Feature Gating Module

Problem Statement :

Design a feature gating module which evaluates whether the user is allowed to access a particular feature or not depending on conditional expression evaluated against user attributes

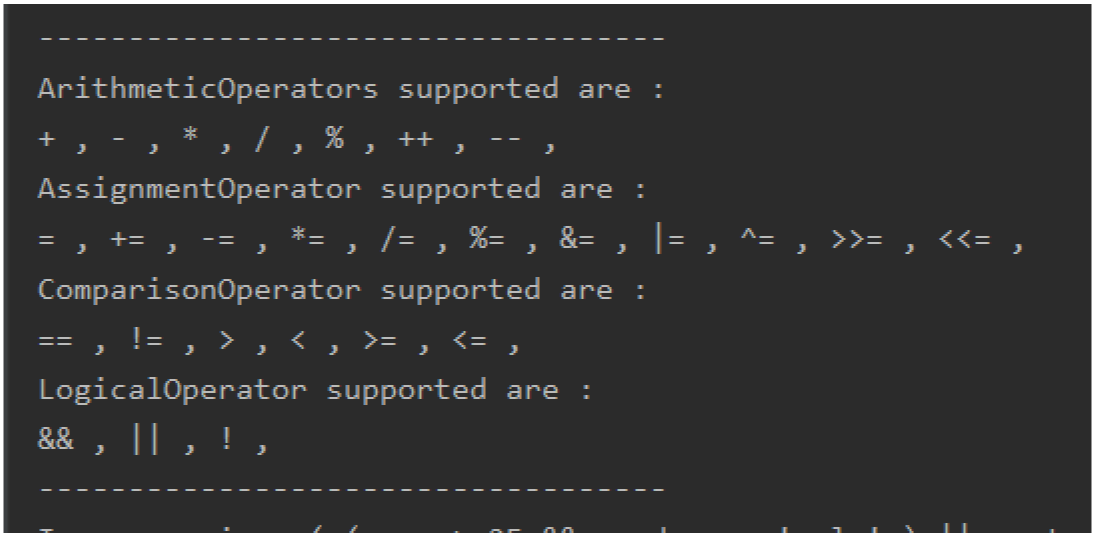
- Swati

Desciption of approach taken to solve above problem statement.

1. Functionlity Handled :

a) Operators which are supported :

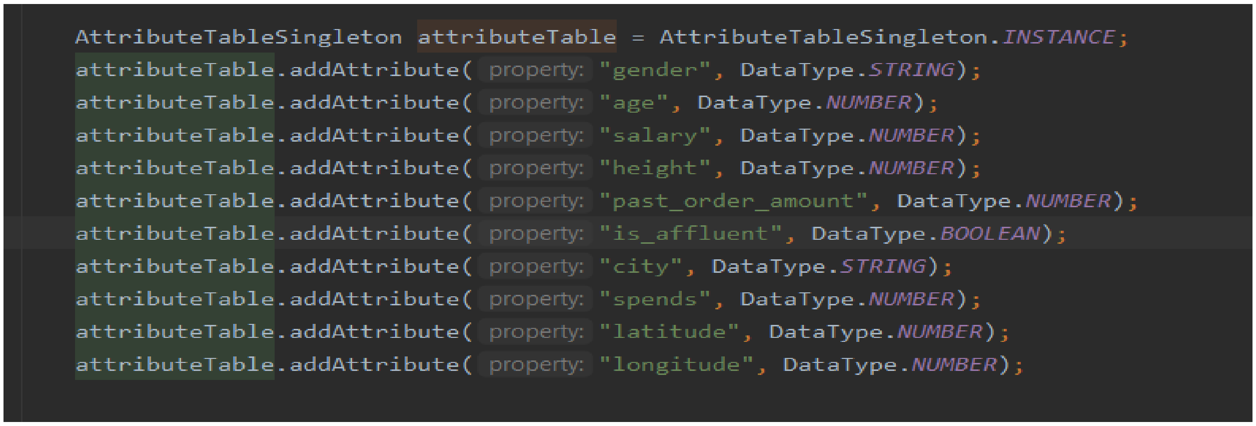
- output:



```
-----------------------------------
ArithmeticOperators supported are :
+ , - , * , / , % , ++ , -- ,
AssignmentOperator supported are :
= , += , -= , *= , /= , %= , &= , |= , ^= , >>= , <<= ,
ComparisonOperator supported are :
== , != , > , < , >= , <= ,
LogicalOperator supported are :
&& , || , ! ,
-----------------------------------
```

