

Alunos:

Ruan Gabriel Gato Barros

Matrícula: 21553690

Rúben Jozafá Silva Belém

Matrícula: 21551560

Professor: André Carvalho

## Algoritmos de Gestão de Espaço Livre

**12/05/17**

### 1 *Building*

Para realizar a tradução de um arquivo *.jtd* para um *.dot* deve-se executar o script de nome *run.sh* com 2 parâmetros. O primeiro é o arquivo com a gramática COCO/R (*javatd.java.atg*), e o segundo é o caminho até o arquivo *.jtd* que se deseja traduzir. Exemplo:

```
1 ./run.sh javatd.java.atg jtd/rels.jtd
```

Após executar o script *run.sh*, um arquivo denominado *diagrama.dot* será gerado.

Para ler os arquivos *diagrama.dot* obtidos e gerar os diagramas UML mostrados na seção 2 deste trabalho, utilizamos o programa GraphicViz para *Windows*, porém executado no *Linux Mint 18 + Wine*.

## 2 Exemplos de Código e Diagramas

Apresentaremos dois exemplos de códigos *JTD* e os seus respectivos diagramas *UML* gerados.

### 2.1 Exemplo 1

Código do primeiro exemplo:

```
1  interface MachineLearning {}
2
3  interface SupervisedLearning extends MachineLearning {}
4
5  interface Regression extends SupervisedLearning {}
6
7  interface UnsupervisedLearning extends MachineLearning {}
8  interface Clustering extends UnsupervisedLearning {}
9  interface AssociationRuleDiscovery extends UnsupervisedLearning {}
10
11 class SupervisedLearningMethod implements SupervisedLearning {}
12
13 class Classifier implements SupervisedLearning {
14     void classify(SupervisedLearningMethod method) {}
15 }
16
17 class Bayesian extends SupervisedLearningMethod {}
18 class NaiveBayes extends Bayesian {}
19 class BayesianBeliefNetwork extends Bayesian {}
20
21 class DecisionTree extends SupervisedLearningMethod {}
22 class C45 extends DecisionTree {}
23 class M5 extends DecisionTree {}
24
25 class Ensemble implements MachineLearning {
26     Classifier[] classifiers;
27 }
```

A Figura 1 contém o diagrama UML correspondente ao código mostrado acima.

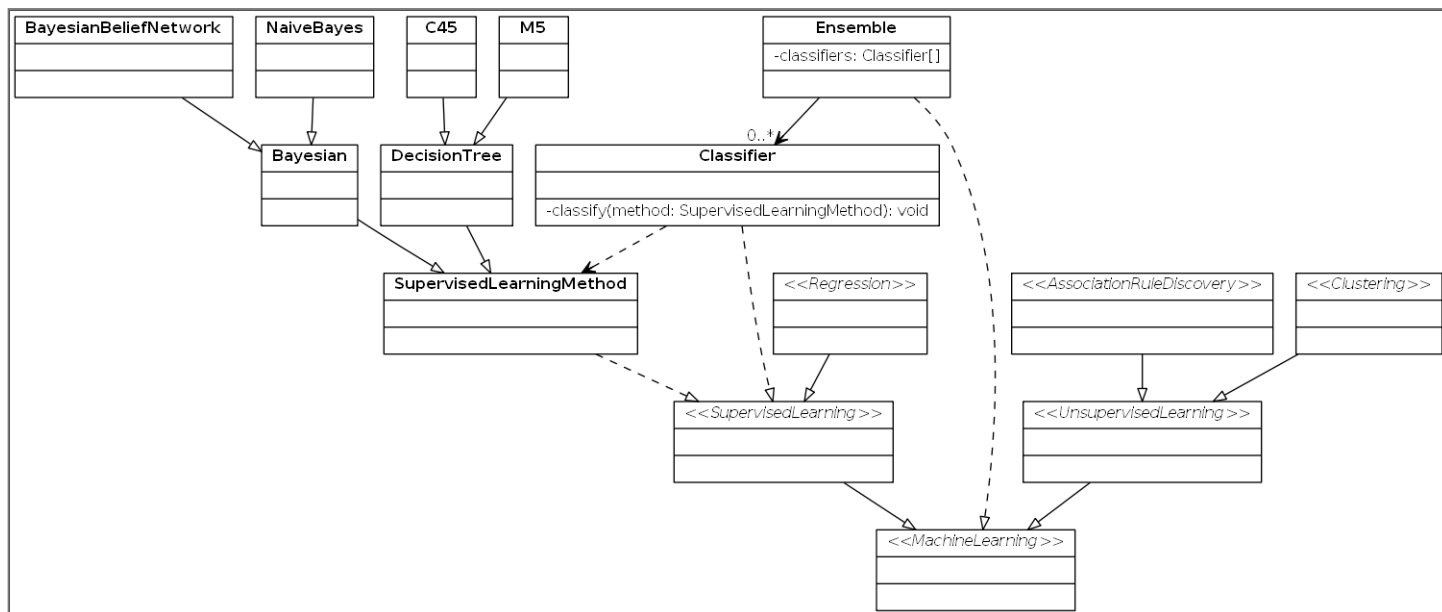


Figura 1: Diagrama de Classes UML do Exemplo 1, vindo da interpretação do arquivo *diagrama.dot* gerado pelo compilador.

