

assignment3.2Version control

Current File

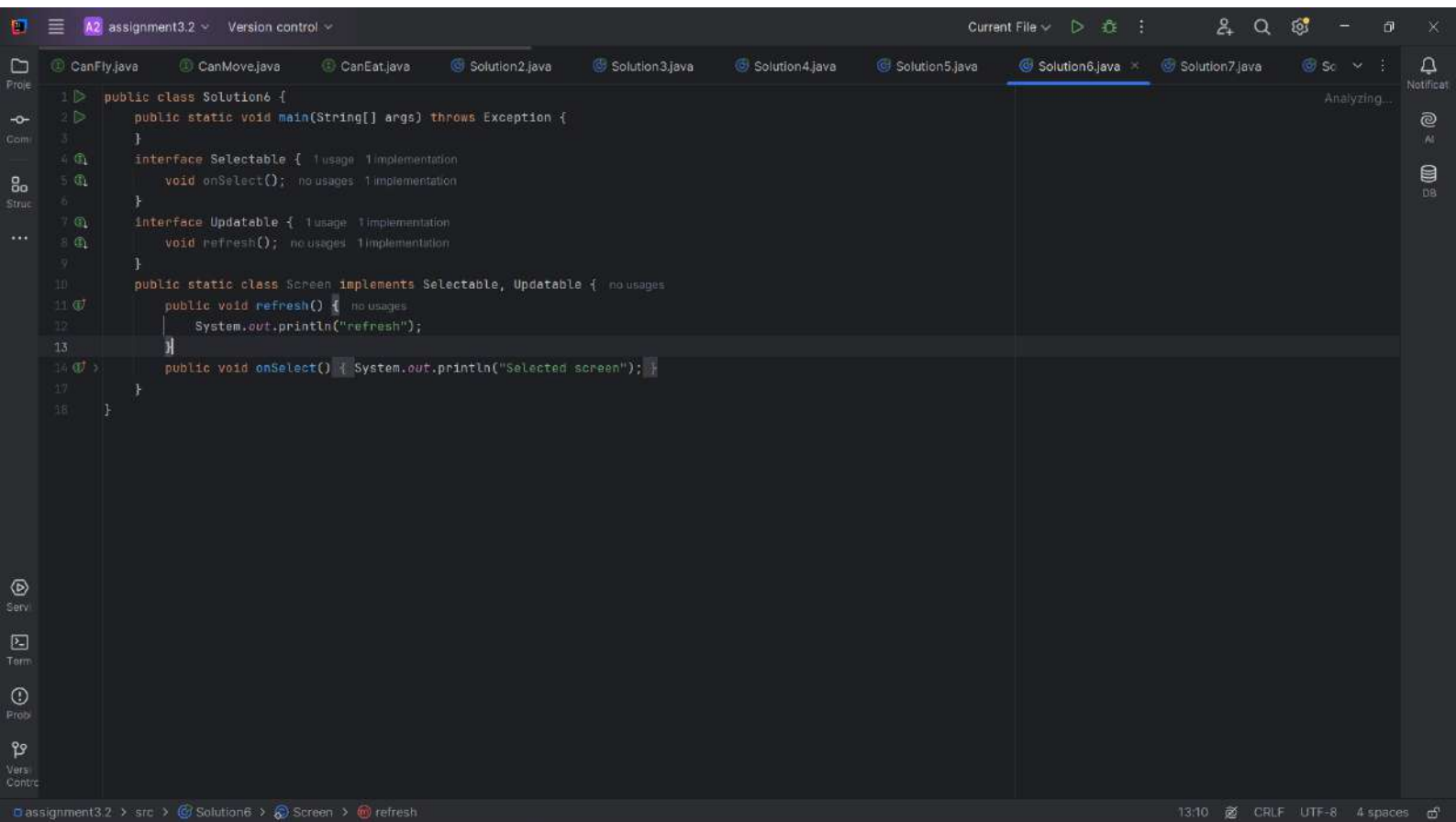
CanFly.javaCanMove.javaCanEat.javaSolution2.javaSolution3.javaSolution4.javaSolution5.javaSolution6.javaSolution7.javaSc

