Week 7 Homework Q26

Telmen Enkhbold

San Fransico Bay University

CE480 - Java and Internet Application

Dr. Chang, Henry

10/12/2023

Author Note

The Question

1. Inheritance + Aggregation

```
------------------------
| A                    |
|   String name        |
|  --------- --------- |
|  | B     | | F     | |
|  | int b | | int f | |
|  --------- --------- |
------------------------
           |
           |
           |
-----------------------
| E                   |
|   int e             |
|  -----------------  |
|  | D             |  |
|  |   int d       |  |
|  -----------------  |
-----------------------
```

Note:

- Each class must implement 4 types of member functions
  - Helping function
  -
  - `private void trace(String s) {`
  - `   System.out.println(s);`
  - `}`

  - Access functions
    - get
    - set
    - predicate
      - class A, B, D, E, F
      -
      - `isLargeValue()`

==> Check whether the values of all its attributes are greater than 100. For example, class E's isLargeValue() returns true if all the values of b, f, d, e are greater than 100.

        - class A
        -
        - `isBruceLee()`
        - `    ==> Check whether the value of the attribute "name" is "Bruce Lee".`

    - Implementor function
        - class A, B, D, E, F
        -
        - `changetoZero()`

==> The values of all its numerical attributes are set to 0. For example, class E's changeroZero() has this result:

```
----------------------------
| A                        |
|   String name            |
|  ----------- ----------- |
|  | B       | | F       | |
|  | int b=0 | | int f=0 | |
|  ----------- ----------- |
----------------------------
           |
           |
           |
----------------------------
| E                        |
|   int e=0                |
|    -----------------     |
|    | D             |     |
|    |   int d=0     |     |
|    -----------------     |
----------------------------
```

class A

```
changeName(String new_name)
    ==> The value of the attribute "name" is changed
to a new name.
```

class A, B, D, E, F

- toString()
- clone()
- equals()

main function

Create this object eObj

```
----------------------------
| A                        |
|   String name="jack"     |
|  ----------- ----------- |
|  | B       | | F       | |
|  | int b=1 | | int f=2 | |
|  ----------- ----------- |
----------------------------
           |
           |
           |
----------------------------
| E                        |
|   int e=3                |
|    -----------------     |
|    | D             |     |
|    |   int d=4     |     |
|    -----------------     |
----------------------------
```

Display the values of all the attributes of eObj

Change the values of eObj to

```
----------------------------
| A                        |
|   String name="jack"     |
|  ----------- ----------- |
|  | B       | | F       | |
|  | int b=4 | | int f=5 | |
|  ----------- ----------- |
----------------------------
           |
           |
           |
----------------------------
| E                        |
|   int e=6                |
|    -----------------     |
|    | D             |     |
|    |   int d=7     |     |
|    -----------------     |
```

