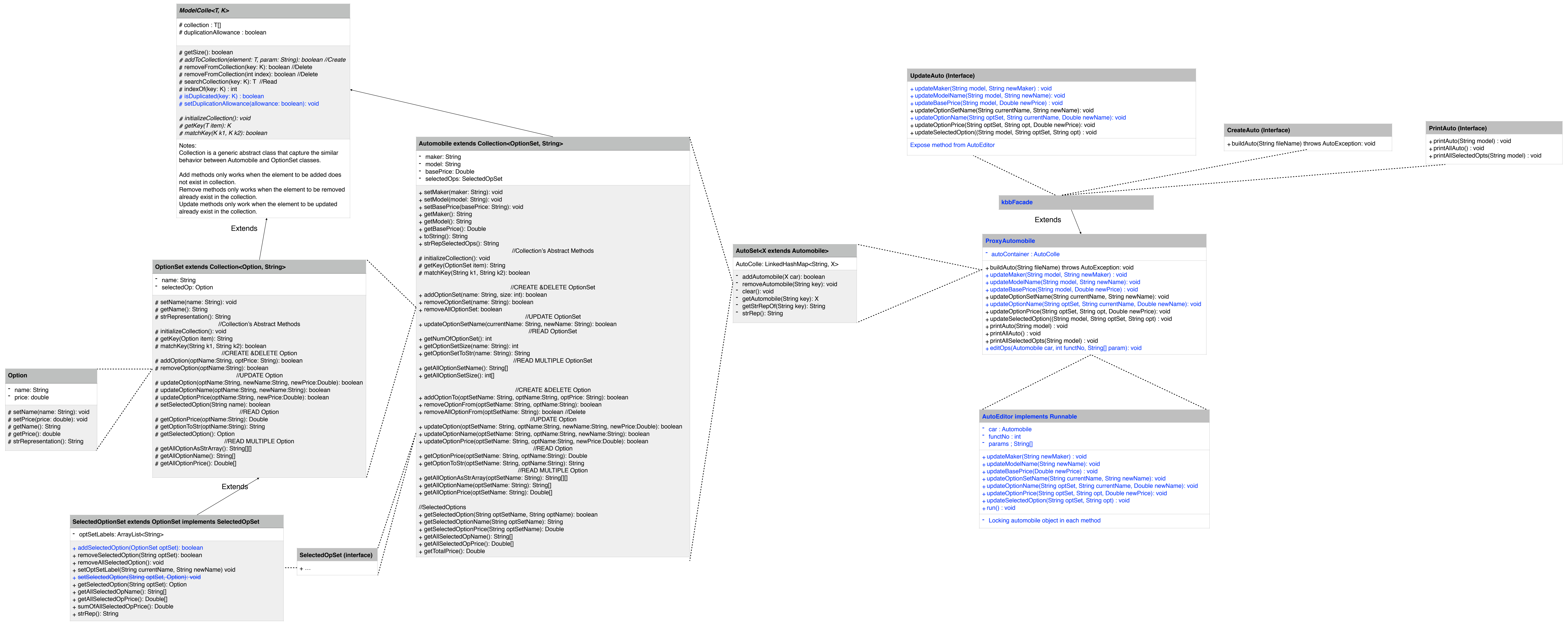


```
ModelColle<T, K>

# collection : T[]
# duplicationAllowance : boolean

# getSize(): boolean
# addToCollection(element: T, param: String): boolean //Create
# removeFromCollection(key: K): boolean //Delete
# removeFromCollection(int index): boolean //Delete
# searchCollection(key: K): T  //Read
# indexOf(key: K) : int
# isDuplicated(key: K) : boolean
# setDuplicationAllowance(allowance: boolean): void

# initializeCollection(): void
# getKey(T item): K
# matchKey(K k1, K k2): boolean

Notes:
Collection is a generic abstract class that capture the similar
behavior between Automobile and OptionSet classes.

Add methods only works when the element to be added does
not exist in collection.