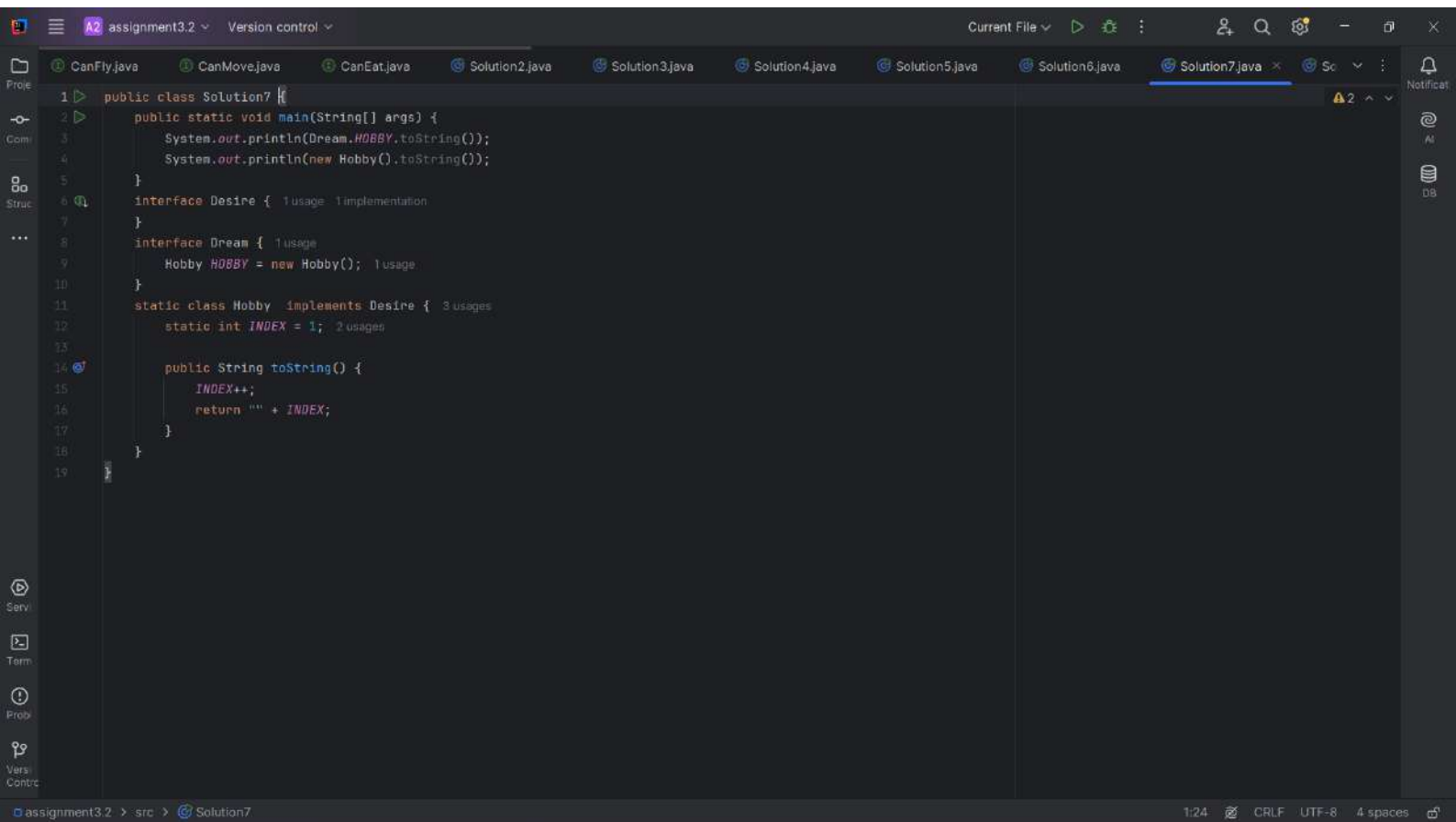
Project

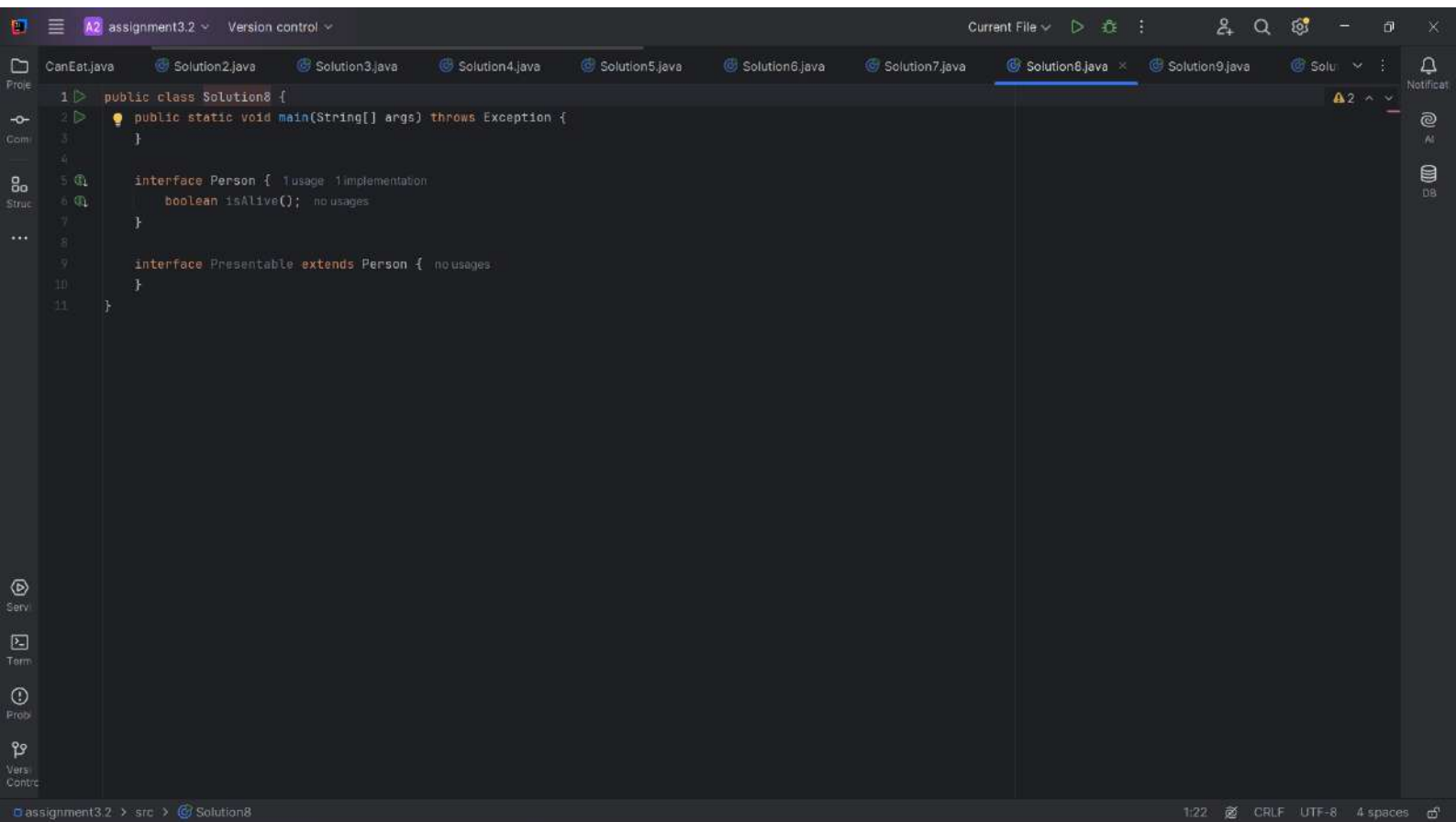
CanFly.javaCanMove.javaCanEat.javaSolution2.javaSolution3.javaSolution4.javaSolution5.javaSolution6.javaSolution7.javaSc