Display the values of all the attributes of eObj

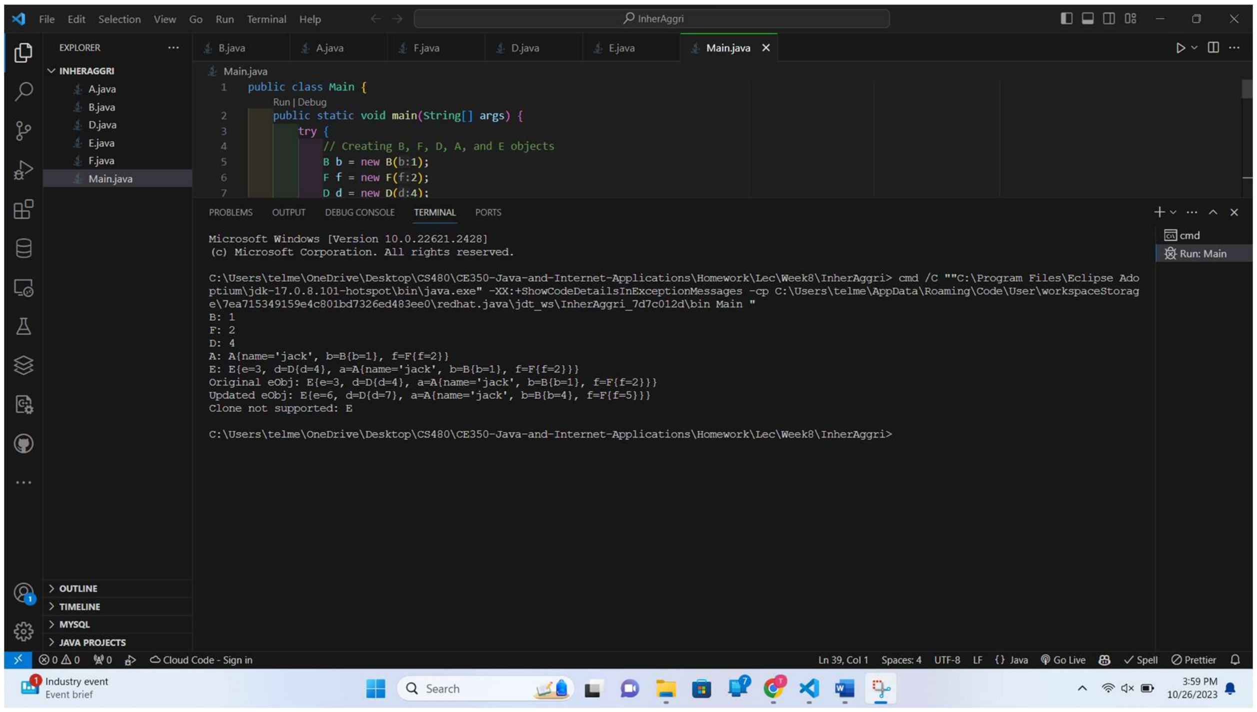
Clone the eObj to create eObj1

Display the values of all the attributes of eObj1

Check whether eObj is equal to eObj1

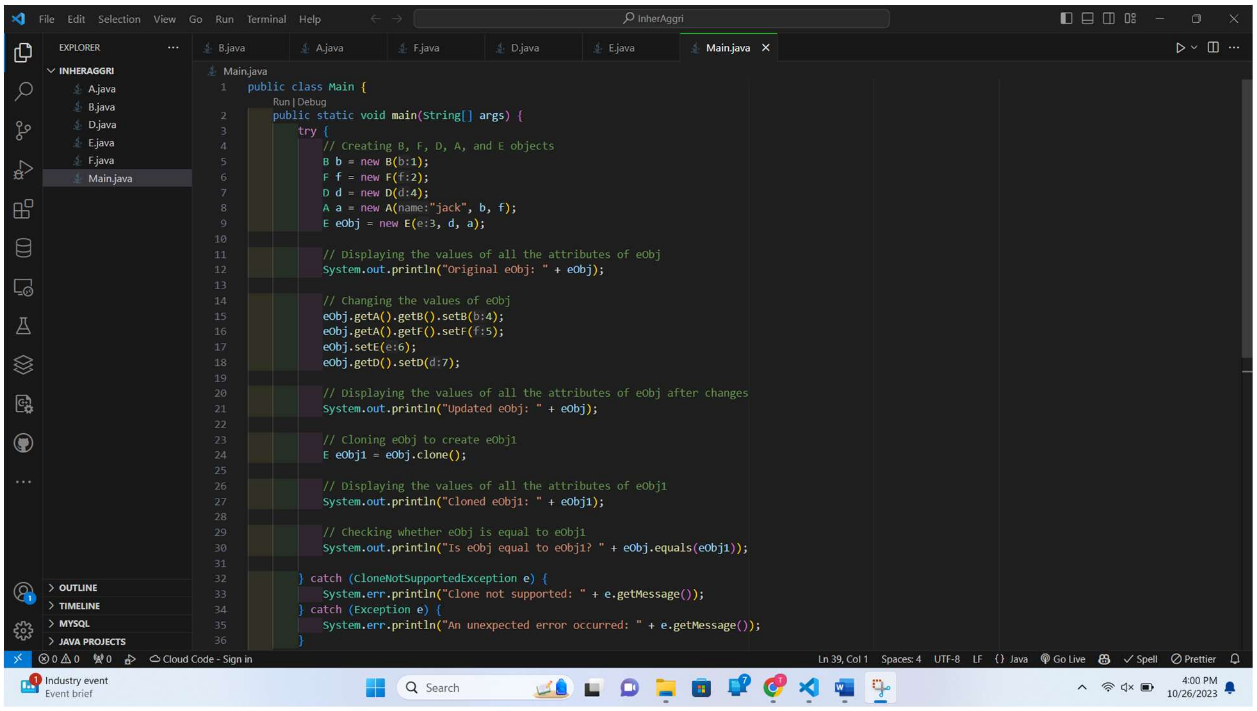
- References
  - Class containg two layers of simple object attribute
  - One layer of simple object attribute + toString() + clone() + equals()
  - get/set
  - 4 Types of Member Functions
  - Summary of toString(), clone(), equals()
  - Inheritance Class Structure

