

The screenshot shows an IDE window with several tabs: PairOfShoes.java, PairOfShoesTest.java, FarmAnimal.java, Food.java, and C. The 'Source' tab is active, displaying the code for the Chicken class. The code is as follows:

```
1  /*
2   * To change this license header, choose License Headers in Project
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package ec.edu.espe.farmsystem.abstraction;
7
8  /**
9   *
10   * @author Darian M. Martinez ESPE DCCO
11   */
12  public class Chicken extends FarmAnimal{
13
14      /*@Override
15      public float computeEarningMoney(float product, int quantity) {
16          System.out.println("Compute earning money in chicken");
17          return 0.0F;
18      }
19
20      @Override
```

A mouse cursor is hovering over the 'Chicken' class name, which has triggered a context menu with the following options:

- Implement all abstract methods
- Make class Chicken abstract

The screenshot shows an IDE window with several tabs: Test.java, JsonManager.java, PairOfShoes.java, PairOfShoesTest.java, and C. The 'Source' tab is active, displaying the code for the Cow class. The code is as follows:

```
1  /*
2   * To change this license header, choose License Headers in Project
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6  package ec.edu.espe.farmsystem.abstraction;
7
8  /**
9   *
10   * @author Darian M. Martinez ESPE DCCO
11   */
12  public class Cow extends FarmAnimal{
```

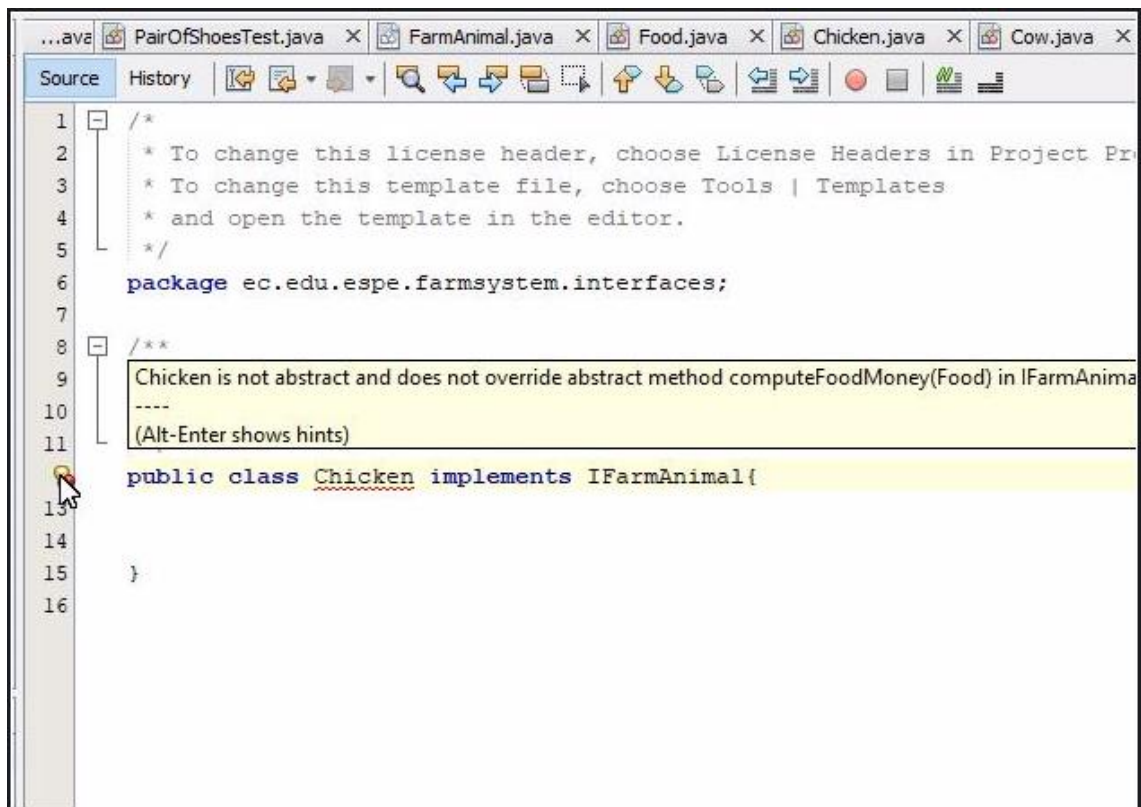
A mouse cursor is hovering over the 'Cow' class name, which has triggered a context menu with the following options:

- Implement all abstract methods
- Make class Cow abstract
- Implement unimplemented abstract methods of ec.edu.espe.farmsystem.abstraction.
- Create Subclass
- Create Test Class [JUnit in Test Packages]
- Create Test Class [Selenium in Test Packages]
- Create Test Class [TestNG in Test Packages]

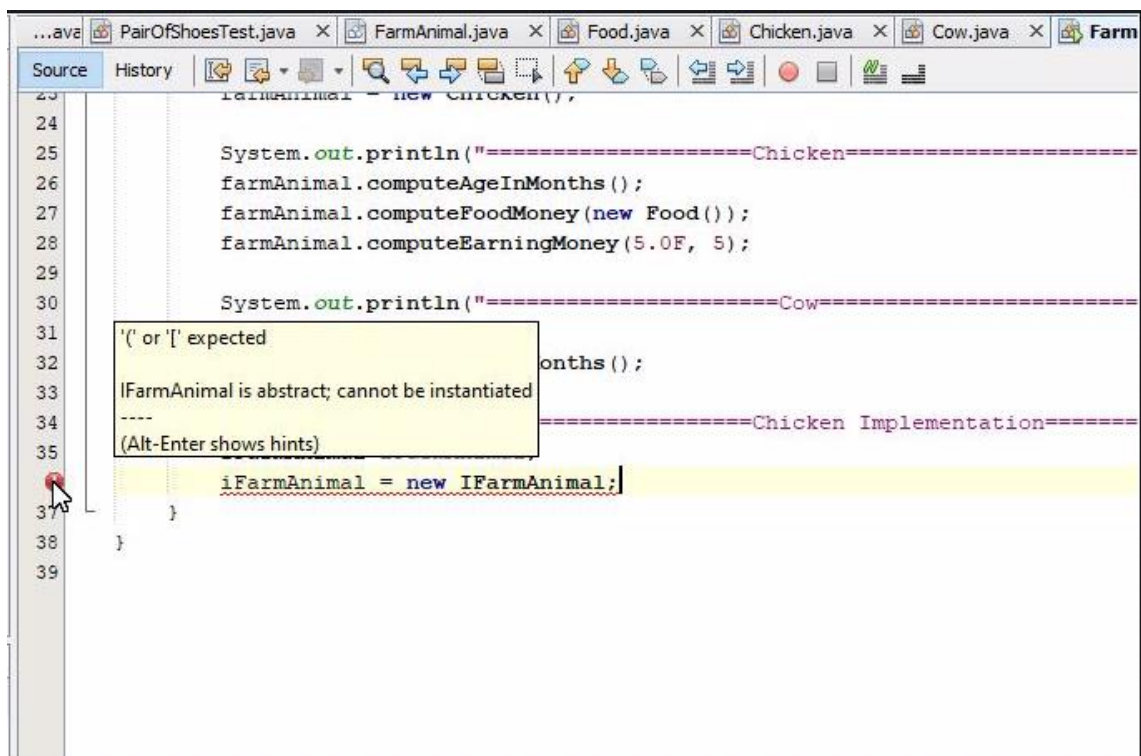
```
...ava PairOfShoes.java X PairOfShoesTest.java X FarmAnimal.java X Food.java X Chicken.java
Source History
1  /*
2   * To change this license header, choose License Headers in Project Properties
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package ec.edu.espe.farmsystem.view;
7
8   import ec.edu.espe.farmsystem.abstraction.FarmAnimal;
9
10  /**
11   *
12   * @author Darian M. Martinez ESPE DCCO
13   */
14  public class FarmSystem {
15      FarmAnimal is abstract; cannot be instantiated
16      -----
17      (Alt-Enter shows hints)
18      public FarmSystem(String[] args) {
19          farmAnimal = new FarmAnimal();
20      }
21  }
```

```
...ava PairOfShoes.java X PairOfShoesTest.java X FarmAnimal.java X Food.java X Chicken.java X Cow.java X FarmSystem.java X
Source History
8  /**
9   *
10  * @author Darian M. Martinez ESPE DCCO
11  */
12  public class Cow extends FarmAnimal{
13
14      @Override
15      public float computeEarningMoney(float product, int quantity) {
16          System.out.println("Compute earning money for cow");
17          return 3.0F;
18      }
19
20      @Override
21      public float computeFoodMoney(Food food) {
22          System.out.println("Compute food money for cow");
23          return 0.0F;
24      }
25
26      public int computeAgeInMonths(){
27          System.out.println("own compute age in months for cow");
28          return 5;
29      }
30  }
31
```

```
Output - POO836223FarmSystem (run) X
run:
=====Chicken=====
Computing age in months of the FarmAnimal
Compute food money in chicken
Compute earning money in chicken
=====Cow=====
own compute age in months for cow
BUILD SUCCESSFUL (total time: 0 seconds)
```



```
1  /*
2   * To change this license header, choose License Headers in Project Properties.
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6   package ec.edu.espe.farmsystem.interfaces;
7
8   /**
9    * Chicken is not abstract and does not override abstract method computeFoodMoney(Food) in IFarmAnimal
10   * ----
11   * (Alt-Enter shows hints)
12   */
13   public class Chicken implements IFarmAnimal{
14
15   }
16
```



```
23   IFarmAnimal = new Chicken();
24
25   System.out.println("=====Chicken=====");
26   farmAnimal.computeAgeInMonths();
27   farmAnimal.computeFoodMoney(new Food());
28   farmAnimal.computeEarningMoney(5.0F, 5);
29
30   System.out.println("=====Cow=====");
31
32   IFarmAnimal = new IFarmAnimal();
33   IFarmAnimal.computeAgeInMonths();
34   IFarmAnimal.computeFoodMoney(new Food());
35   IFarmAnimal.computeEarningMoney(5.0F, 5);
36   System.out.println("=====Chicken Implementation=====");
37
38   }
39
```

```
...ave FarmAnimal.java x Food.java x Chicken.java x Cow.java x FarmSystem.java x IFarmAnimal.java x Chicken.java x
Source History
23 farmAnimal = new Chicken();
24
25 System.out.println("====Chicken====");
26 farmAnimal.computeAgeInMonths();
27 farmAnimal.computeFoodMoney(new Food());
28 farmAnimal.computeEarningMoney(5.0F, 5);
29
30 System.out.println("====Cow====");
31 farmAnimal = new Cow();
32 farmAnimal.computeAgeInMonths();
33 incompatible types: Chicken cannot be converted to IFarmAnimal
34 ----
35 (Alt-Enter shows hints)
36 iFarmAnimal = new Chicken();
37 }
38
39
```

```
...ave FarmAnimal.java x Food.java x Chicken.java x Cow.java x FarmSystem.java x IFarmAnimal.java x Chicken.java x
Source History
23 farmAnimal = new Chicken();
24
25 System.out.println("====Chicken====");
26 farmAnimal.computeAgeInMonths();
27 farmAnimal.computeFoodMoney(new Food());
28 farmAnimal.computeEarningMoney(5.0F, 5);
29
30 System.out.println("====Cow====");
31 farmAnimal = new Cow();
32 farmAnimal.computeAgeInMonths();
33
34 System.out.println("====Chicken Implementation====");
35 IFarmAnimal iFarmAnimal;
36 iFarmAnimal = new ec.edu.espe.farmsystem.interfaces.Chicken();
37 }
38
39
```

