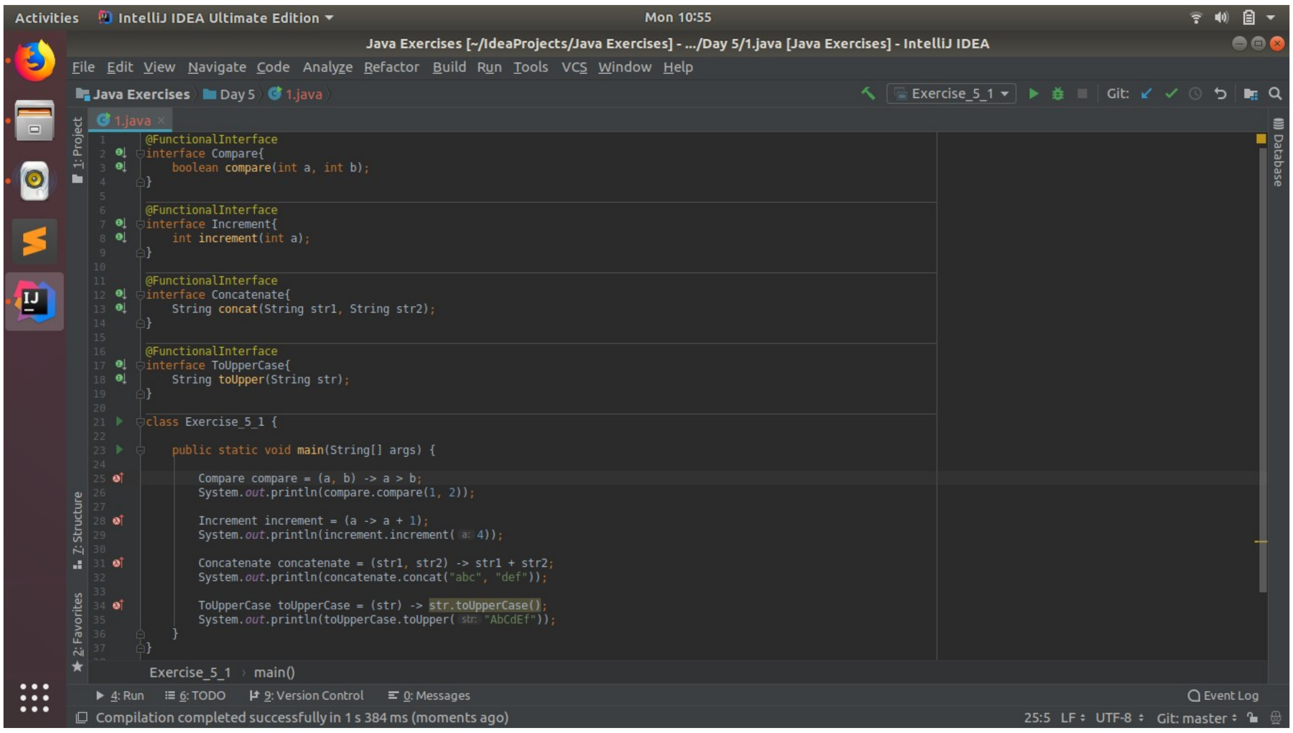
Q1. Write the following a functional interface and implement it using lambda:

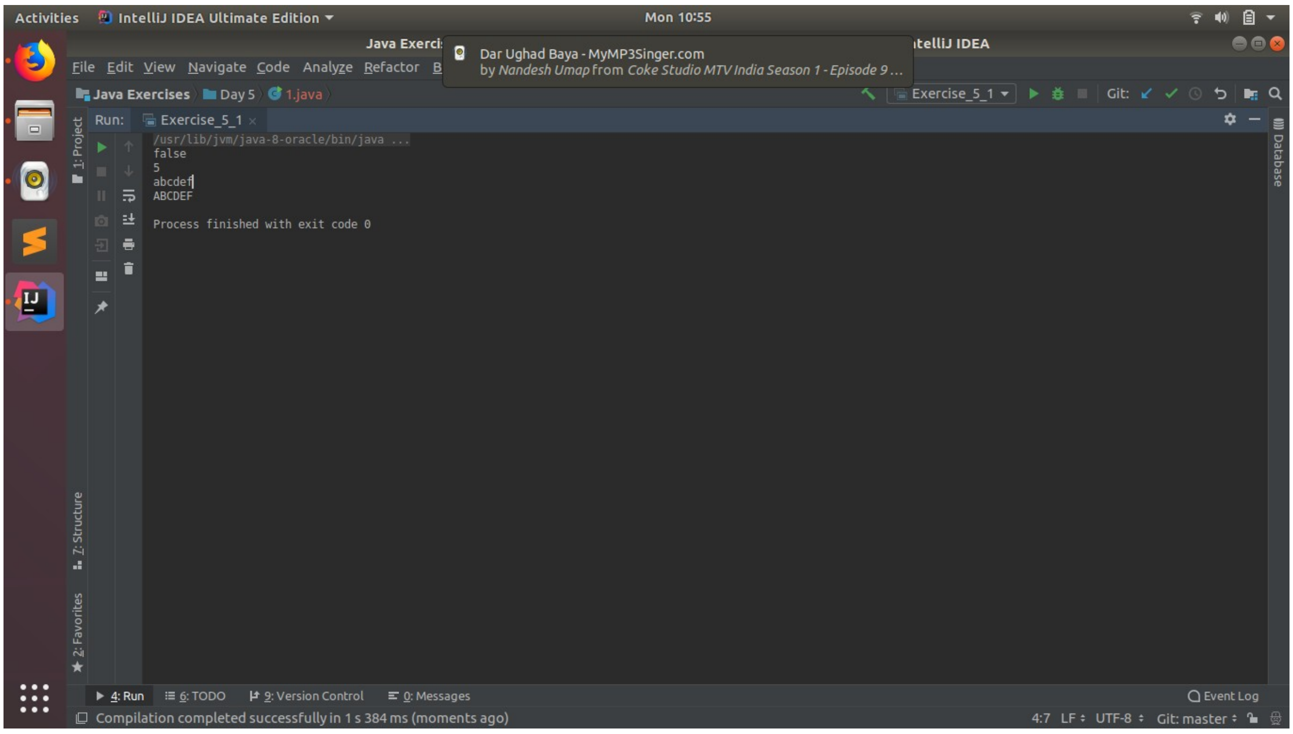
(1) First number is greater than second number or not   Parameter (int ,int) Return boolean
(2) Increment the number by 1 and return incremented value   Parameter (int) Return int
(3) Concatination of 2 string   Parameter (String , String) Return (String)
(4) Convert a string to uppercase and return .   Parameter (String) Return (String)

