NegateConditionalsMutator

Below is an example of Java code and its mutated version using the NegateConditionals mutator, which negates a conditional:

Orginal:

```
public int numbers(int x, int y) {
    if (x == y){
        return x;
        }
        return y;
}
```

Mutated:

```
public int numbers(int x, int y) {
    if (x != y){
        return x;
        }
        return y;
}
```

—

Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Only mutate one conditional. Do not

include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

VoidMethodCallMutator

Below is an example of Java code and its mutated version using the VoidMethodCall mutator, The void method call mutator removes method calls to void methods (method that do not return a value). For example:

Original:

```
public void someVoidMethod(int i) {
  // does something
}

public int foo() {
  int i = 5;
  someVoidMethod(i);
  return i;
}
```

Mutated:

```
public void someVoidMethod(int i) {
  // does something
```

}

public int foo() {
  int i = 5;
  return i;
}

Notice how the removal of the void method call was only applied once.
Based on the above example, mutate the following java method, remove void one void method call such as handleException(e) for example, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

MathMutator
Below is an example of Java code and its mutated version using the Math mutator, which replaces binary arithmetic operations for either integer or floating-point arithmetic with another operation. The replacements will be selected according to the table below:
+ changes to -
- changes to +

* changes to /
/ changes to *
% changes to *
& changes to |
| changes to &
^ changes to &
<< changes to >>
>> changes to <<
>>> changes to <<
For example:

Original:

```
public void foo(int a, int b) {
  int c = a + b;
  int d = a - b;
}
```

Mutated:

```
public void foo(int a, int b) {
  int c = a - b;
  int d = a - b;
}
```

Notice how the mutator was only applied once.

Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

returns.NullReturnValsMutator
Below is an example of Java code and its mutated version using the NullReturnsValue mutator, which replaces a methods return value that is not already null with null. Here is an example:

Original:

```
public String getGreeting() {
    if (getNumbs() > 2) {
        return "Hello, World!";
        }
    else {
        return "Hello, Universe!";
        }
}
```

Mutated:

```
public String getGreeting() {
    if (getNumbs() > 2) {
        return "Hello, World!";
        }
    else {
        return null;
        }
}
```

Notice how the mutator was only applied once.
Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

returns.PrimitiveReturnsMutator
Below is an example of Java code and its mutated version using the PrimitiveReturnsMutator mutator, which replaces an int, short, long, char, float, or double that is not already 0 with a 0. Here is an example:

Original:
```
public int getGreeting() {
```

```
    if (getNumbs() > 2) {
        return 2;
        }
    else {
        return 1;
        }
}
```

Mutated:

```
public int getGreeting() {
    if (getNumbs() > 2) {
        return 2;
        }
    else {
        return 0;
        }
}
```

Notice how the mutator was only applied once.
Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

returns.BooleanTrueReturnValsMutator

Below is an example of Java code and its mutated version using the BooleanTrueReturnValsMutator mutator, which replaces a boolean return value that is not already true with true. Here is an example:

Original:

```
public boolean nums(int number) {
    if (number > 2) {
        return true;
    }
    return number % 2 == 0;
}
```

Mutated:

```
public boolean nums(int number) {
    if (number > 2) {
        return true;
    }
    return true;
}
```

Notice how the mutator was only applied once.

Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

ConditionalsBoundaryMutator

Below is an example of Java code and its mutated version using the conditional boundary mutator. The conditionals boundary mutator replaces the relational operators `<, <=, >, >=`
with their boundary counterpart as per the table below:
< is changed to <=
<= is changed to <
> is changed to >=
>= is changed to >
Here is an example:

Original:

```
public int adjustValue(int value) {
    if (value < 10) {
        return value * 2;
    }
    if (value >= 10 && value < 20) {
        return value + 5;
    }
```

```
    if (value >= 20) {
        return value - 3;
    }
    return value;
}
```

Mutated:

```
public int adjustValue(int value) {
    if (value <= 10) {  // This condition is mutated
        return value * 2;
    }
    if (value >= 10 && value < 20) {
        return value + 5;
    }
    if (value >= 20) {
        return value - 3;
    }
    return value;
}
```

Notice how the mutator was only applied once.
Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Do not include any labels, annotations,

other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

returns.EmptyObjectReturnValsMutator
Below is an example of Java code and its mutated version using the EmptyObjectReturnVals mutator. This mutator replaces a return value that is not already empty with an empty value for that type, as per the table below:
java.lang.String -> " "
java.util.Optional -> Optional.empty()
java.util.List -> Collections.emptyList()
java.util.Collection -> Collections.emptyList()
java.util.Set -> Collections.emptySet()
java.lang.Integer -> 0
java.lang.Short -> 0
java.lang.Long -> 0
java.lang.Character -> 0
java.lang.Float -> 0
java.lang.Double -> 0
Here is an example:

Original:
```
public int adjustValue(int value) {
    if (value < 10) {
        return value * 2;
```

```
    }
    if (value >= 10 && value < 20) {
        return value + 5;
    }
    if (value >= 20) {
        return value - 3;
    }
    return value;
}
```

Mutated:

```
public int adjustValue(int value) {
    if (value <= 10) {  // This condition is mutated
        return value * 2;
    }
    if (value >= 10 && value < 20) {
        return value + 5;
    }
    if (value >= 20) {
        return value - 3;
    }
    return 0;
}
```

Notice how the mutator was only applied once.
Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

returns.BooleanFalseReturnValsMutator
Below is an example of Java code and its mutated version using the BooleanFalseReturnValsMutator mutator, which replaces a boolean return value that is not already false with false. Here is an example:

Original:

```
public boolean isEven(int number) {
    if (number > 2) {
        return false;
    }
     return number % 2 == 0;
}
```

Mutated:

```
public boolean isEven(int number) {
    if (number > 2) {
```

```
        return false;
    }
    return false;
}
```

Notice how the mutator was only applied once.
Based on the above example, mutate the following java method, you must mutate one return value, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method:

IncrementsMutator
Below is an example of Java code and its mutated version using the Increments mutator. The increments mutator will mutate increments, decrements and assignment increments and decrements of local variables (stack variables). It will replace increments with decrements and vice versa.. Here is an example:

Original:
```
public int method(int i, int x) {
  i++;
  x++;
```

```
  return i;
}
```

Mutated:

```
public int method(int i, int x) {
  i--;
  x++;
  return i;
}
```

Notice how the mutator was only applied once.

Based on the above example, mutate the following java method, but include only the entire mutated method in your response. Do not include any labels, annotations, other text, or formatting markers (e.g., ```java). Please only apply the mutation once in the following method: