i;

**static** **int** *j* = 2;

**public** Main() {

i = 1;

}

**public** **static** **void** main(String[] args) {

System.***out***.println("Output for useLocalA(30)");

System.***out***.println("Value of Main.j = " + *j*); //Prints: Value of Main.j = 2

*useLocalA*(30);

System.***out***.println("Value of Main.j = " + *j*); //Prints: Value of Main.j = 200

System.***out***.println("\nOutput for useLocalB(30)");

Main m = **new** Main();

System.***out***.println("Value of m.i = " + m.i + ", value of Main.j = " + *j*); //Prints: Value of m.i = 1, value of Main.j = 200

m.useLocalB(30);

System.***out***.println("Value of m.i = " + m.i + ", value of Main.j = " + *j*); //Prints: Value of m.i = 1000, value of Main.j = 2000

}

**private** **static** **void** useLocalA(**final** **int** p) { //Static method

**int** x = 40;

**final** **int** y = 50;

**int** z;

z = 70;

//p = 90; //if p is changed at this line then compliation fails at line where aP = p;

//Local variable p defined in an enclosing scope must be final or effectively final

**class** A {

**int** aI;

**int** aJ;

**int** aP;

**int** aX;

**int** aY;

**int** aZ;

//static int aS; //Compilation fails. The field aS cannot be declared static in a non-static inner type, unless initialized with a constant expression

**static** **final** **int** ***aFS*** = 800;

**private** **void** updateValues() {

//aI = i; //Compilation fails. Cannot make a static reference to the non-static field i

aJ = *j*;

aP = p;

aX = x;

aY = y;

aZ = z;

//i = 100; //Cannot make a static reference to the non-static field i

*j* = 200;

//p = 300; //Compilation fails. Local variable p defined in an enclosing scope must be final or effectively final

//x = 400; //Compilation fails. Local variable x defined in an enclosing scope must be final or effectively final

//y = 500; //Compilation fails. Local variable y defined in an enclosing scope must be final or effectively final

//z = 600; //Compilation fails. Local variable z defined in an enclosing scope must be final or effectively final

System.***out***.println("aJ = " + aJ + ", aP = " + aP + ", aX = " + aX + ", aY = " + aY + ", aZ = " + aZ + ", Main.j = " + *j* + ", aFS = " + ***aFS***);

//Prints: aJ = 2, aP = 30, aX = 40, aY = 50, aZ = 70, Main.j = 200, aFS = 800

}

//private static void updateValuesStatic() {} //Compilation fails. The method updateValuesStatic cannot be declared static; static methods can only be declared in a static or top level type

}

A a = **new** A();

a.updateValues();

}

**private** **void** useLocalB(**int** p) { //Instance method

//int p can also be written as final int p

**int** x = 40;

**final** **int** y = 50;

**int** z = 60;

z = 70;

**class** B {

**int** bI;

**int** bJ;

**int** bP;

**int** bX;

**int** bY;

**int** bZ;

//static int bS; //Compilation fails. The field bS cannot be declared static in a non-static inner type, unless initialized with a constant expression

**static** **final** **int** ***bFS*** = 800;

**private** **void** updateValues() {

bI = i;

bJ = *j*;

bP = p;

bX = x;

bY = y;

//bZ = z; //Compilation fails. Local variable z defined in an enclosing scope must be final or effectively final

i = 1000;

*j* = 2000;

//p = 300; //Compilation fails. The final local variable p cannot be assigned, since it is defined in an enclosing type

//x = 400; //Compilation fails. Local variable x defined in an enclosing scope must be final or effectively final

//y = 500; //Compilation fails. Local variable y defined in an enclosing scope must be final or effectively final

//z = 600; //Compilation fails. Local variable z defined in an enclosing scope must be final or effectively final

System.***out***.println("bI = " + bI + ", bJ = " + bJ + ", bP = " + bP + ", bX = " + bX + ", bY = " + bY + ", Main.this.i = " + i + ", Main.j = " + *j* + ", bFS = " + ***bFS***);

//Prints: bI = 1, bJ = 200, bP = 30, bX = 40, bY = 50, Main.this.i = 1000, Main.j = 2000, bFS = 800

}

//private static void updateValuesStatic() {} //Compilation fails. The method updateValuesStatic cannot be declared static; static methods can only be declared in a static or top level type

}

B b = **new** B();

b.updateValues();

}

}