A1.



```java
@FunctionalInterface
interface Compare{
    boolean compare(int a, int b);
}

@FunctionalInterface
interface Increment{
    int increment(int a);
}

@FunctionalInterface
interface Concatenate{
    String concat(String str1, String str2);
}

@FunctionalInterface
interface ToUpperCase{
    String toUpper(String str);
}

class Exercise_5_1 {

    public static void main(String[] args) {

        Compare compare = (a, b) -> a > b;
        System.out.println(compare.compare(1, 2));

        Increment increment = (a -> a + 1);
        System.out.println(increment.increment(4));

        Concatenate concatenate = (str1, str2) -> str1 + str2;
        System.out.println(concatenate.concat("abc", "def"));

        ToUpperCase toUpperCase = (str) -> str.toUpperCase();
        System.out.println(toUpperCase.toUpper("AbCdEf"));
    }
}
```

Compilation completed successfully in 1 s 384 ms (moments ago)

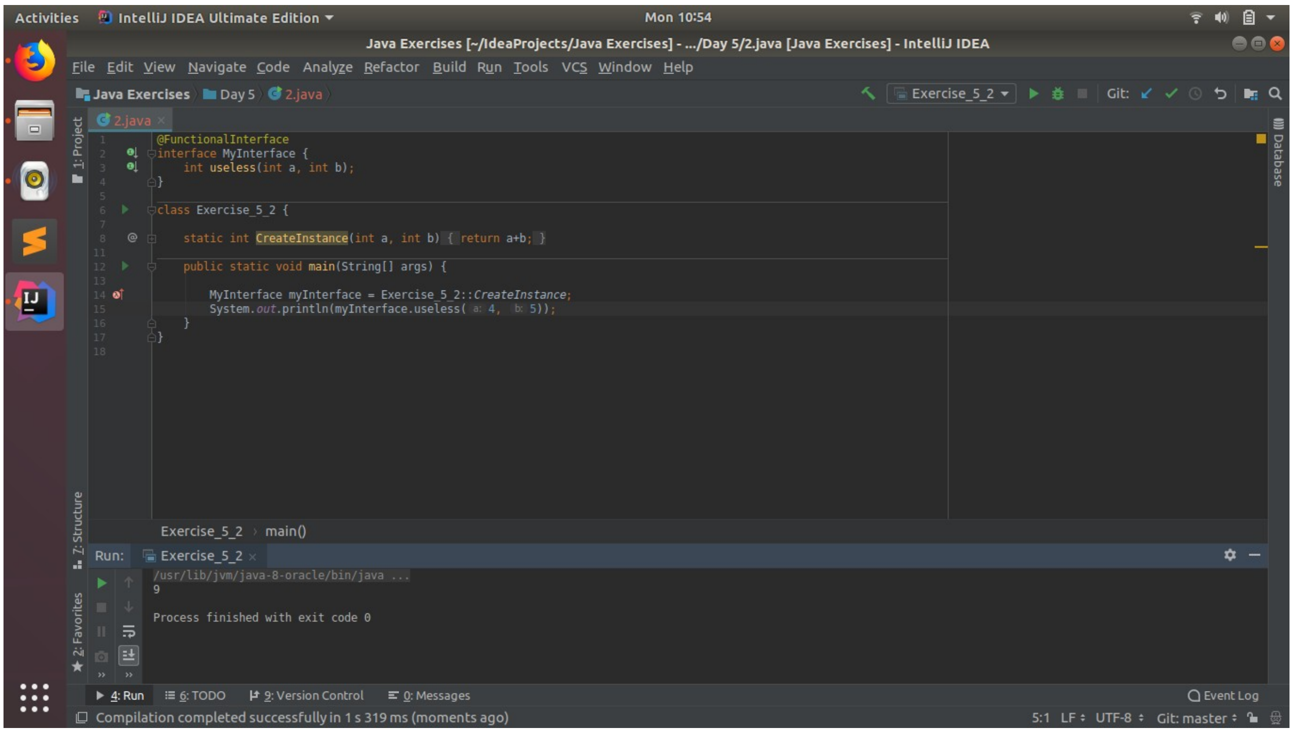


```
/usr/lib/jvm/java-8-oracle/bin/java ...
false
5
abcdef
ABCDEF

Process finished with exit code 0
```

Compilation completed successfully in 1 s 384 ms (moments ago)

Q2. Create a functional interface whose method takes 2 integers and return one integer.

A2.



Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/2.java [Java Exercises] - IntelliJ IDEA

