

Software Design Patterns

讀書會#7

2013.12.25

@qcl

讀書會#7

- Template Method Pattern
- Strategy Pattern

Template Method Pattern

Intent

- Define the skeleton of an algorithm in an operation, deferring some steps to subclasses.
- Let subclasses redefine certain steps of an algorithm without changing structure of algorithm.

Applicability

- To implement the invariant parts of an algorithm once and leave it up to subclasses to implement the behavior that can vary.
- When common behavior among subclasses should be factored and localized in a common class to avoid code duplication.
- To control subclasses extensions.

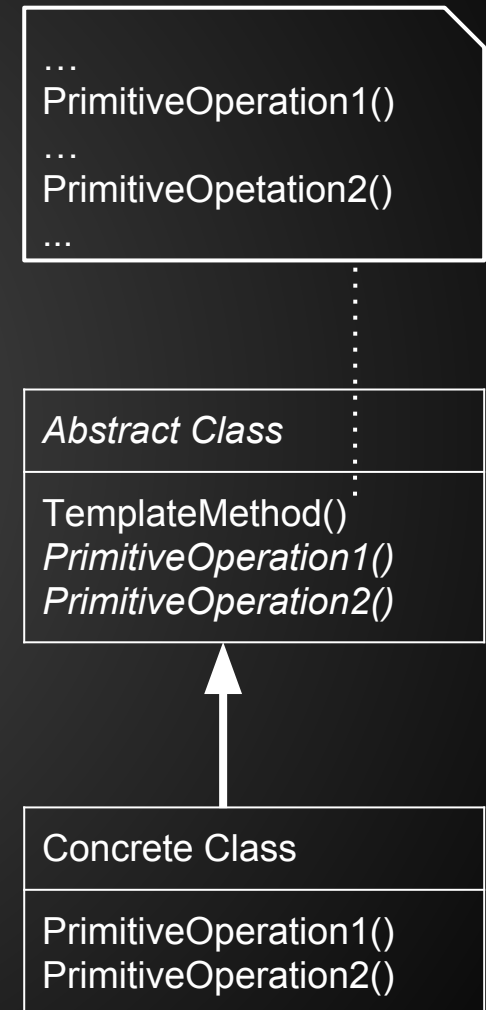
Structure & Participants

- Abstract Class

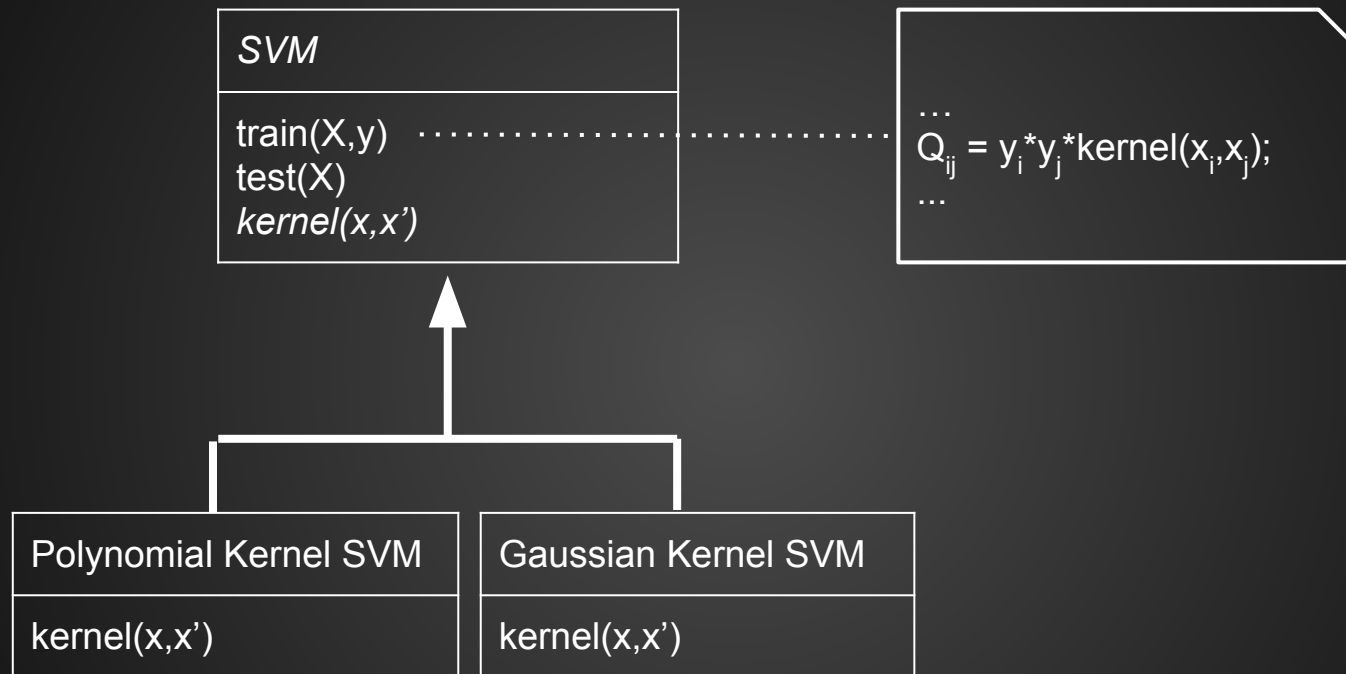
- Define abstract primitive operations that concrete subclasses define to implement steps of an algorithm
- Implements a template method defining the skeleton of an algo.