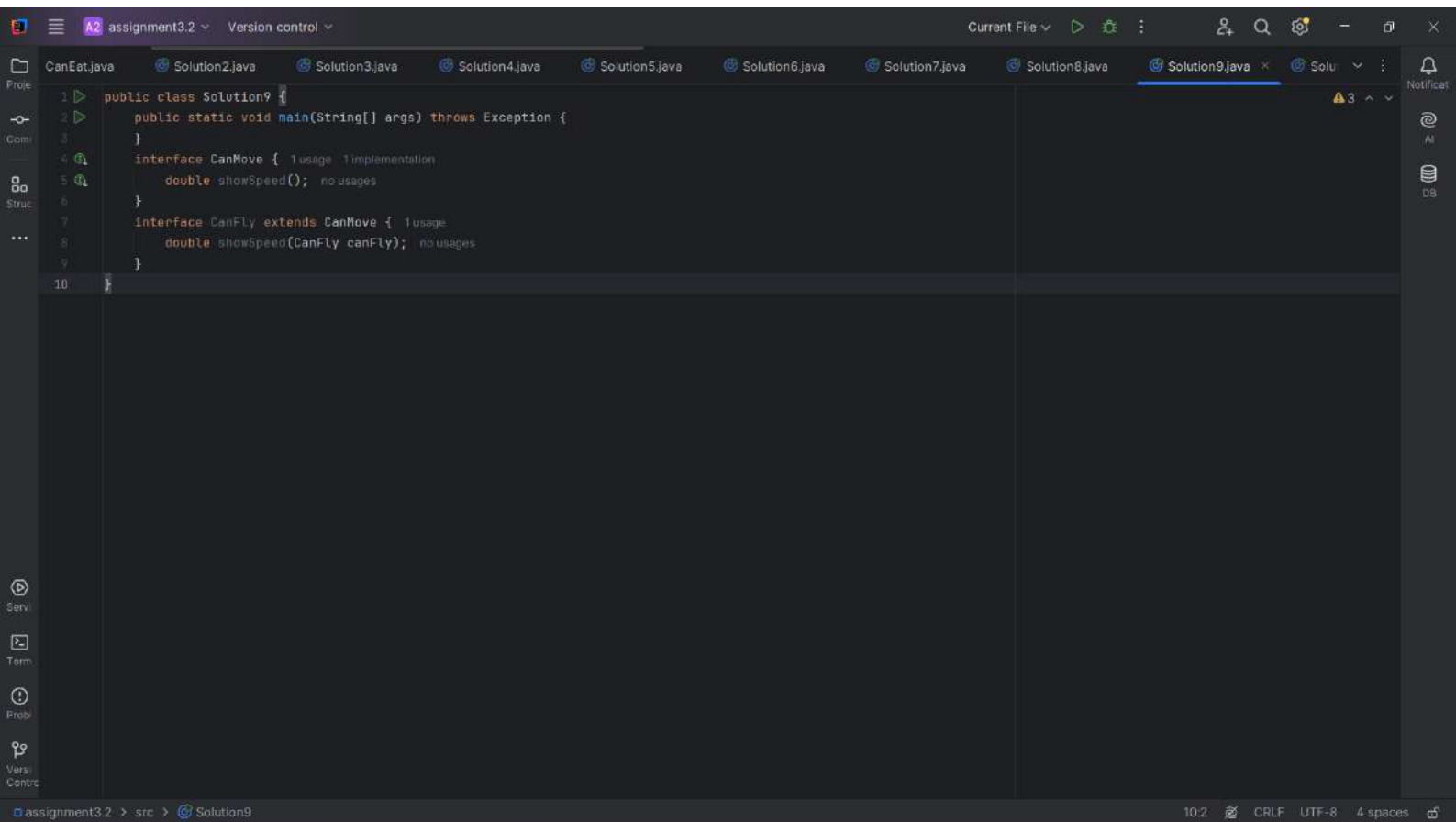
1 public class Solution5 {
2 public static void main(String[] args) throws Exception {
3 BeerLover beelover = new BeerLover();
4 System.out.println(beelover.isReadyToGoHome());
5 }
6 public interface Drinker { 1 usage 2 implementations
7 void askForMore(String message); no usages 1 implementation
8 void sayThankYou(); no usages 1 implementation
9 boolean isReadyToGoHome(); 1 usage 1 implementation
10 }
11 public interface Alcoholic extends Drinker { 1 usage 1 implementation
12 boolean READY_TO_GO_HOME = false; 1 usage
13 void sleepOnTheFloor(); no usages 1 implementation
14 }
15
16 public static class BeerLover implements Alcoholic { 2 usages
17 public void askForMore(String message) { no usages
18 }
19 public void sayThankYou() { no usages
20 }
21 public boolean isReadyToGoHome() { return READY_TO_GO_HOME; }
22 public void sleepOnTheFloor() { no usages
23 }
24 }
25 }
26 }
27 }