==============================Screenshot==================================

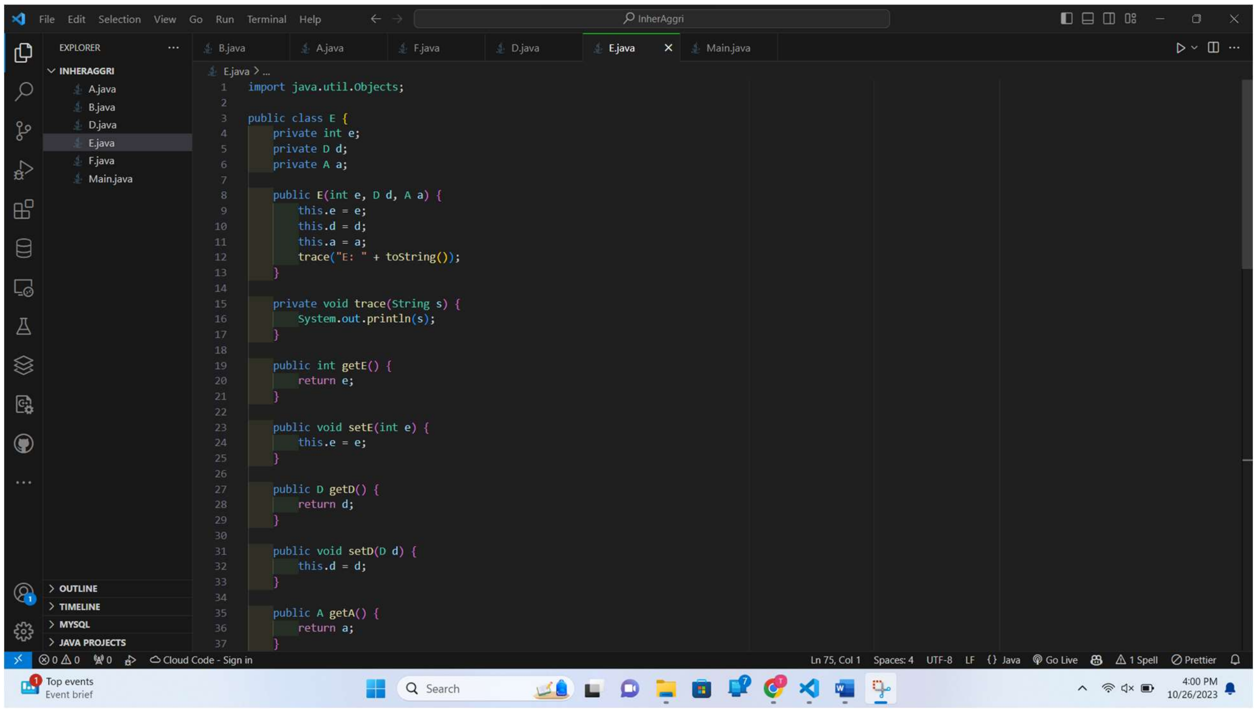


```
public class Main {
    public static void main(String[] args) {
        try {
            // Creating B, F, D, A, and E objects
            B b = new B(b:1);
            F f = new F(f:2);
            D d = new D(d:4);
```

```
Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

C:\Users\telme\OneDrive\Desktop\CS480\CE350-Java-and-Internet-Applications\Homework\Lec\Week8\InherAggri> cmd /C ""C:\Program Files\Eclipse Ado
ptium\jdk-17.0.8.101-hotspot\bin\java.exe" -XX:+ShowCodeDetailsInExceptionMessages -cp C:\Users\telme\AppData\Roaming\Code\User\workspaceStorag
e\7ea715349159e4c801bd7326ed483ee0\redhat.java\jdt_ws\InherAggri_7d7c012d\bin Main "
B: 1
F: 2
D: 4
A: A{name='jack', b=B{b=1}, f=F{f=2}}
E: E{e=3, d=D{d=4}, a=A{name='jack', b=B{b=1}, f=F{f=2}}}
Original eObj: E{e=3, d=D{d=4}, a=A{name='jack', b=B{b=1}, f=F{f=2}}}
Updated eObj: E{e=6, d=D{d=7}, a=A{name='jack', b=B{b=4}, f=F{f=5}}}
Clone not supported: E

C:\Users\telme\OneDrive\Desktop\CS480\CE350-Java-and-Internet-Applications\Homework\Lec\Week8\InherAggri>
```