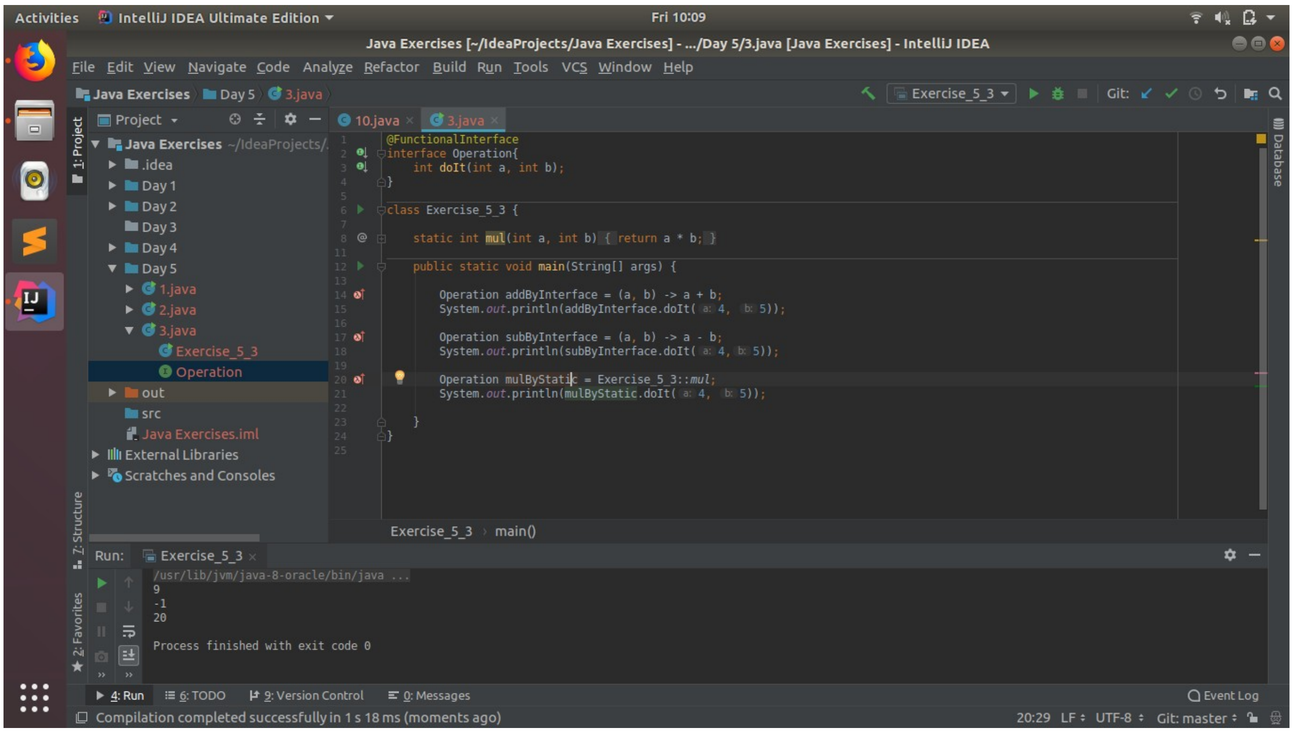
```java
@FunctionalInterface
interface MyInterface {
    int useless(int a, int b);
}

class Exercise_5_2 {

    static int CreateInstance(int a, int b) { return a+b; }

    public static void main(String[] args) {

        MyInterface myInterface = Exercise_5_2::CreateInstance;
        System.out.println(myInterface.useless( a: 4,  b: 5));
    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
9

Process finished with exit code 0
```

Compilation completed successfully in 1 s 319 ms (moments ago)

Q3. Using (instance) Method reference create and apply add and subtract method and using (Static) Method reference create and apply multiplication method for the functional interface created.

A3.



Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/3.java [Java Exercises] - IntelliJ IDEA

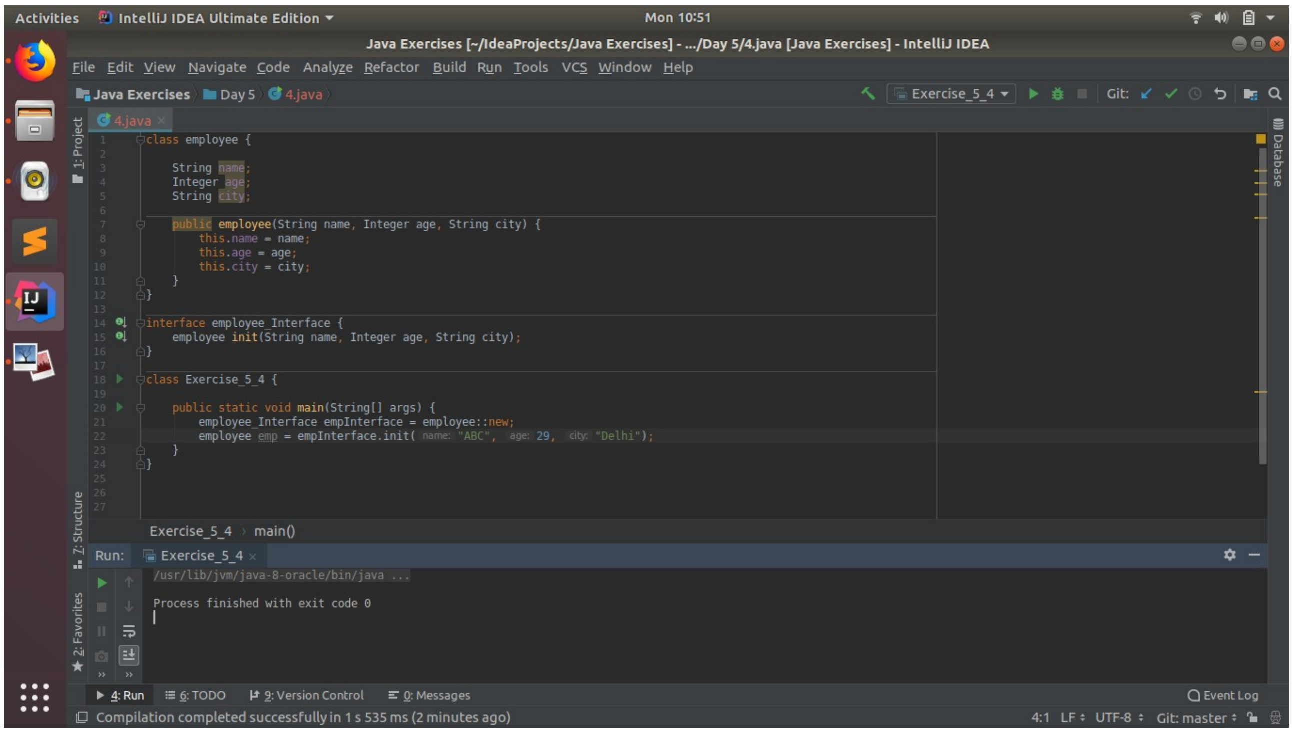
```java
@FunctionalInterface
interface Operation{
    int doIt(int a, int b);
}

class Exercise_5_3 {

    static int mul(int a, int b) { return a * b; }

    public static void main(String[] args) {

        Operation addByInterface = (a, b) -> a + b;
        System.out.println(addByInterface.doIt( a: 4,  b: 5));

        Operation subByInterface = (a, b) -> a - b;
        System.out.println(subByInterface.doIt( a: 4, b: 5));

        Operation mulByStatic = Exercise_5_3::mul;
        System.out.println(mulByStatic.doIt( a: 4,  b: 5));

    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
9
-1
20

Process finished with exit code 0
```

Compilation completed successfully in 1 s 18 ms (moments ago)

Q4. Create an Employee Class with instance variables (String) name, (Integer)age, (String)city and get the instance of the Class using constructor reference.