## 2.2 Código 2

Código do segundo exemplo:

```
1  class Engine{
2      private Car car daggregation @ "1" to "1" title "has a";
3      private int cylinders;
4      private int volume;
5      private int gas;
6  }
7
8  class Careen{
9      private String color;
10     private static Icon icon;
11     private Car car dcomposition @ "1" to "1" title "has a";
12 }
13
14 class Car extends Automobile{
15     private Engine engine;
16     private Careen careen;
17 }
18
19 class Automobile implements Accelerable{
20 }
21
22 interface Accelerable{
23     public void xlr8();
24     public void break();
25     public static void run();
26 }
```

A Figura 2 contém o diagrama UML correspondente ao código mostrado acima.

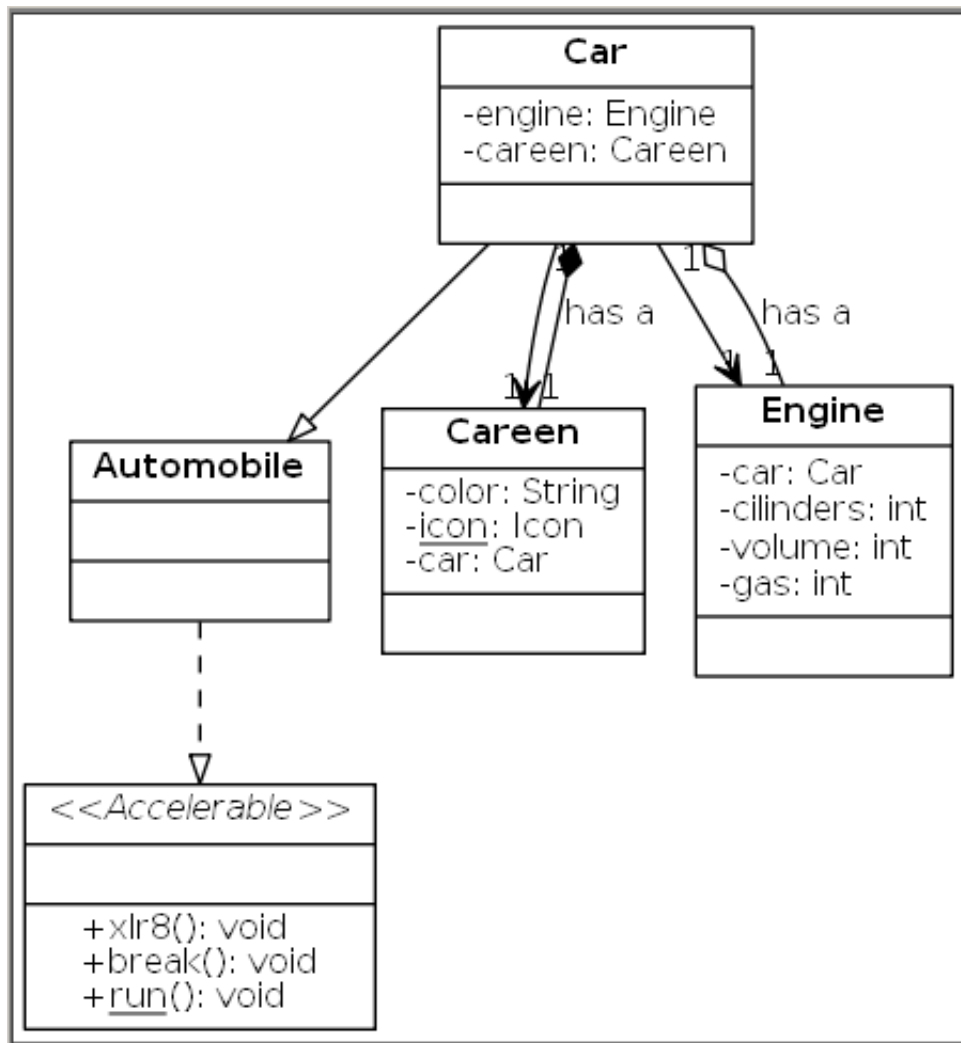


Figura 2: Diagrama de Classes UML do Exemplo 1, vindo da interpretação do arquivo *diagrama.dot* gerado pelo compilador.