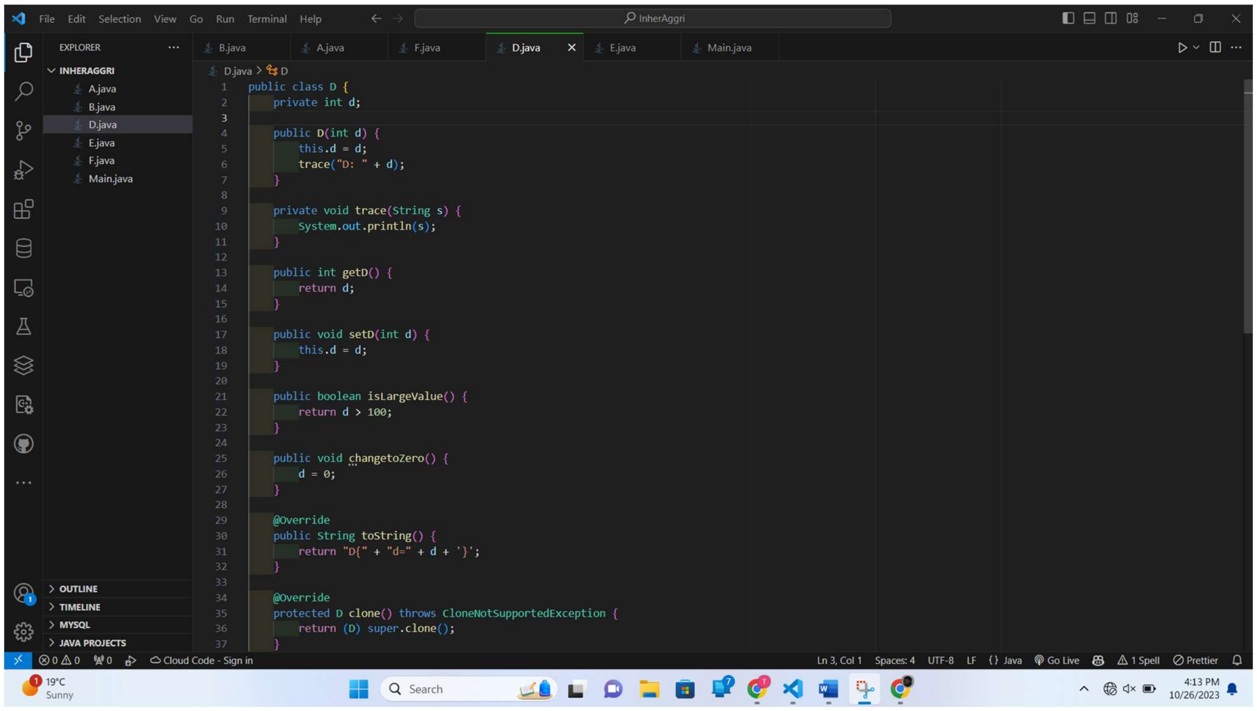


```
public class Main {
    public static void main(String[] args) {
        try {
            // Creating B, F, D, A, and E objects
            B b = new B(b:1);
            F f = new F(f:2);
            D d = new D(d:4);
            A a = new A(name:"jack", b, f);
            E eObj = new E(e:3, d, a);

            // Displaying the values of all the attributes of eObj
            System.out.println("Original eObj: " + eObj);

            // Changing the values of eObj
            eObj.getA().getB().setB(b:4);
            eObj.getA().getF().setF(f:5);
            eObj.setE(e:6);
            eObj.getD().setD(d:7);

            // Displaying the values of all the attributes of eObj after changes
            System.out.println("Updated eObj: " + eObj);

            // Cloning eObj to create eObj1
            E eObj1 = eObj.clone();

            // Displaying the values of all the attributes of eObj1
            System.out.println("Cloned eObj1: " + eObj1);

            // Checking whether eObj is equal to eObj1
            System.out.println("Is eObj equal to eObj1? " + eObj.equals(eObj1));

        } catch (CloneNotSupportedException e) {
            System.err.println("Clone not supported: " + e.getMessage());
        } catch (Exception e) {
            System.err.println("An unexpected error occurred: " + e.getMessage());
        }
```