A4.



```
class employee {

    String name;
    Integer age;
    String city;

    public employee(String name, Integer age, String city) {
        this.name = name;
        this.age = age;
        this.city = city;
    }
}

interface employee_Interface {
    employee init(String name, Integer age, String city);
}

class Exercise_5_4 {

    public static void main(String[] args) {
        employee_Interface empInterface = employee::new;
        employee emp = empInterface.init( name: "ABC",  age: 29,  city: "Delhi");
    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...

Process finished with exit code 0
```

Compilation completed successfully in 1 s 535 ms (2 minutes ago)

Q5. Implement following functional interfaces from java.util.function using lambdas:

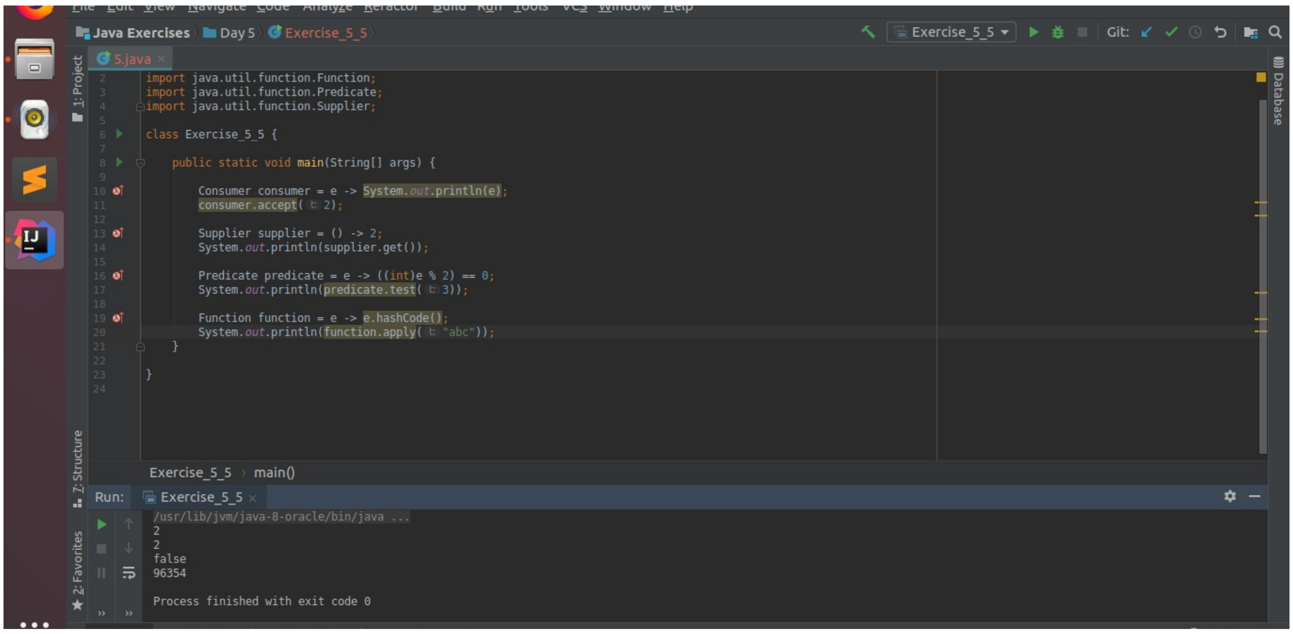
(1) Consumer

(2) Supplier

(3) Predicate

(4) Function

A5.

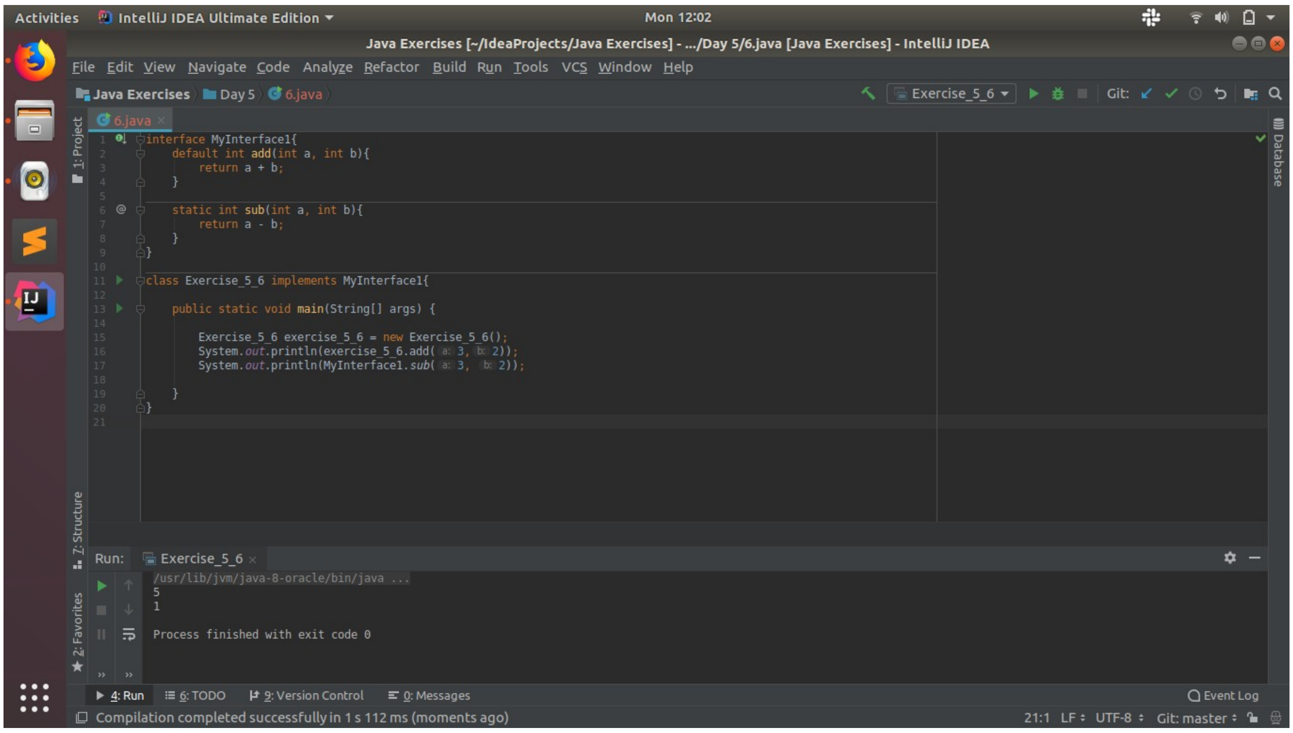


```
import java.util.function.Function;
import java.util.function.Predicate;
import java.util.function.Supplier;

class Exercise_5_5 {

    public static void main(String[] args) {

        Consumer consumer = e -> System.out.println(e);
        consumer.accept( t: 2);

        Supplier supplier = () -> 2;
        System.out.println(supplier.get());

        Predicate predicate = e -> ((int)e % 2) == 0;
        System.out.println(predicate.test( t: 3));

        Function function = e -> e.hashCode();
        System.out.println(function.apply( t: "abc"));
    }

}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
2
2
false
96354

Process finished with exit code 0
```