1:1 CRLF UTF-8 4 spaces









assignment3.2Version control

Current File

JavaSolution3.javaSolution4.javaSolution5.javaSolution6.javaSolution7.javaSolution8.javaSolution9.javaSolution10.javaSolution11

12345678910111213141516171819202122232425262728293031

```
public class Solution10 {  
    public static void main(String[] args) throws Exception {  
    }  
    interface Person { 5 usages 4 implementations  
        void use(Person person ); no usages 2 implementations  
        void startToWork(); 1 usage 2 implementations  
    }  
    interface HasManagementPotential { 1 usage 2 implementations  
        boolean inspiresOthersToWork(); no usages 1 implementation  
    }  
    interface Secretary extends Person { 1 usage 1 implementation  
    }  
    interface Boss extends Person, HasManagementPotential{ 1 usage 1 implementation  
    }  
    class Manager implements Boss { no usages  
        public void use(Person person ) { no usages  
            person.startToWork();  
        }  
        public void startToWork() { 1 usage  
        }  
        public boolean inspiresOthersToWork() { no usages  
            return true;  
        }  
    }  
    class Subordinate implements Secretary { no usages  
        public void use(Person person ) { no usages  
        }  
        public void startToWork() { 1 usage  
        }  
    }  
}
```

6

11

1:1 CRLF UTF-8 4 spaces

assignment3.2Version control

Current File

Solution4.javaSolution5.javaSolution6.javaSolution7.javaSolution8.javaSolution9.javaSolution10.javaSolution11.javaSolution12.java

12345678910111213141516171819202122232425262728293031323334

```
public class Solution11 {  
    public static void main(String[] args) {}  
    public interface Movable { 3 usages 3 implementations  
        void move(); no usages 3 implementations  
    }  
    public interface Edible { 2 usages 2 implementations  
        void beEaten(); no usages 2 implementations  
    }  
    public interface Eat { 2 usages 2 implementations  
        void eat(); no usages 2 implementations  
    }  
    class Cat implements Movable, Edible, Eat { no usages  
        @Override no usages  
        public void eat() {}  
        @Override no usages  
        public void beEaten() {}  
        @Override no usages  
        public void move() {}  
    }  
    class Dog implements Movable, Eat { no usages  
        @Override no usages  
        public void eat() {}  
    }  
        @Override no usages  
        public void move() {}  
    }  
    }  
    class Mouse implements Movable, Edible { no usages  
        @Override no usages  
        public void beEaten() {}  
        @Override no usages  
        public void move() {}  
    }  
}
```

1:26CRLFUTF-84 spaces

assignment3.2 ▾ Version control ▾

Current File ▾

Solution8.java Solution9.java Solution10.java Solution11.java Solution12.java x Solution13.java Solution14.java Solution15.java Solution16 ▾

1 public class Solution12 {
2 public static void main(String[] args) {
3 System.out.println(new Today(WeatherType.CLOUDY));
4 System.out.println(new Today(WeatherType.FOGGY));
5 System.out.println(new Today(WeatherType.FREEZING));
6
7
8 static class Today implements Weather { 3 usages
9 private String type; 2 usages
10
11 Today(String type) { 3 usages
12 this.type = type;
13 }
14
15 @Override 1 usage
16 public String getWeatherType() {
17 return type;
18 }
19
20 @Override
21 public String toString() { return String.format("Today it will be %s", this.getWeatherType()); }
22 }
23 }
24
25
26
27
28

Notification

assignment3.2 > src > Solution12 > main 2:45 CRLF UTF-8 4 spaces

assignment3.2 ▾ Version control ▾

Current File ▾

Solution8.java Solution9.java Solution10.java Solution11.java Solution12.java Solution13.java × Solution14.java Solution15.java Solution16 ▾

