

Task 5: Functional Interfaces

Create a method that accepts functions as parameters using Predicate, Function, Consumer, and Supplier interfaces to operate on a Person object.

```
package Day19;

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

public class Task5 {
    public static void operateOnPerson(
        Person person,
        Predicate<Person> predicate,
        Function<Person, String> function,
        Consumer<Person> consumer,
        Supplier<Person> supplier) {

        if (predicate.test(person)) {
            System.out.println("Person matches the predicate");
        }

        String result = function.apply(person);
        System.out.println("Result of function: " + result);

        consumer.accept(person);

        Person newPerson = supplier.get();
        System.out.println("New person created: " + newPerson.getName());
    }

    public static void main(String[] args) {
        Person person = new Person("John", 30);

        Predicate<Person> predicate = p -> p.getAge() > 18;
        Function<Person, String> function = p -> "Hello, " + p.getName();
        Consumer<Person> consumer = p -> System.out.println("Consuming person: " + p.getName());
        Supplier<Person> supplier = () -> new Person("Jane", 25);

        operateOnPerson(person, predicate, function, consumer, supplier);
    }
}
```

```

13         Predicate<Person> predicate,
14         Function<Person, String> function,
15         Consumer<Person> consumer,
16         Supplier<Person> supplier) {
17
18         if (predicate.test(person)) {
19             System.out.println("Person matches the predicate");
20         }
21
22         String result = function.apply(person);
23         System.out.println("Result of function: " + result);
24
25         consumer.accept(person);
26
27         Person newPerson = supplier.get();
28         System.out.println("New person created: " + newPerson.getName());
29     }
30
31     public static void main(String[] args) {
32         Person person = new Person("John", 30);
33
34         Predicate<Person> predicate = p -> p.getAge() > 18;
35         Function<Person, String> function = p -> "Hello, " + p.getName();
36         Consumer<Person> consumer = p -> System.out.println("Consuming person: " + p.getName());
37         Supplier<Person> supplier = () -> new Person("Jane", 25);
38
39         operateOnPerson(person, predicate, function, consumer, supplier);
40     }
41 }

```

Markers Properties Terminal Console Coverage

```

<terminated> Task5 [Java Application] C:\Users\Nikita\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.2.v20210721-1149\
Person matches the predicate
Result of function: Hello, John
Consuming person: John
New person created: Jane

```