Q6. Create and access default and static method of an interface.

A6.


Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/6.java [Java Exercises] - IntelliJ IDEA

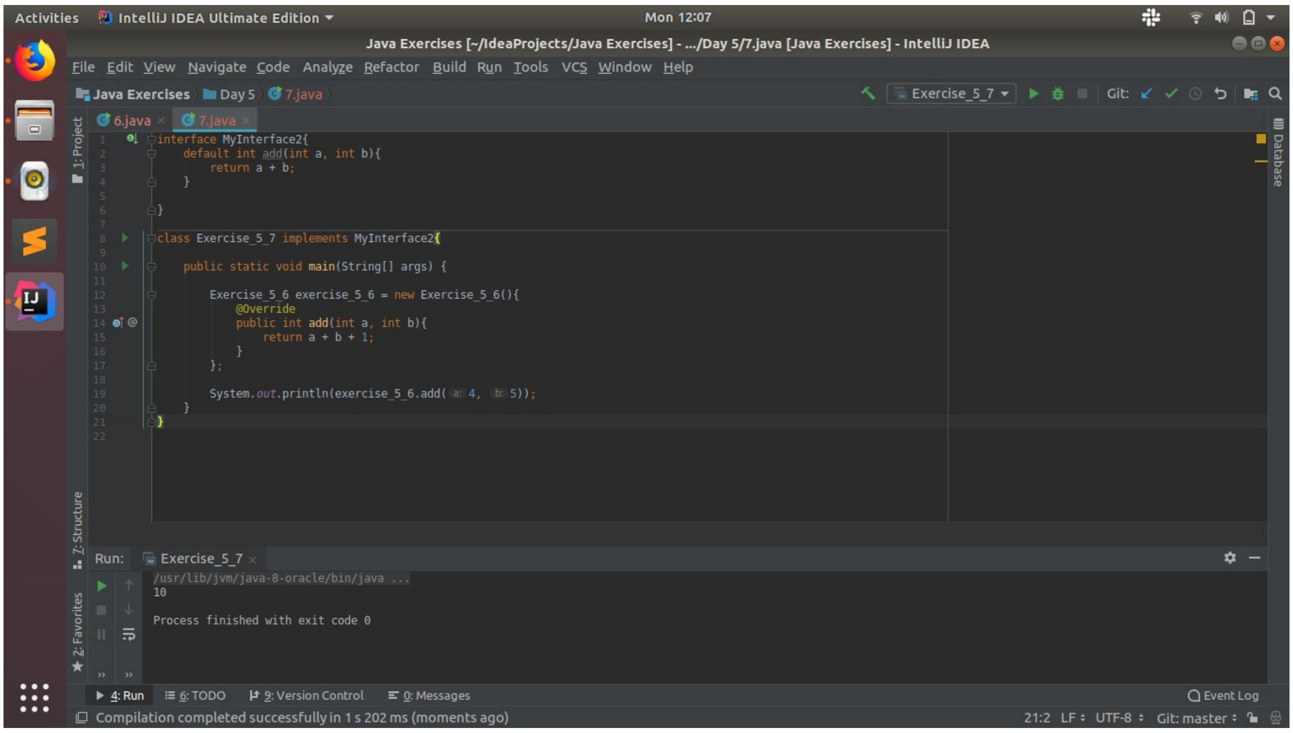
```java
interface MyInterface1{
    default int add(int a, int b){
        return a + b;
    }

    static int sub(int a, int b){
        return a - b;
    }
}

class Exercise_5_6 implements MyInterface1{

    public static void main(String[] args) {

        Exercise_5_6 exercise_5_6 = new Exercise_5_6();
        System.out.println(exercise_5_6.add( a: 3, b: 2));
        System.out.println(MyInterface1.sub( a: 3,  b: 2));

    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
5
1

Process finished with exit code 0
```

Compilation completed successfully in 1 s 112 ms (moments ago)

Q7. Override the default method of the interface.

A7.


Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/7.java [Java Exercises] - IntelliJ IDEA

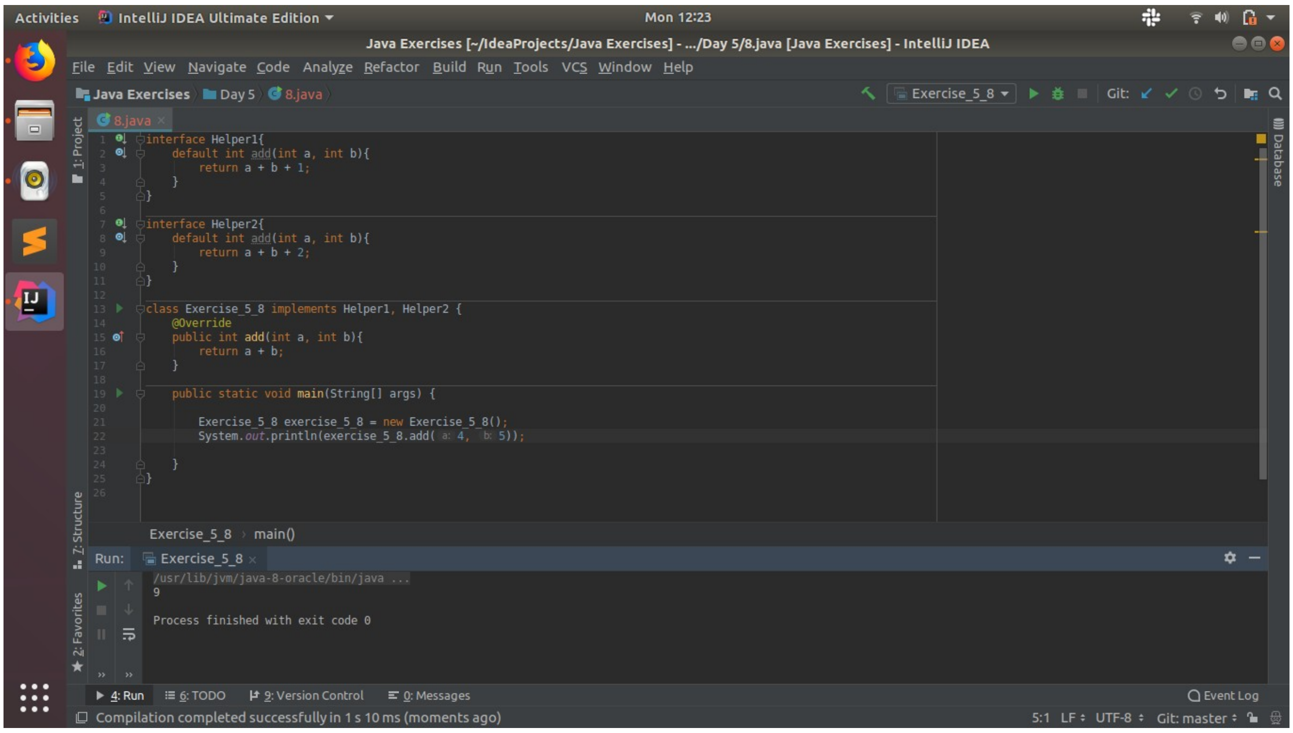
```java
interface MyInterface2{
    default int add(int a, int b){
        return a + b;
    }

}

class Exercise_5_7 implements MyInterface2{

    public static void main(String[] args) {

        Exercise_5_6 exercise_5_6 = new Exercise_5_6(){
            @Override
            public int add(int a, int b){
                return a + b + 1;
            }
        };

        System.out.println(exercise_5_6.add( a: 4,  b: 5));
    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
10

Process finished with exit code 0
```

Compilation completed successfully in 1 s 202 ms (moments ago)

Q8. Implement multiple inheritance with default method inside interface.

A8.



Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/8.java [Java Exercises] - IntelliJ IDEA

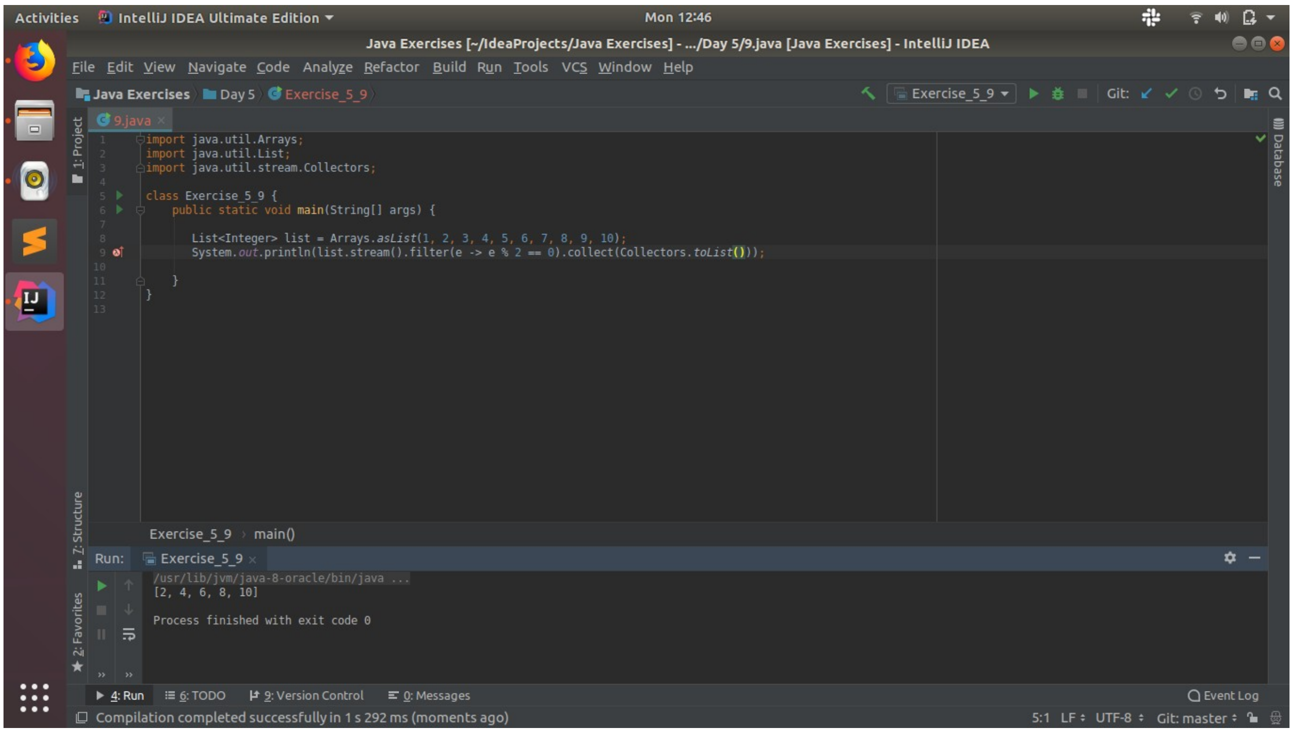
```java
interface Helper1{
    default int add(int a, int b){
        return a + b + 1;
    }
}

interface Helper2{
    default int add(int a, int b){
        return a + b + 2;
    }
}

class Exercise_5_8 implements Helper1, Helper2 {
    @Override
    public int add(int a, int b){
        return a + b;
    }

    public static void main(String[] args) {

        Exercise_5_8 exercise_5_8 = new Exercise_5_8();
        System.out.println(exercise_5_8.add( a: 4,  b: 5));

    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
9

Process finished with exit code 0
```

Compilation completed successfully in 1 s 10 ms (moments ago)

Q9. Collect all the even numbers from an integer list.