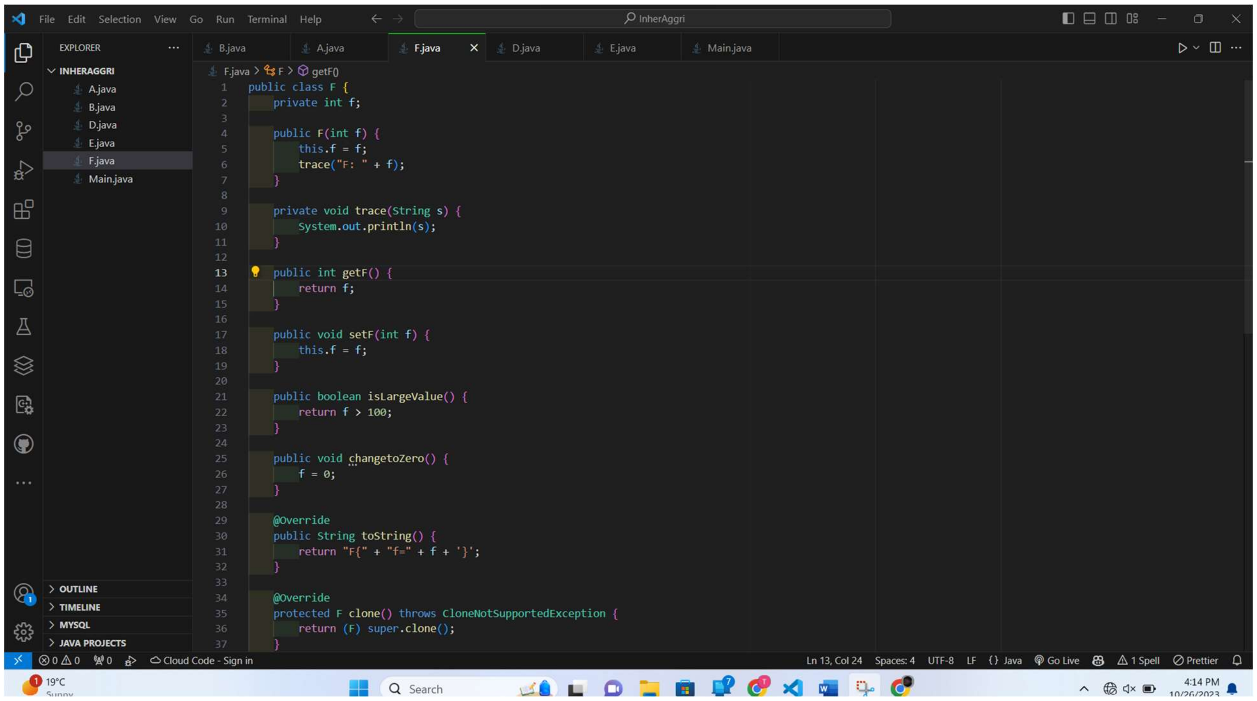
E.java > ...

```java
import java.util.Objects;

public class E {
    private int e;
    private D d;
    private A a;

    public E(int e, D d, A a) {
        this.e = e;
        this.d = d;
        this.a = a;
        trace("E: " + toString());
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public int getE() {
        return e;
    }

    public void setE(int e) {
        this.e = e;
    }

    public D getD() {
        return d;
    }

    public void setD(D d) {
        this.d = d;
    }

    public A getA() {
        return a;
    }
```