Remove methods only works when the element to be removed
already exist in the collection.
Update methods only work when the element to be updated
already exist in the collection.
```

```
OptionSet extends Collection<Option, String>

-  name: String
-  selectedOp: Option

# setName(name: String): void
# getName(): String
# strRepresentation(): String
                        //Collection's Abstract Methods
# initializeCollection(): void
# getKey(Option item): String
# matchKey(String k1, String k2): boolean
                        //CREATE &DELETE Option
# addOption(optName:String, optPrice: String): boolean
# removeOption(optName:String): boolean
                             //UPDATE Option
# updateOption(optName:String, newName:String, newPrice:Double): boolean
# updateOptionName(optName:String, newName:String): boolean
# updateOptionPrice(optName:String, newPrice:Double): boolean
# setSelectedOption(String name): boolean
                              //READ Option
# getOptionPrice(optName:String): Double
# getOptionToStr(optName:String): String
# getSelectedOption(): Option
                        //READ MULTIPLE Option
# getAllOptionAsStrArray(): String[][]
# getAllOptionName(): String[]
# getAllOptionPrice(): Double[]
```

```
Option

-  name: String
-  price: double

# setName(name: String): void
# setPrice(price: double): void
# getName(): String
# getPrice(): double
# strRepresentation(): String
```

```
SelectedOptionSet extends OptionSet implements SelectedOpSet

-  optSetLabels: ArrayList<String>

+ addSelectedOption(OptionSet optSet): boolean
+ removeSelectedOption(String optSet): boolean
+ removeAllSelectedOption(): void
+ setOptSetLabel(String currentName, String newName) void
```
~~+ setSelectedOption(String optSet, Option): void~~
```
+ getSelectedOption(String optSet): Option
+ getAllSelectedOpName(): String[]
+ getAllSelectedOpPrice(): Double[]
+ sumOfAllSelectedOpPrice(): Double
+ strRep(): String
```

```
SelectedOpSet (interface)

+ ...
```

```
Automobile extends Collection<OptionSet, String>

-  maker: String
-  model: String
-  basePrice: Double
-  selectedOps: SelectedOpSet

+ setMaker(maker: String): void
+ setModel(model: String): void
+ setBasePrice(basePrice: String): void
+ getMaker(): String
+ getModel(): String
+ getBasePrice(): Double
+ toString(): String
+ strRepSelectedOps(): String
                        //Collection's Abstract Methods
# initializeCollection(): void
# getKey(OptionSet item): String
# matchKey(String k1, String k2): boolean

                        //CREATE &DELETE OptionSet
+ addOptionSet(name: String, size: int): boolean
+ removeOptionSet(name: String): boolean
+ removeAllOptionSet: boolean
                              //UPDATE OptionSet
+ updateOptionSetName(currentName: String, newName: String): boolean
                               //READ OptionSet
+ getNumOfOptionSet(): int
+ getOptionSetSize(name: String): int
+ getOptionSetToStr(name: String): String
                        //READ MULTIPLE OptionSet
+ getAllOptionSetName(): String[]
+ getAllOptionSetSize(): int[]

                        //CREATE &DELETE Option