1 public class Solution13 {
2 public static void main(String[] args) throws Exception {
3 System.out.println(Matrix.NEO);
4 }
5 static class Matrix { 1 usage
6 public static DBObject NEO = new User().initializeIdAndName(id: 1, name: "Neo"); 1 usage
7 }
8 interface DBObject { 4 usages 1 implementation
9 DBObject initializeIdAndName(long id, String name); 1 usage 1 implementation
10 }
11 static class User implements DBObject { 1 usage
12 long id; 2 usages
13 String name; 2 usages
14 @Override 1 usage
15 public DBObject initializeIdAndName(long id, String name) {
16 this.id = id;
17 this.name = name;
18 return this;
19 }
20
21 @Override
22 public String toString() { return String.format("The user's name is %s, id = %d", name, id); }
23 }
24 }
25 }
26 }

Project

Comm

Struc

...

Search

Term

Prob

Versi

Conte

Notification

AI

DB

assignment3.2 > src > Solution13 26/2 CRLF UTF-8 4 spaces

assignment3.2Version control

Current File

Solution8.javaSolution9.javaSolution10.javaSolution11.javaSolution12.javaSolution13.javaSolution14.javaSolution15.javaSolution16

1public class Solution14 {
2 public static void main(String[] args) {
3 System.out.println(Dream.HOBBY.toString());
4 System.out.println(new Hobby().INDEX);
5 }
6 interface Desire { 1 usage 1 implementation
7 }
8 interface Dream { 2 usages 1 implementation
9 Hobby HOBBY = new Hobby(); 1 usage
10 }
11 static class Hobby implements Desire, Dream { 3 usages
12 static int INDEX = 1; 3 usages
13 @Override
14 public String toString() {
15 INDEX++;
16 return "" + INDEX;
17 }
18 }
19 }
20 }

Project

Comm

Struc

...

Search

Test

Prob

Veri

Cont

Notification

AI

DB

assignment3.2 > src > Solution1419:2 CRLF UTF-8 4 spaces

assignment3.2Version control

Current File

Solution8.javaSolution9.javaSolution10.javaSolution11.javaSolution12.javaSolution13.javaSolution14.javaSolution15.javaSolution16

Project

Comm

Struc

...

Sev

Term

Prob

Versi

Cont

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

```
public class Solution15 {  
    public static void main(String[] args) throws Exception {  
    }  
    interface Selectable { 1 usage 2 implementations  
        void onSelect(); no usages 1 implementation  
    }  
    interface Updatable extends Selectable { 1 usage 1 implementation  
        void refresh(); no usages 1 implementation  
    }  
    class Screen implements Updatable { no usages  
        public void refresh() { no usages  
            System.out.println("Screen refreshed");  
        }  
  
        @Override no usages  
        public void onSelect() {  
            System.out.println("Screen onSelected");  
        }  
    }  
}
```

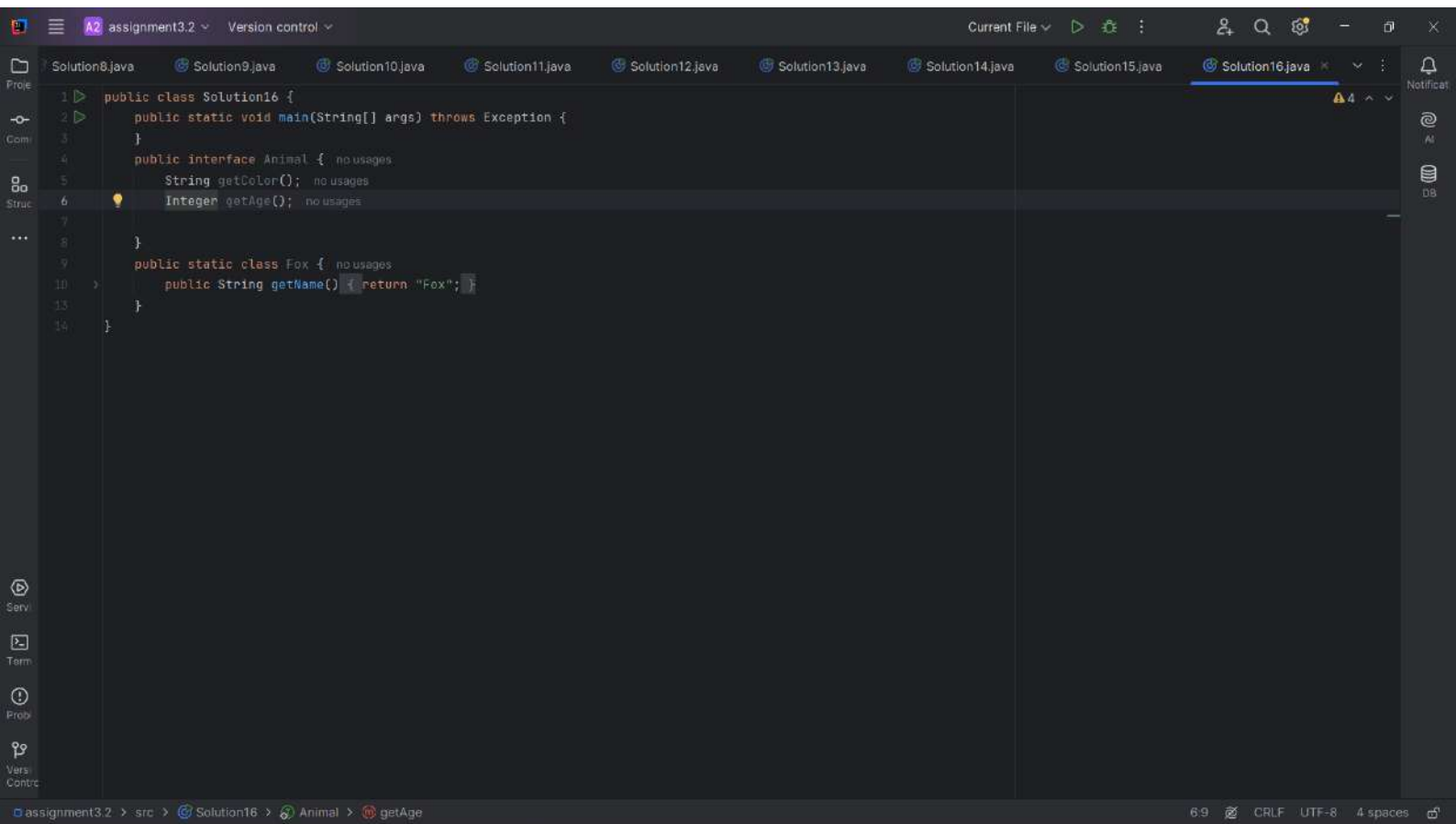
4

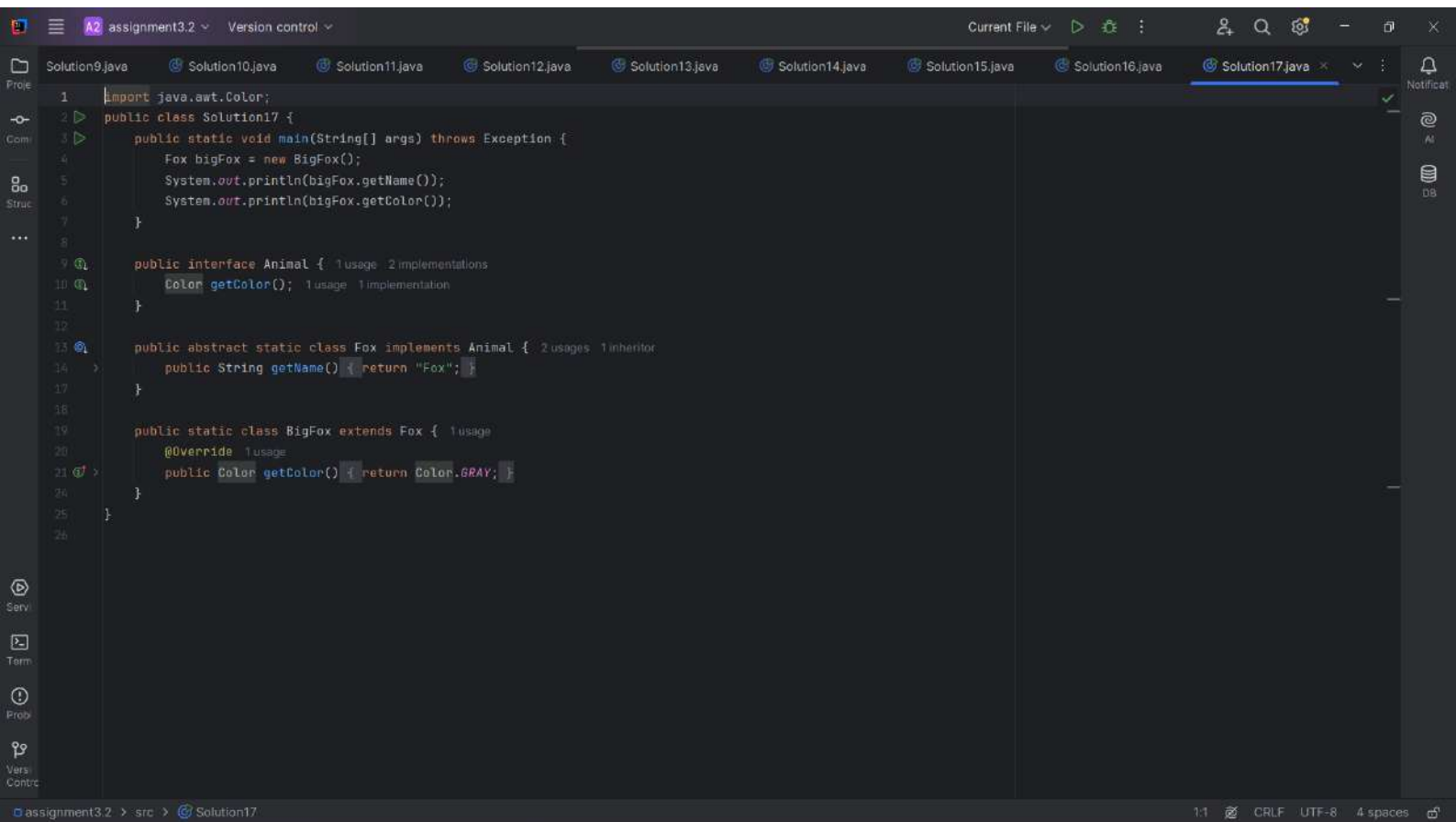
Notifications

AI

DB

assignment3.2 > src > Solution15 > Screen19:6 CRLF UTF-8 4 spaces





```
1
2 import java.util.ArrayList;
3 import java.util.List;
4 public class Solution18 {
5     public static void main(String[] args) {
6         List<Person> plot = new ArrayList<Person>();
7         plot.add(new Person( name: "Repka", namePadezh: "Repka"));
8         plot.add(new Person( name: "Dedka", namePadezh: "Dedku"));
9         plot.add(new Person( name: "Babka", namePadezh: "Babka"));
10        plot.add(new Person( name: "Granddaughter", namePadezh: "Granddaughter"));
11        RepkeStory.tell(plot);
12    }
13 }
```

assignment3.2 > src > Solution18 > main

5:41 CRLF UTF-8 4 spaces

assignment3.2Version control

Current File

Solution12.javaSolution13.javaSolution14.javaSolution15.javaSolution16.javaSolution17.javaSolution18.javaPerson.javaRepkaItem.java

1234567891011121314151617181920212223242526272829303132

```
public class Person implements RepkaItem {
    private String name;
    private String namePadezh;

    public Person(String name, String namePadezh) {
        this.name = name;
        this.namePadezh = namePadezh;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getNamePadezh() {
        return namePadezh;
    }

    public void setNamePadezh(String namePadezh) {
        this.namePadezh = namePadezh;
    }

    public void pull(Person person) {
        System.out.println(this.name + " for " + person.getNamePadezh());
    }
}
```

1:1CRLFUTF-84 spaces

