**Теоретическая часть**

**Архитектура, управляемая моделью (Model Driven Architecture, MDA)** — создаваемая консорциумом OMG разновидность концепции «Разработка, управляемая моделями»: модельно-ориентированного подхода к разработке программного обеспечения. Его суть состоит в построении абстрактной метамодели управления и обмена метаданными (моделями) и задании способов её трансформации в поддерживаемые технологии программирования (Java, CORBA, XML и др.). Создание метамодели определяется технологией моделирования MOF (Meta Object Facility), являющейся частью концепции MDA. Название концепции не совсем удачно, так как она определяет вовсе не архитектуру, а именно метод разработки программного обеспечения.

**Практическая часть**

1. **Создание MDA модели:**

При разработке модели воспользуемся инструментом, который позволяет разрабатывать UML модель (<https://habr.com/post/150041/> ) приложения и по ней генерировать код, в данном примере будет использоваться бесплатный инструмент:

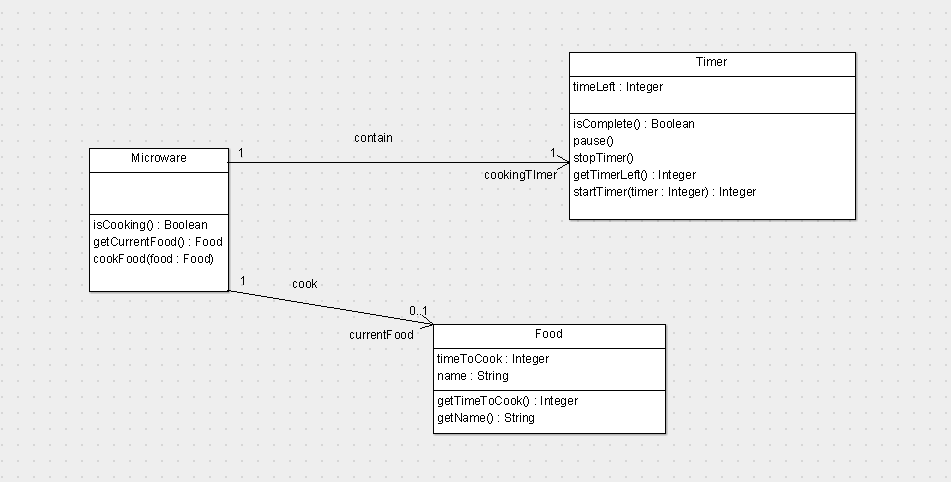
<http://argouml.tigris.org/>

Для примера разработаем приложение показывающее жизненный цикл Микроволновой печи.

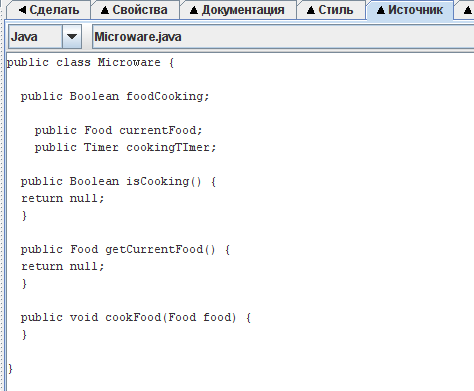
Следовательно, у нас будет 3 основных сущности:

* Еда, для неё мы будем хранить информацию о названии и времени приготовления.
* Таймер, в нем будет хранится информация об оставшемся времени приготовления и методы по запуску таймера, его приостановке и остановке таймера
* Микроволновка, в ней будет хранится информация о готовящейся еде или её отсутствию и внутреннем таймере,

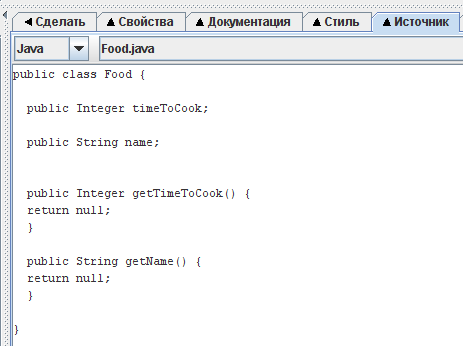
Получим такую UML модель:



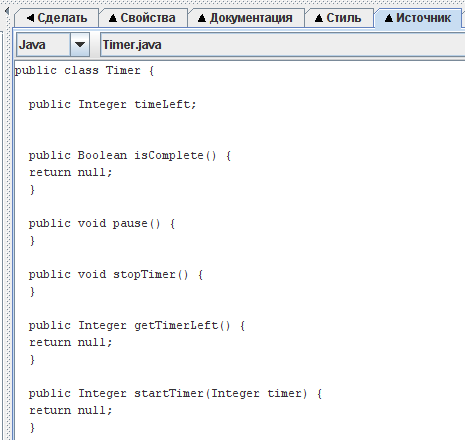
И с помощью средств AgroUML получим классы:



*Класс Микроволновка*



*Класс Еда*



*Класс Таймер*

**Теоретическая часть**

**Диаграмма состояний (statechart diagram)**

Объекты характеризуются поведением и состоянием, в котором находятся. Например, человек может быть новорожденным, младенцем, ребенком, подростком или взрослым. Другими словами, объекты что-то делают и что-то "знают". Диаграммы состояний применяются для того, чтобы объяснить, каким образом работают сложные объекты. Несмотря на то что смысл понятия "состояние" интуитивно понятен, все же приведем его определение в таком виде, в каком его дают классики и Zicom Mentor:

Состояние (state) - ситуация в жизненном цикле объекта, во время которой он удовлетворяет некоторому условию, выполняет определенную деятельность или ожидает какого-то события. Состояние объекта определяется значениями некоторых его атрибутов и присутствием или отсутствием связей с другими объектами.

Диаграмма состояний показывает, как объект переходит из одного состояния в другое. Очевидно, что диаграммы состояний служат для моделирования динамических аспектов системы (как и диаграммы последовательностей, кооперации, прецедентов и, как мы увидим далее, диаграммы деятельности). Часто можно услышать, что диаграмма состояний показывает автомат, но об этом мы поговорим подробнее чуть позже. Диаграмма состояний полезна при моделировании жизненного цикла объекта (как и ее частная разновидность - диаграмма деятельности, о которой мы будем говорить далее).

От других диаграмм диаграмма состояний отличается тем, что описывает процесс изменения состояний только одного экземпляра определенного класса - одного объекта, причем объекта реактивного, то есть объекта, поведение которого характеризуется его реакцией на внешние события. Понятие жизненного цикла применимо как раз к реактивным объектам, настоящее состояние (и поведение) которых обусловлено их прошлым состоянием. Но диаграммы состояний важны не только для описания динамики отдельного объекта. Они могут использоваться для конструирования исполняемых систем путем прямого и обратного проектирования. И они действительно с успехом применяются в таком качестве, вспомним существующие варианты "исполняемого UML", такие как UNIMOD, FLORA и др.

Но поговорим об обозначениях на диаграммах состояний. Скругленные прямоугольники представляют состояния, через которые проходит объект в течение своего жизненного цикла. Стрелками показываются переходы между состояниями, которые вызваны выполнением методов описываемого диаграммой объекта. Существует также два вида псевдосостояний: начальное, в котором находится объект сразу после его создания (обозначается сплошным кружком), и конечное, которое объект не может покинуть, если перешел в него (обозначается кружком, обведенным окружностью)..

**Практическая часть**

Для использования внутри нашего MDA приложения машины состояний нам необходимо составить диаграмму состояний.

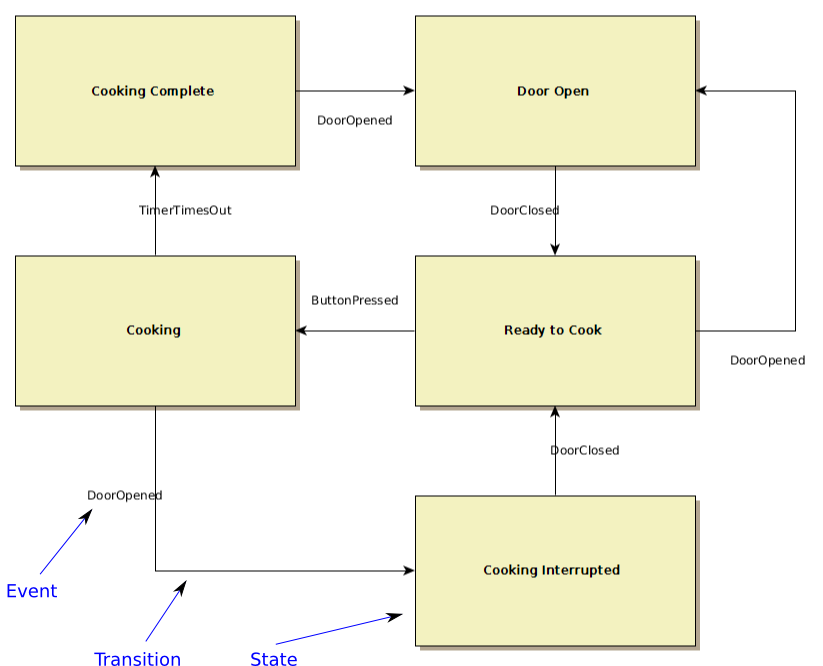
Состояниями системы-микроволной печи будут являться:

* Готова к приготовлению
* Дверца открыта
* Идет приготовление
* Приготовление завершено
* Приготовление прервано

Событиями будут являться:

* Открыли дверь
* Закрыли дверь
* Время таймера окончено
* Кнопка нажата

Получим диаграмму состояний:



Для её практической реализации воспользуемся библиотекой позволяющей реализовывать машины состояний на Java:

<https://github.com/Beh01der/EasyFlow>

Для использования данной библиотеки