+ addOptionTo(optSetName: String, optName:String, optPrice: String): boolean
+ removeOptionFrom(optSetName: String, optName:String): boolean
+ removeAllOptionFrom(optSetName: String): boolean //Delete
                              //UPDATE Option
+ updateOption(optSetName: String, optName:String, newName:String, newPrice:Double): boolean
+ updateOptionName(optSetName: String, optName:String, newName:String): boolean
+ updateOptionPrice(optSetName: String, optName:String, newPrice:Double): boolean
                               //READ Option
+ getOptionPrice(optSetName: String, optName:String): Double
+ getOptionToStr(optSetName: String, optName:String): String
                        //READ MULTIPLE Option
+ getAllOptionAsStrArray(optSetName: String): String[][]
+ getAllOptionName(optSetName: String): String[]
+ getAllOptionPrice(optSetName: String): Double[]

//SelectedOptions
+ getSelectedOption(String optSetName, String optName): boolean
+ getSelectedOptionName(String optSetName): String
+ getSelectedOptionPrice(String optSetName): Double
+ getAllSelectedOpName(): String[]
+ getAllSelectedOpPrice(): Double[]
+ getTotalPrice(): Double
```

```
AutoSet<X extends Automobile>

AutoColle: LinkedHashMap<String, X>

-  addAutomobile(X car): boolean
-  removeAutomobile(String key): void
-  clear(): void
-  getAutomobile(String key): X
-  getStrRepOf(String key): String
-  strRep(): String
```

```
UpdateAuto (Interface)

+ updateMaker(String model, String newMaker) : void
+ updateModelName(String model, String newName): void
+ updateBasePrice(String model, Double newPrice) : void
+ updateOptionSetName(String currentName, String newName): void
+ updateOptionName(String optSet, String currentName, Double newName): void
+ updateOptionPrice(String optSet, String opt, Double newPrice): void
+ updateSelectedOption((String model, String optSet, String opt) : void

Expose method from AutoEditor
```

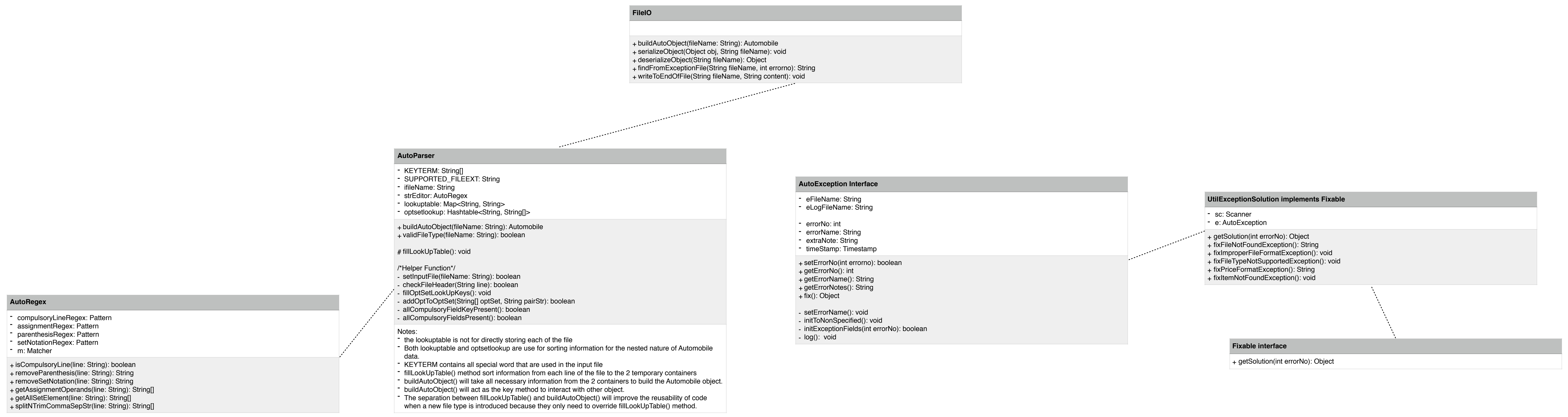
```
CreateAuto (Interface)

+ buildAuto(String fileName) throws AutoException: void
```

```
PrintAuto (Interface)

+ printAuto(String model) : void
+ printAllAuto() : void
+ printAllSelectedOpts(String model) : void
```

```
kbbFacade
```

```
ProxyAutomobile

-  autoContainer : AutoColle

+ buildAuto(String fileName) throws AutoException: void
+ updateMaker(String model, String newMaker) : void
+ updateModelName(String model, String newName): void
+ updateBasePrice(String model, Double newPrice) : void
+ updateOptionSetName(String currentName, String newName): void
+ updateOptionName(String optSet, String currentName, Double newName): void
+ updateOptionPrice(String optSet, String opt, Double newPrice): void
+ updateSelectedOption((String model, String optSet, String opt) : void
+ printAuto(String model) : void
+ printAllAuto() : void
+ printAllSelectedOpts(String model) : void
+ editOps(Automobile car, int functNo, String[] param): void
```

```
AutoEditor implements Runnable

-  car : Automobile
-  functNo : int
-  params ; String[]

+ updateMaker(String newMaker) : void
+ updateModelName(String newName): void
+ updateBasePrice(Double newPrice) : void