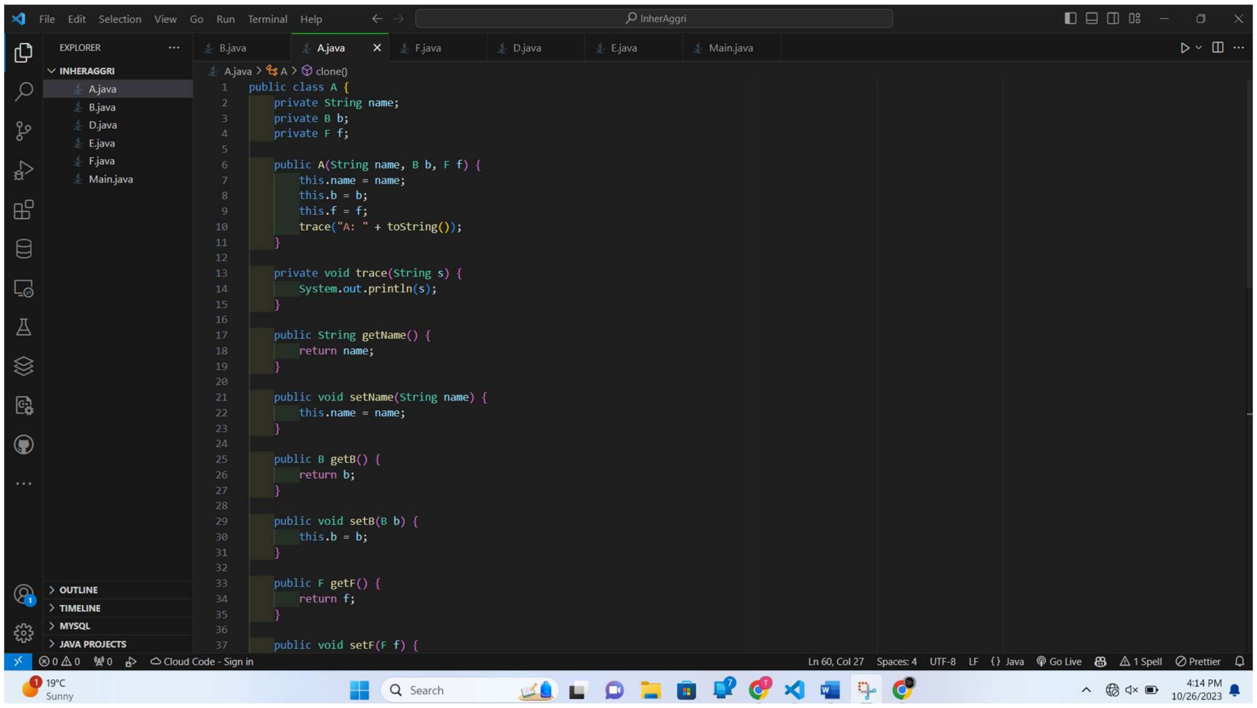


D.java > D

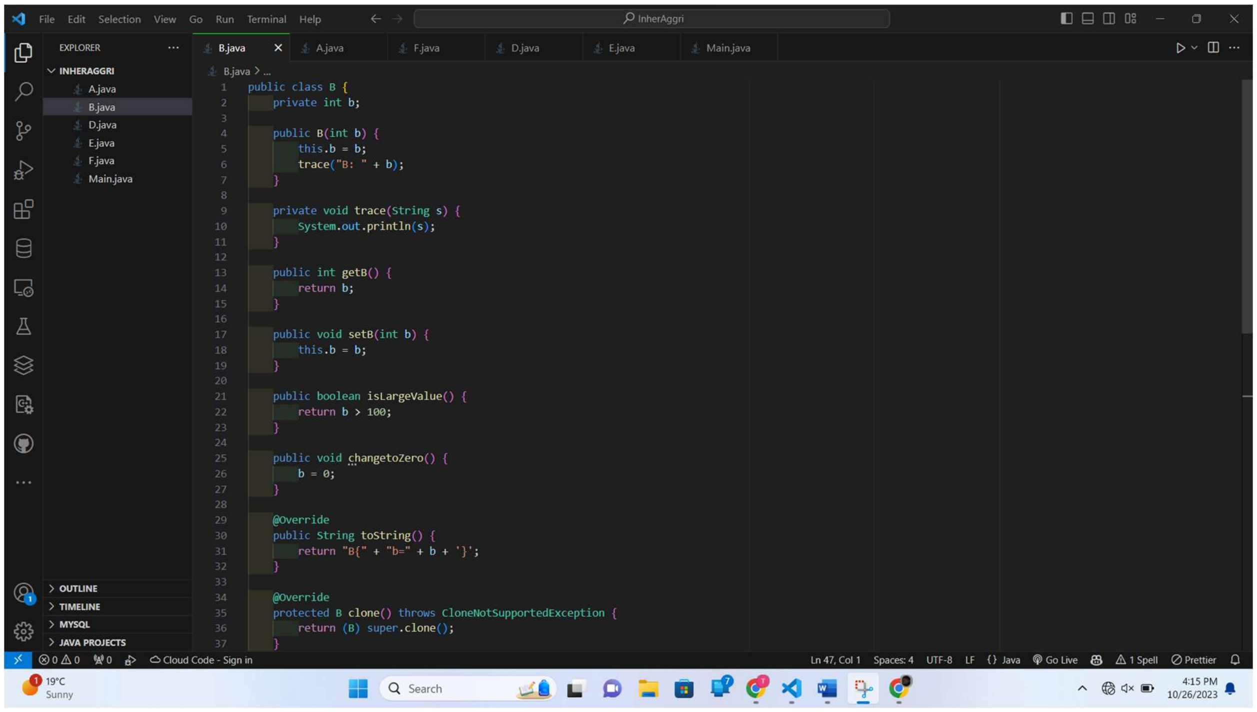
```java
public class D {
    private int d;

    public D(int d) {
        this.d = d;
        trace("D: " + d);
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public int getD() {
        return d;
    }

    public void setD(int d) {
        this.d = d;
    }

    public boolean isLargeValue() {
        return d > 100;
    }

    public void changetoZero() {
        d = 0;
    }

    @Override
    public String toString() {
        return "D{" + "d=" + d + '}';
    }

    @Override
    protected D clone() throws CloneNotSupportedException {
        return (D) super.clone();
    }
```



```java
public class F {
    private int f;

    public F(int f) {
        this.f = f;
        trace("F: " + f);
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public int getF() {
        return f;
    }

    public void setF(int f) {
        this.f = f;
    }

    public boolean isLargeValue() {
        return f > 100;
    }

    public void changetoZero() {
        f = 0;
    }

    @Override
    public String toString() {
        return "F{" + "f=" + f + '}';
    }

    @Override
    protected F clone() throws CloneNotSupportedException {
        return (F) super.clone();
    }
```



```java
public class A {
    private String name;
    private B b;
    private F f;

    public A(String name, B b, F f) {
        this.name = name;
        this.b = b;
        this.f = f;
        trace("A: " + toString());
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public String getName() {
        return name;
    }

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

    public B getB() {
        return b;
    }

    public void setB(B b) {
        this.b = b;
    }

    public F getF() {
        return f;
    }