A9.



Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/9.java [Java Exercises] - IntelliJ IDEA

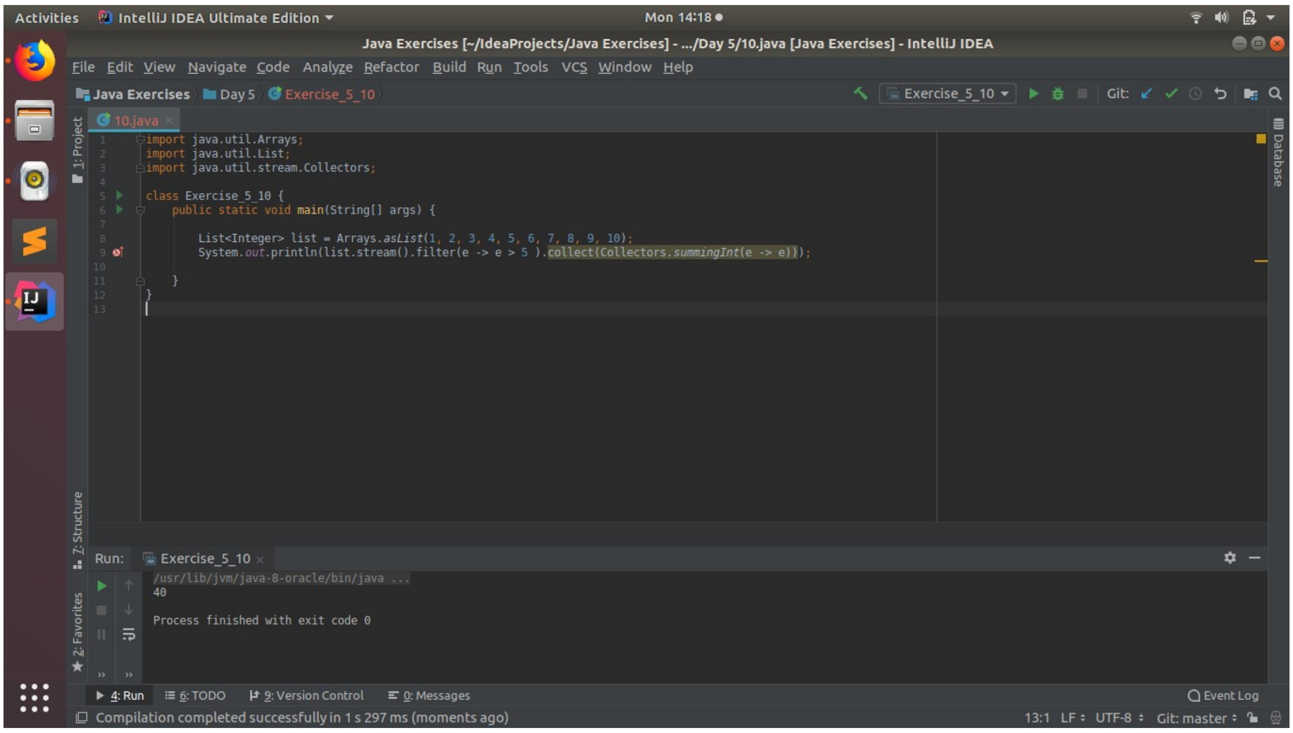
```java
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

class Exercise_5_9 {
    public static void main(String[] args) {

        List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
        System.out.println(list.stream().filter(e -> e % 2 == 0).collect(Collectors.toList()));

    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
[2, 4, 6, 8, 10]

Process finished with exit code 0
```

Compilation completed successfully in 1 s 292 ms (moments ago)

Q10. Sum all the numbers greater than 5 in the integer list.

A10.


Activities
IntelliJ IDEA Ultimate Edition
Mon 14:18
Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/10.java [Java Exercises] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Java Exercises › Day 5 › Exercise_5_10
Exercise_5_10
Git:
10.java
Run: Exercise_5_10
4: Run 6: TODO 9: Version Control 0: Messages Event Log
13:1 LF UTF-8 Git: master

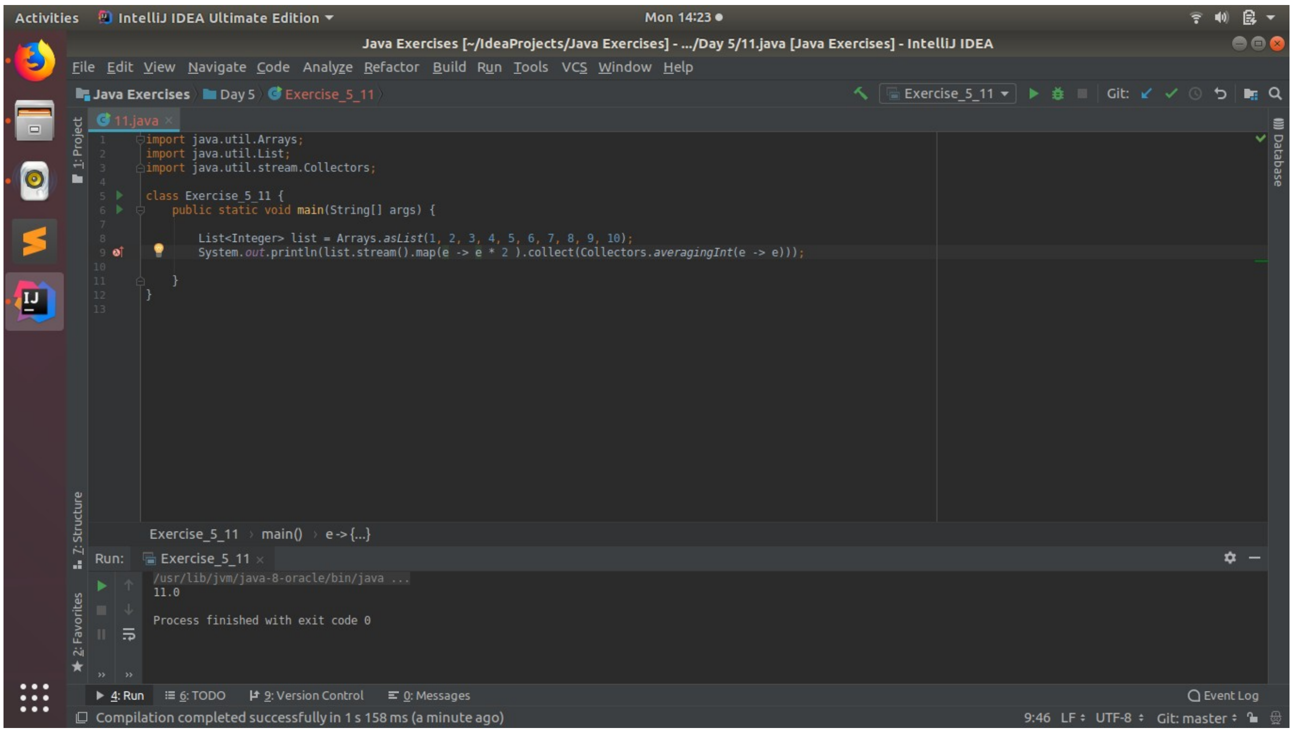
```java
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

class Exercise_5_10 {
    public static void main(String[] args) {

        List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
        System.out.println(list.stream().filter(e -> e > 5 ).collect(Collectors.summingInt(e -> e)));

    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
40

Process finished with exit code 0
```