- Concrete Class

- Implements the primitive operations to carry out subclass-specific steps of algo.

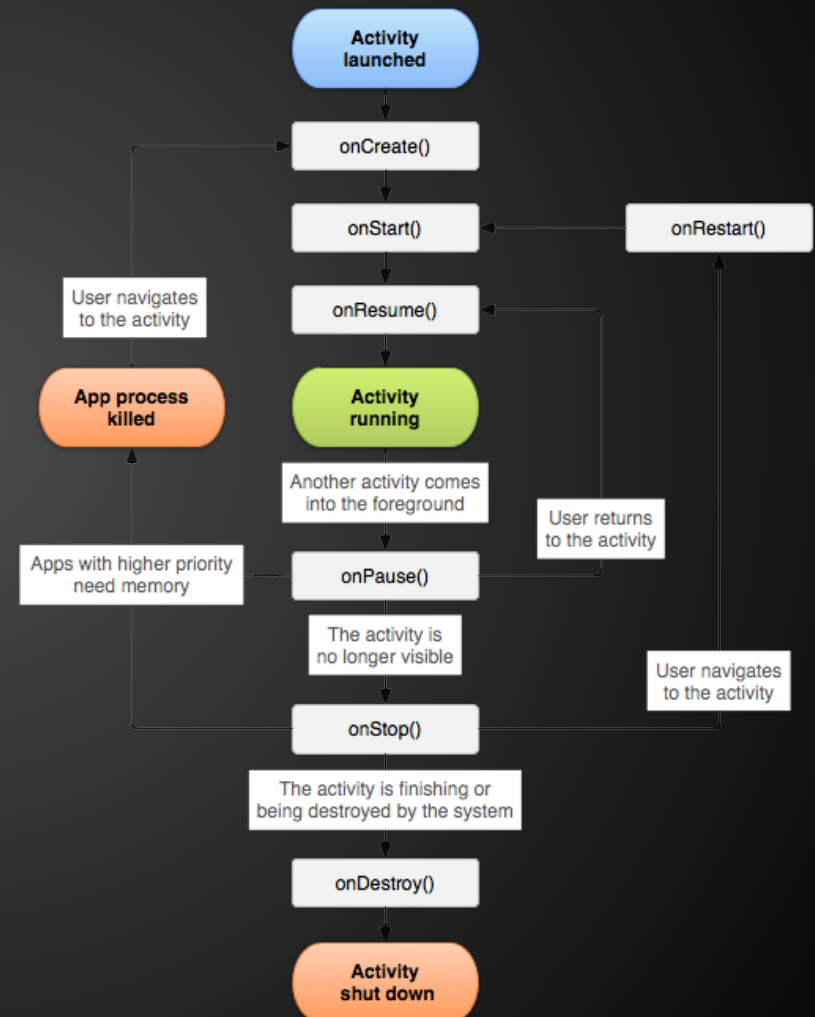


Examples



Examples

- Android Activity Lifecycle
- Every Activity extends *android.app.activity*
- And implements
 - onCreate()
 - onStart()
 - ...