+ updateOptionSetName(String currentName, String newName): void
+ updateOptionName(String optSet, String currentName, Double newName): void
+ updateOptionPrice(String optSet, String opt, Double newPrice): void
+ updateSelectedOption(String optSet, String opt) : void
+ run() : void

-  Locking automobile object in each method
```



```
FileIO

+ buildAutoObject(fileName: String): Automobile
+ serializeObject(Object obj, String fileName): void
+ deserializeObject(String fileName): Object
+ findFromExceptionFile(String fileName, int errorno): String
+ writeToEndOfFile(String fileName, String content): void
```

```
AutoParser

-  KEYTERM: String[]
-  SUPPORTED_FILEEXT: String
-  ifileName: String
-  strEditor: AutoRegex
-  lookuptable: Map<String, String>
-  optsetlookup: Hashtable<String, String[]>

+ buildAutoObject(fileName: String): Automobile
+ validFileType(fileName: String): boolean

# fillLookUpTable(): void

/*Helper Function*/
- setInputFile(fileName: String): boolean
- checkFileHeader(String line): boolean
- fillOptSetLookUpKeys(): void
- addOptToOptSet(String[] optSet, String pairStr): boolean
- allCompulsoryFieldKeyPresent(): boolean
- allCompulsoryFieldsPresent(): boolean

Notes:
-  the lookuptable is not for directly storing each of the file
-  Both lookuptable and optsetlookup are use for sorting information for the nested nature of Automobile
   data.
-  KEYTERM contains all special word that are used in the input file
-  fillLookUpTable() method sort information from each line of the file to the 2 temporary containers
-  buildAutoObject() will take all necessary information from the 2 containers to build the Automobile object.
-  buildAutoObject() will act as the key method to interact with other object.
-  The separation between fillLookUpTable() and buildAutoObject() will improve the reusability of code
   when a new file type is introduced because they only need to override fillLookUpTable() method.
```

```
AutoRegex

-  compulsoryLineRegex: Pattern
-  assignmentRegex: Pattern
-  parenthesisRegex: Pattern
-  setNotationRegex: Pattern
-  m: Matcher

+ isCompulsoryLine(line: String): boolean
+ removeParenthesis(line: String): String
+ removeSetNotation(line: String): String
+ getAssignmentOperands(line: String): String[]
+ getAllSetElement(line: String): String[]
+ splitNTrimCommaSepStr(line: String): String[]
```

```
AutoException Interface

-  eFileName: String
-  eLogFileName: String

-  errorNo: int
-  errorName: String
-  extraNote: String
-  timeStamp: Timestamp

+ setErrorNo(int errorno): boolean
+ getErrorNo(): int
+ getErrorName(): String
+ getErrorNotes(): String
+ fix(): Object

- setErrorName(): void
- initToNonSpecified(): void
- initExceptionFields(int errorNo): boolean
- log():  void
```

```
UtilExceptionSolution implements Fixable

-  sc: Scanner
-  e: AutoException

+ getSolution(int errorNo): Object
+ fixFileNotFoundException(): String
+ fixImproperFileFormatException(): void
+ fixFileTypeNotSupportedException(): void
+ fixPriceFormatException(): String
+ fixItemNotFoundException(): void
```

```
Fixable interface

+ getSolution(int errorNo): Object
```