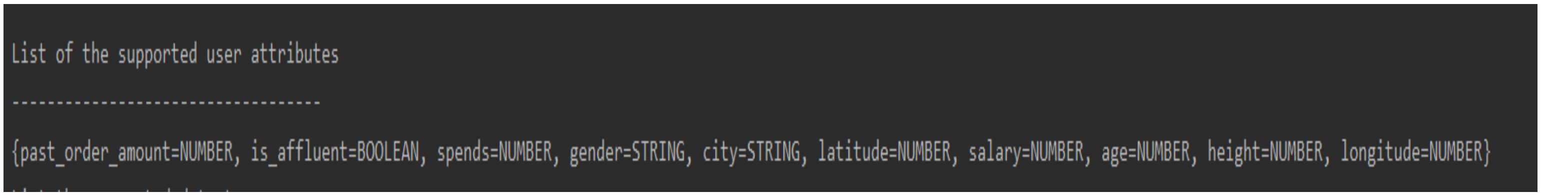
b) currently defined Attributes :

- code structure sample:



```
AttributeTableSingleton attributeTable = AttributeTableSingleton.INSTANCE;
attributeTable.addAttribute( property: "gender", DataType.STRING);
attributeTable.addAttribute( property: "age", DataType.NUMBER);
attributeTable.addAttribute( property: "salary", DataType.NUMBER);
attributeTable.addAttribute( property: "height", DataType.NUMBER);
attributeTable.addAttribute( property: "past_order_amount", DataType.NUMBER);
attributeTable.addAttribute( property: "is_affluent", DataType.BOOLEAN);
attributeTable.addAttribute( property: "city", DataType.STRING);
attributeTable.addAttribute( property: "spends", DataType.NUMBER);
attributeTable.addAttribute( property: "latitude", DataType.NUMBER);
attributeTable.addAttribute( property: "longitude", DataType.NUMBER);
```

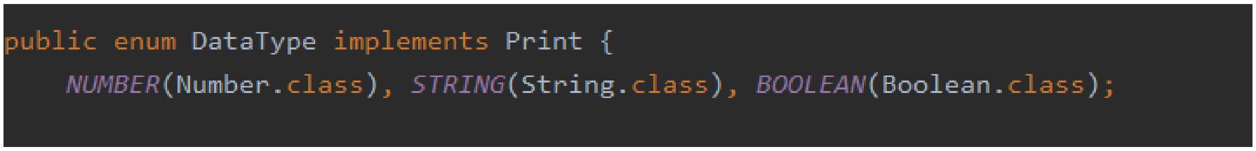
- output sample :



```
List of the supported user attributes
---------------------------------
{past_order_amount=NUMBER, is_affluent=BOOLEAN, spends=NUMBER, gender=STRING, city=STRING, latitude=NUMBER, salary=NUMBER, age=NUMBER, height=NUMBER, longitude=NUMBER}
```

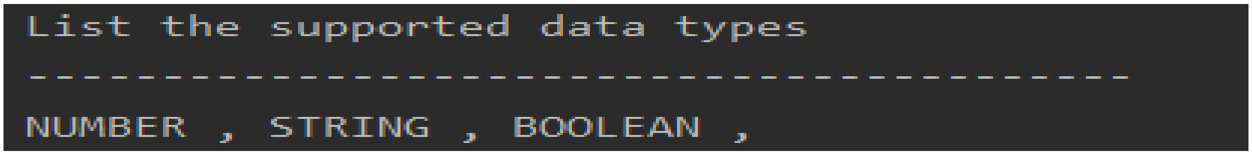
c) supported dataType:

- Code strucrure sample:



```
public enum DataType implements Print {
    NUMBER(Number.class), STRING(String.class), BOOLEAN(Boolean.class);
```

- sample output :



```
List the supported data types
-----------------------------------------
NUMBER , STRING , BOOLEAN ,
```

2. Assumption and case to handle later vs what's working .

a) In expression only single space must be maintained.

- below will work

  "((age > 25 && gender == 'male') || past_order_amount > 10000)";

- below will not work because there is no space or have multiple space :

  "((age > 25 && gender== 'male') || past_order_amount   > 10000)";

b) format of expression are supported vs format of expression not supported :

- tested format for "expression syntax" supported vs not supported :

  - "((salary > 250 || city == 'BLR') && gender == 'female')"; --> work
  - "(salary > 250)"; -- > work
  - "80 > (25 + 3)"; --> work
  - "(age > 50)"; --> work
  - "((age > (25 + 3)) && gender == 'male')" --> work