    public void setF(F f) {
```



```java
public class B {
    private int b;

    public B(int b) {
        this.b = b;
        trace("B: " + b);
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public int getB() {
        return b;
    }

    public void setB(int b) {
        this.b = b;
    }

    public boolean isLargeValue() {
        return b > 100;
    }

    public void changetoZero() {
        b = 0;
    }

    @Override
    public String toString() {
        return "B{" + "b=" + b + '}';
    }

    @Override
    protected B clone() throws CloneNotSupportedException {
        return (B) super.clone();
    }
```

================================The Code================================

A.java

```java
public class A {
    private String name;
    private B b;
    private F f;

    public A(String name, B b, F f) {
        this.name = name;
        this.b = b;
        this.f = f;
        trace("A: " + toString());
    }

    private void trace(String s) {
        System.out.println(s);
    }

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

    public B getB() {
        return b;
    }

    public void setB(B b) {
        this.b = b;
    }

    public F getF() {
        return f;
    }

    public void setF(F f) {
        this.f = f;
    }

    public boolean isBruceLee() {
        return "Bruce Lee".equals(name);
    }

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

    public void changetoZero() {
        b.changetoZero();
        f.changetoZero();
    }

    @Override
    public String toString() {
        return "A{" + "name='" + name + '\'' + ", b=" + b + ", f=" + f +
'}';
    }

    @Override
    protected A clone() throws CloneNotSupportedException {
        A cloned = (A) super.clone();
        cloned.b = this.b.clone();
        return cloned;
    }
```

```
}
```

B.java

```
public class B {
    private int b;

    public B(int b) {
        this.b = b;
        trace("B: " + b);
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public int getB() {
        return b;
    }

    public void setB(int b) {
        this.b = b;
    }

    public boolean isLargeValue() {
        return b > 100;
    }

    public void changetoZero() {
        b = 0;
    }

    @Override
    public String toString() {
        return "B{" + "b=" + b + '}';
    }

    @Override
    protected B clone() throws CloneNotSupportedException {
        return (B) super.clone();
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        B b1 = (B) obj;
        return b == b1.b;
    }
}
```

D.java

```
public class D {
    private int d;

    public D(int d) {
        this.d = d;
        trace("D: " + d);
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public int getD() {
        return d;
    }

    public void setD(int d) {
        this.d = d;
    }

    public boolean isLargeValue() {
        return d > 100;
    }

    public void changetoZero() {
        d = 0;
    }

    @Override
    public String toString() {
        return "D{" + "d=" + d + '}';
    }

    @Override
    protected D clone() throws CloneNotSupportedException {
        return (D) super.clone();
```

```java
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        D d1 = (D) obj;
        return d == d1.d;
    }
}
```

E.java

```java
import java.util.Objects;

public class E {
    private int e;
    private D d;
    private A a;

    public E(int e, D d, A a) {
        this.e = e;
        this.d = d;
        this.a = a;
        trace("E: " + toString());
    }

    private void trace(String s) {
        System.out.println(s);
    }

    public int getE() {
        return e;
    }

    public void setE(int e) {
        this.e = e;
    }

    public D getD() {
        return d;
    }
    public void setD(D d) {
        this.d = d;
    }

    public A getA() {
        return a;
    }

    public void setA(A a) {
        this.a = a;
    }

    public boolean isLargeValue() {
        return e > 100 && d.isLargeValue() && a.getB().isLargeValue() &&
a.getF().isLargeValue();
    }

    public void changetoZero() {
        e = 0;
        d.changetoZero();
        a.changetoZero();
    }

    @Override
    public String toString() {
        return "E{" + "e=" + e + ", d=" + d + ", a=" + a + '}';
    }

    @Override
    protected E clone() throws CloneNotSupportedException {
        E cloned = (E) super.clone();
        cloned.d = this.d.clone();
        cloned.a = this.a.clone();
        return cloned;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        E e1 = (E) obj;
        return e == e1.e && Objects.equals(d, e1.d) && Objects.equals(a,
e1.a);
    }
}
```

Main.java

```java
public class Main {
    public static void main(String[] args) {
        try {
            // Creating B, F, D, A, and E objects
            B b = new B(1);
            F f = new F(2);
            D d = new D(4);
            A a = new A("jack", b, f);
            E eObj = new E(3, d, a);

            // Displaying the values of all the attributes of eObj
            System.out.println("Original eObj: " + eObj);

            // Changing the values of eObj
            eObj.getA().getB().setB(4);
            eObj.getA().getF().setF(5);
            eObj.setE(6);
            eObj.getD().setD(7);

            // Displaying the values of all the attributes of eObj after
changes
            System.out.println("Updated eObj: " + eObj);

            // Cloning eObj to create eObj1
            E eObj1 = eObj.clone();

            // Displaying the values of all the attributes of eObj1
            System.out.println("Cloned eObj1: " + eObj1);

            // Checking whether eObj is equal to eObj1
            System.out.println("Is eObj equal to eObj1? " +
eObj.equals(eObj1));

        } catch (CloneNotSupportedException e) {
            System.err.println("Clone not supported: " + e.getMessage());
        } catch (Exception e) {
            System.err.println("An unexpected error occurred: " +
e.getMessage());
        }
    }
}
```

Refrence

https://hc.labnet.sfbu.edu/~henry/sfbu/course/introjava/inheritance/slide/exercises4.htmlLinks to an external site.

Q26 ==> Aggregation and inheritance

Github-https://github.com/Georgycas/CE350-Java-and-Internet-Applications/tree/main/Homework/Lec/Week8/InherAggri