Cookie jar

Implement a system simulating baking and eating cookies. A cookie baker bakes 16 cookies at a time and each "cookie eater" eats 1 cookie at a time. (Note that the number of cookies on a baking plate is 16 but could change tomorrow to e.g. 24. Therefore, let it be a variable)

Implement the following:

- A class `CookieJar` as a Monitor class holding information of the cookies in the jar in form of a list or a queue. The class may have the methods `startBaking`, `finishedBaking` and `eat`. Find out proper parameters and return types.
- A class `CookieBaker` as a thread/runnable simulating baking cookies and thereby updating the cookies in the jar (the list/queue). A cookie baker bakes a new set of cookies when the number of cookies in the jar is less than say 5 (use a parameter to set the limit). Use sleep to simulate the time for baking. Note that there is only one oven and thus, only one baker.
- A class `CookieEater` as a thread/runnable simulating eating cookies and thereby removing a cookie from the jar (the list/queue). A cookie eater waits if the jar is empty. Use sleep to simulate the time for eating.
- A class with a `main` method in which you create one baker and several eaters and starts the simulation

Insert print statements in proper places (in the monitor class) to see when the jar is empty, when eaters are eating and bakers are baking and when eaters and bakers are waiting (and who, i.e. the name of the thread).

A cookie could be an object from the following class:

```
public class Cookie
{
  private String type;

  public Cookie(String type)
  {
    this.type = type;
  }

  public String getType()
  {
    return type;
  }

  @Override public String toString()
  {
    return "Cookie: " + type;
  }
}
```