FilterFactory cleanup

Motivation:	FilterFactory2 causes some confusion
Contact:	Jody Garnett
Issue:	https://jira.codehaus.org/browse/GEOT-3056
Tagline:	Pull up methods from FilterFactory2

Children:

Description

Out of discussion with Micheal Bedward it seems that FilterFactory2 is the cause for some confusion.

Formally this is the difference between:

- FilterFactory - strictly limited to the data structure defined by Filter specifications
- FilterFactory2 - allows for extensions unique to the GeoTools project

Taking a look at the issue today I also note that many methods accept a literal ISO Geometry;
since we don't use that in practice this results in a lot of useless methods that nobody
will ever call.

Status

This proposal is under construction.

Voting has not started yet:

- Andrea Aime
- Ben Caradoc-Davies
- Christian Mueller
- Ian Turton
- Justin Deoliveira
- Jody Garnett (current OSGeo representative)
- Michael Bedward
- Simone Giannecchini

Tasks

This section is used to make sure your proposal is complete (did you remember documentation?) and has enough paid or volunteer time lined up to be a success

	no progress	✓	done	✗	impeded	⚠	lack mandate/funds/time	?	volunteer needed

1. Moves all methods into FilterFactory
    - Pull up methods into FilterFactory; update javadocs to indiacte what is "Strict" and what is a geotools extension
2. Allows methods that expected a literal ISO Geometry to accept a JTS geometry.
    - Change methods using an ISO Geometry to take a Object instead
3. Update default implementation

API Changes

BEFORE

```
public interface FilterFactory {
    ...
    /** Checks if the feature's geometry touches, but does not overlap with the
geometry held by this object. */
    Touches touches(String propertyName, Geometry geometry);
    ...
}
public interface FilterFactory2 extends FilterFactory {
    /** Checks if the feature's geometry touches, but does not overlap with the
geometry held by this object. */
    Touches     touches(Expression propertyName1, Expression geometry2);
}
```

AFTER

```
public interface FilterFactory {
    ...
    /** Checks if the feature's geometry touches, but does not overlap with the
geometry held by this object. */
    Touches touches(String propertyName, Object geometry);
    /**
     * GeoTools extension to check if a geometry touches, but does not overlap with a
second geometry.
     */
    Touches     touches(Expression propertyName1, Expression geometry2);
    ...
}
public interface FilterFactory2 extends FilterFactory {
    // empty
}
```

Documentation Changes

Code examples won't be effected by this proposal.