Strategy Pattern

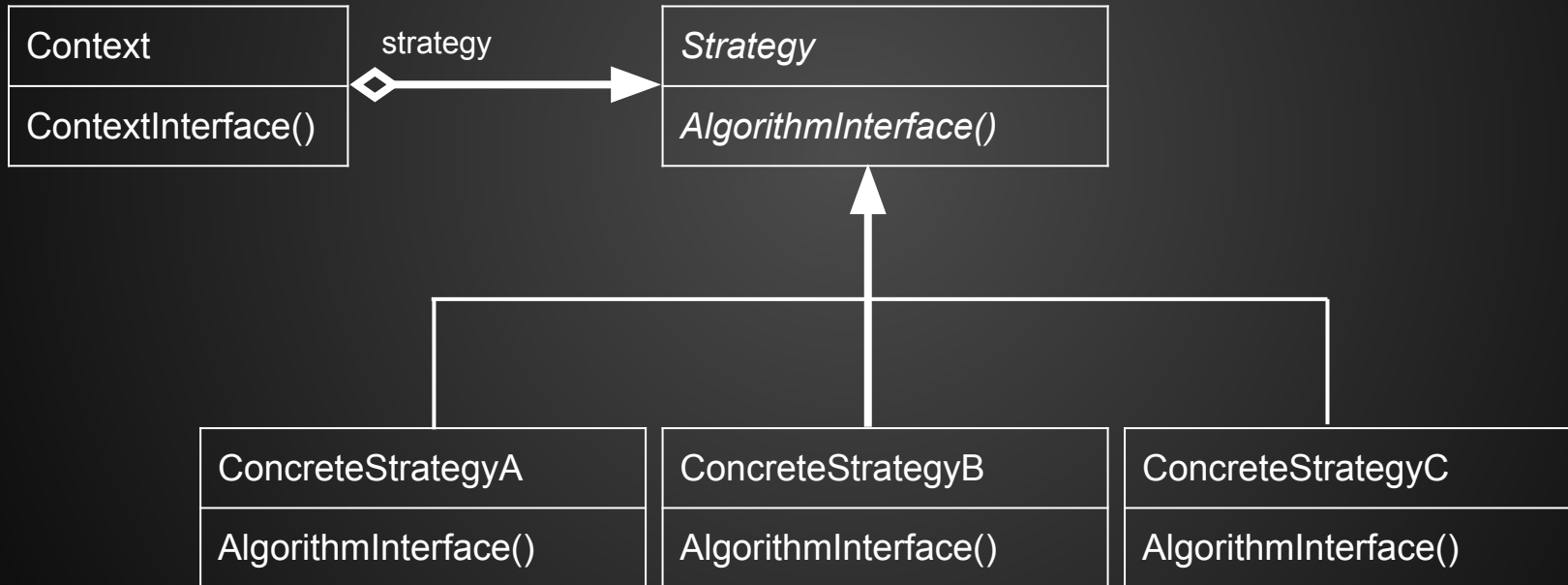
Intent

- Define a family of algorithms, encapsulate each one, and make them interchangeable.
- Strategy lets the algorithm vary independently from clients that use it.

Applicability

- Many related classes differ only in their behavior. Strategies provide a way to configure a class with one of many behaviors
- Need different variants of an algo. e.g. may define algos reflecting diff space/time trade-offs.
- Class defines many behaviors, appear as multiple conditional statements in its operations. Instead of many conditionals, move related conditional branches into their own Strategy class.