Compilation completed successfully in 1 s 297 ms (moments ago)

Q11. Find average of the number inside integer list after doubling it.

A11.


Activities
IntelliJ IDEA Ultimate Edition
Mon 14:23
Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/11.java [Java Exercises] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Java Exercises › Day 5 › Exercise_5_11
Exercise_5_11
Git:
11.java
Exercise_5_11 › main() › e -> {...}
Run: Exercise_5_11
4: Run 6: TODO 9: Version Control 0: Messages Event Log
9:46 LF UTF-8 Git: master

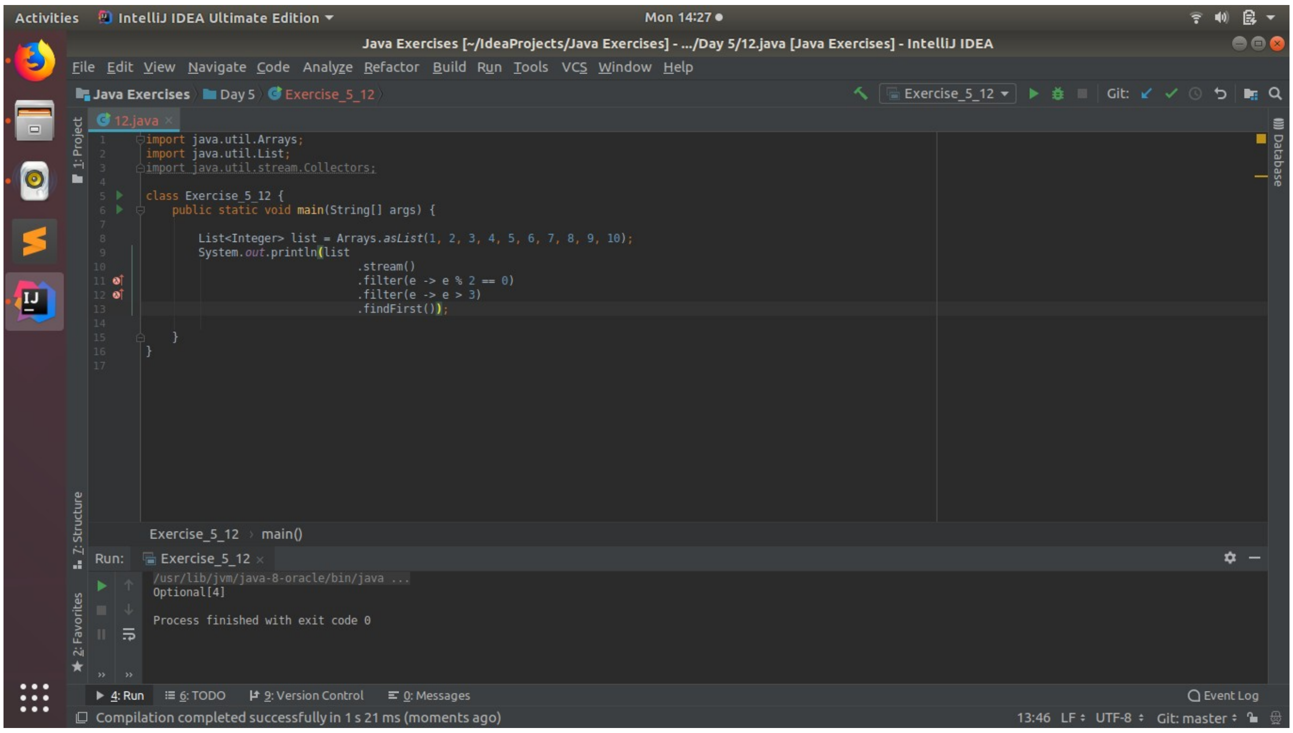
```java
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

class Exercise_5_11 {
    public static void main(String[] args) {

        List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
        System.out.println(list.stream().map(e -> e * 2 ).collect(Collectors.averagingInt(e -> e)));

    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
11.0

Process finished with exit code 0
```

Compilation completed successfully in 1 s 158 ms (a minute ago)

Q12. Find the first even number in the integer list which is greater than 3.

A12.



```java
import java.util.Arrays;
import java.util.List;
import java.util.stream.Collectors;

class Exercise_5_12 {
    public static void main(String[] args) {

        List<Integer> list = Arrays.asList(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);
        System.out.println(list
                                .stream()
                                .filter(e -> e % 2 == 0)
                                .filter(e -> e > 3)
                                .findFirst());

    }
}
```

```
/usr/lib/jvm/java-8-oracle/bin/java ...
Optional[4]

Process finished with exit code 0
```

Compilation completed successfully in 1 s 21 ms (moments ago)

Activities
IntelliJ IDEA Ultimate Edition
Mon 14:27
Java Exercises [~/IdeaProjects/Java Exercises] - .../Day 5/12.java [Java Exercises] - IntelliJ IDEA
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
Java Exercises › Day 5 › Exercise_5_12
Exercise_5_12
Git:
12.java
1: Project
Database
Exercise_5_12 › main()
Run: Exercise_5_12
7: Structure
2: Favorites
4: Run
6: TODO
9: Version Control
0: Messages
Event Log
13:46 LF UTF-8 Git: master