Activar W

```
...ave PairOfShoesTest.java x FarmAnimal.java x Food.java x Chicken.java x Cow.java x FarmSystem.java x IFarmAnimal.java x Chicken.java x
Source History
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
FarmAnimal = new Chicken();

System.out.println("====Chicken====");
farmAnimal.computeAgeInMonths();
farmAnimal.computeFoodMoney(new Food());
farmAnimal.computeEarningMoney(5.0F, 5);

System.out.println("====Cow====");
farmAnimal = new Cow();
farmAnimal.computeAgeInMonths();

System.out.println("====Chicken Implementation====");
IFarmAnimal iFarmAnimal;
iFarmAnimal = new ec.edu.espe.farmsystem.interfaces.Chicken();
iFarmAnimal.computeEarningMoney(4.0F, 5);
iFarmAnimal.computeFoodMoney(new Food());
}
}

Output - POO836223FarmSystem (run) x
```

```
Output - POO836223FarmSystem (run) x
run:
Darian M. Martinez ESPE DCCO
====Chicken====
Computing age in months of the FarmAnimal
Compute food money in chicken
Compute earning money in chicken
====Cow====
own compute age in months for cow
====Chicken Implementation====
Compute earning money for Chicken implements IFarmAnimal
Compute food money for Chicken implements IFarmAnimal
BUILD SUCCESSFUL (total time: 0 seconds)
```