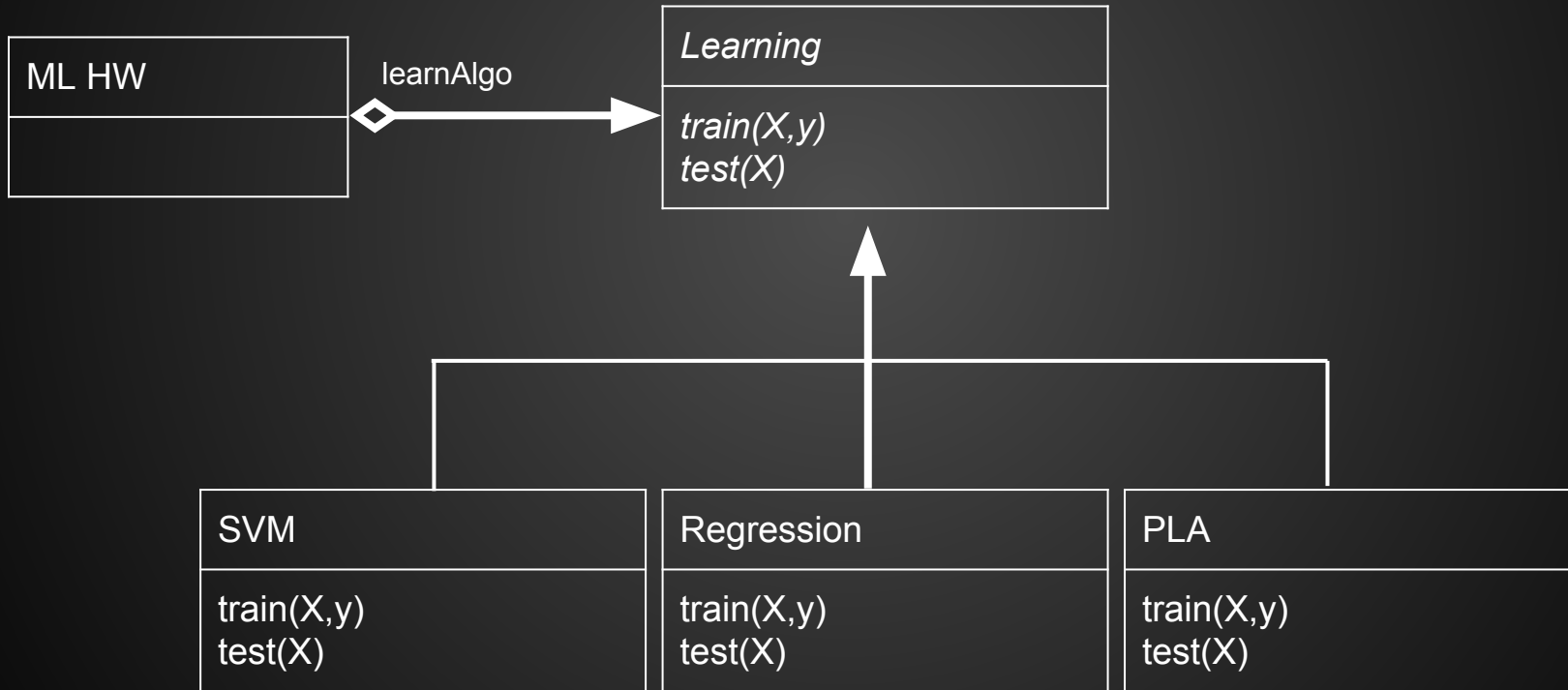
Structure



Participants

- **Strategy**
 - Declares an interface common to all supported algo
 - Context uses this interface to call algo
- **ConcreteStrategy**
 - Implements the algo using the Strategy interface
- **Context**
 - is configured with a ConcreteStrategy obj
 - maintains a ref to a Strategy obj
 - may define an interface that lets Strategy access its data

Examples



Questions?

Thanks

qingcheng.li @ qcl.tw