- "((age > 70))"; --> not work
- "(salary > 250 || city == 'BLR')"; --> not work

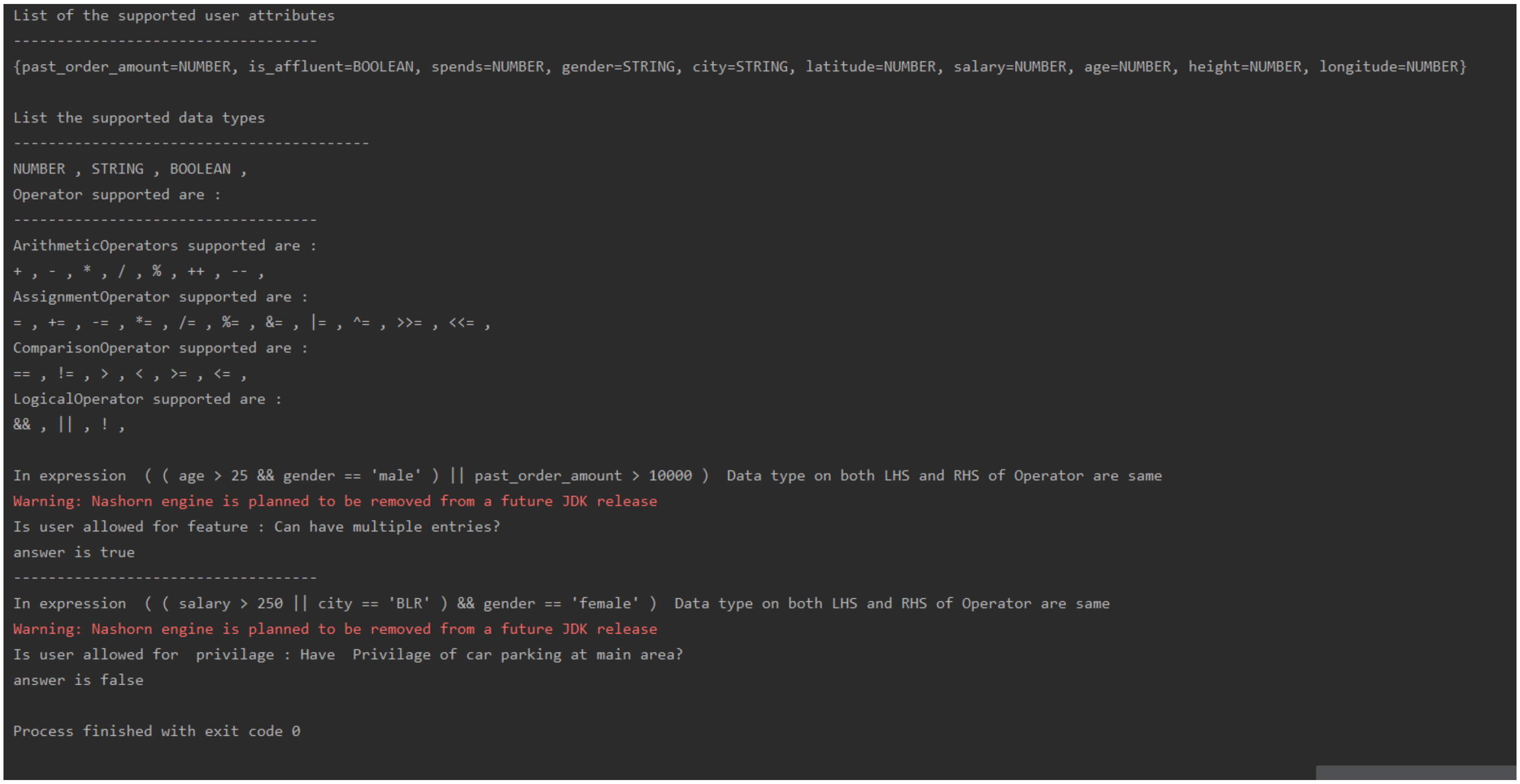
c) for now fearture and privilage added are :

- feature1 = "feature : Can have multiple entries";
- privilage1 = " privilage : Have Privilage of car parking at main area";

it can be easily extended as we have to just mention name of feature and what will be condition for it.

3) regarding classes and enums and Junit test cases, i have mentioned in code properly.

4) Final output looks like this:



```
List of the supported user attributes
----------------------------------
{past_order_amount=NUMBER, is_affluent=BOOLEAN, spends=NUMBER, gender=STRING, city=STRING, latitude=NUMBER, salary=NUMBER, age=NUMBER, height=NUMBER, longitude=NUMBER}

List the supported data types
-----------------------------------------
NUMBER , STRING , BOOLEAN ,
Operator supported are :
----------------------------------
ArithmeticOperators supported are :
+ , - , * , / , % , ++ , -- ,
AssignmentOperator supported are :
= , += , -= , *= , /= , %= , &= , |= , ^= , >>= , <<= ,
ComparisonOperator supported are :
== , != , > , < , >= , <= ,
LogicalOperator supported are :
&& , || , ! ,

In expression  ( ( age > 25 && gender == 'male' ) || past_order_amount > 10000 )  Data type on both LHS and RHS of Operator are same
Warning: Nashorn engine is planned to be removed from a future JDK release
Is user allowed for feature : Can have multiple entries?
answer is true
----------------------------------
In expression  ( ( salary > 250 || city == 'BLR' ) && gender == 'female' )  Data type on both LHS and RHS of Operator are same
Warning: Nashorn engine is planned to be removed from a future JDK release
Is user allowed for  privilage : Have  Privilage of car parking at main area?
answer is false

Process finished with exit code 0
```

--   **Thankyou**   ------------------------------