Ingegneria del Software

Progettazione orientata agli oggetti

Antonino Staiano
e-mail: **antonino.staiano@uniparthenope.it**

Introduzione

- Spiegare come il progetto del software può essere rappresentato come un insieme di oggetti interagenti che gestiscono il proprio stato e le proprie operazioni
- Descrivere le attività del processo di progettazione orientato agli oggetti
- Introdurre i vari modelli che possono essere usati per descrivere un progetto orientato agli oggetti

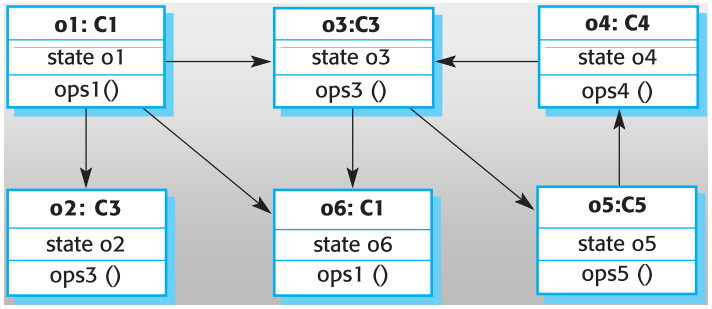
Sviluppo orientato agli oggetti

- L'analisi, la progettazione e la programmazione orientata agli oggetti sono attività tra loro legate ma distinte
  - L'analisi orientata agli oggetti riguarda lo sviluppo di un modello ad oggetti del dominio applicativo
  - La progettazione orientata agli oggetti riguarda lo sviluppo di un modello di sistema orientato agli oggetti per implementare i requisiti
    - gli oggetti del progetto sono relativi alla soluzione del problema
  - La programmazione orientata agli oggetti riguarda la realizzazione di un progetto orientato agli oggetti usando un linguaggio di programmazione orientato agli oggetti come Java o C++

Caratteristiche della progettazione orientata agli oggetti

- Gli oggetti sono astrazioni delle entità del mondo reale o del sistema che si gestiscono autonomamente
- Gli oggetti sono indipendenti ed incapsulano le informazioni sullo stato e di rappresentazione
- La funzionalità del sistema è espressa in termini di servizi dell'oggetto
- Le aree dati condivise sono eliminate. Gli oggetti comunicano mediante scambio dei messaggi
- Gli oggetti possono essere distribuiti e possono essere eseguiti sequenzialmente o parallelamente

Oggetti interagenti



Vantaggi della progettazione orientata agli oggetti

- Manutenzione più semplice. Gli oggetti possono essere capiti come entità a se stanti
- Gli oggetti sono componenti potenzialmente riusabili
- Per alcuni sistemi, può esserci un mapping ovvio dalle entità del mondo reale agli oggetti del sistema

Oggetti e classi di oggetti

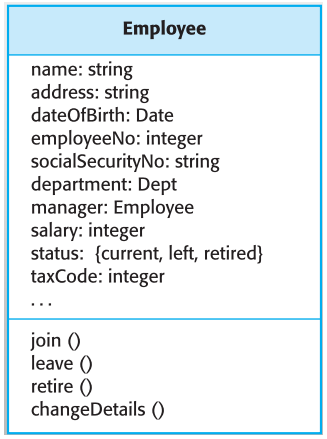
- Gli oggetti sono entità di un sistema software che rappresentano istanze delle entità del mondo reale e del sistema
- Le classi di oggetti sono schemi degli oggetti. Esse possono essere usate per creare gli oggetti
- Le classi di oggetti possono ereditare attributi e servizi da altre classi di oggetti

Oggetti e classi di oggetti

Un oggetto è un'entità che ha uno stato ed un insieme definito di operazioni che operano sullo stato. Lo stato è rappresentato come un insieme di attributi dell'oggetto. Le operazioni associate con l'oggetto forniscono servizi ad altri oggetti (i clienti) che richiedono questi servizi quando è necessaria qualche computazione.

Gli oggetti sono creati in accordo a qualche definizione di classe di oggetti. Una definizione di classe di oggetti funge da schema per gli oggetti. Essa include le dichiarazioni di tutti gli attributi e dei servizi che dovrebbero essere associati con un oggetto di quella classe.

Esempio: classe di oggetti Employee (Impiegato)



Comunicazione tra oggetti

- Concettualmente, gli oggetti comunicano mediante lo scambio di messaggi
- Messaggi
  - Il nome del servizio richiesto dall'oggetto invocante;
  - Copie delle informazioni richieste per eseguire il servizio ed il nome del titolare del risultato del servizio
  - In pratica, i messaggi sono spesso implementati da chiamate di procedura
  - Nome = nome procedura;
  - Informazione = lista dei parametri

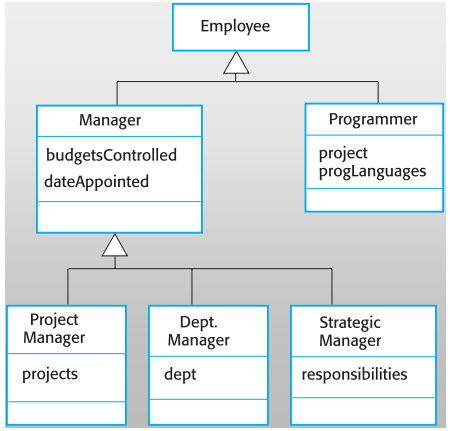
Esempi di messaggi

```
// Invoca un metodo associato con un buffer
// dell'oggetto che restituisce il valore
// successivo nel buffer

        v = circularBuffer.Get() ;

// Invoca il metodo associato con un
// oggetto termostato che imposta la
// temperatura

        thermostat.setTemp (20) ;
```


Generalizzazione ed ereditarietà

- Gli oggetti sono membri di classi che definiscono i tipi di attributo e le operazioni
- Le classi possono essere organizzate in una gerarchia delle classi dove una classe (la superclasse) è la generalizzazione di una o più altre classi (le sottoclassi)
- Una sottoclasse eredita gli attributi e le operazioni dalla propria superclasse e può aggiungere propri nuovi metodi o attributi
- La generalizzazione è implementata come ereditarietà nei linguaggi di programmazione orientati agli oggetti

Una gerarchia di generalizzazione



Vantaggi dell'ereditarietà

- E' un meccanismo di astrazione che può essere usato per classificare le entità
- E' un meccanismo di riuso in ambo i livelli di progettazione e programmazione
- Il grafo dell'ereditarietà è una sorgente di conoscenza organizzativa sui domini e sui sistemi

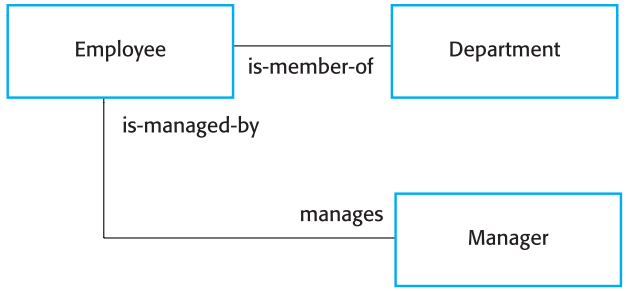
Problemi con l'ereditarietà

- Le classi di oggetti non sono autonome, nel senso che non possono essere capite senza il riferimento alle rispettive superclassi
- I progettisti hanno la tendenza a riusare i grafi dell'ereditarietà creati durante l'analisi. Ciò può portare ad inefficienze significative
- I grafi dell'ereditarietà nell'analisi, progettazione ed implementazione hanno funzioni differenti e dovrebbero essere mantenuti separatamente

Associazioni

- Gli oggetti e le classi di oggetti partecipano in relazioni con altri oggetti e classi di oggetti
- Le associazioni possono essere annotate con informazioni che descrivono l'associazione
- Le associazioni sono generali ma possono indicare che un attributo di un oggetto è un oggetto associato o che la sua implementazione si basa sull'oggetto associato

Modello di associazione



Oggetti concorrenti

- La natura degli oggetti come entità auto-contenute li rende adatti per implementazioni concorrenti
- Il modello dello scambio dei messaggi della comunicazione tra gli oggetti può essere implementato direttamente se gli oggetti sono in esecuzione su processori diversi in un sistema distribuito

Oggetti attivi e server

- Server
  - gli oggetti sono realizzati come processi paralleli con metodi corrispondenti alle operazioni definite nell'oggetto. I metodi vengono avviati in risposta ad un messaggio esterno e possono essere eseguiti in parallelo ai metodi associati ad altri oggetti. Quando le operazioni sono completate, l'oggetto si auto-sospende e attende ulteriori richieste di servizio
- Oggetti attivi
  - lo stato dell'oggetto può essere cambiato da operazioni interne che sono eseguite all'interno dell'oggetto stesso. Il processo rappresentnte l'oggetto esegue continuamente queste operazioni e quindi non viene mai sospeso

Oggetti server ed attivi

- I server sono usati in ambienti distribuiti dove gli oggetti chiamati e chiamanti possono essere eseguiti su macchine diverse
  - il tempo di risposta è imprevedibile -> bisognerebbe progettare il sistema in modo che l'oggetto chiamante non debba aspettare il completamento del servizio che invoca
- Gli oggetti attivi sono usati quando un oggetto ha necessità di di aggiornare il proprio stato ad intervalli precisi
  - usati nei sistemi real-time in cui gli oggetti sono associati a dispositivi HW che raccolgono informazioni sull'ambiente del sistema

Esempio: oggetto attivo

- Consideriamo la classe di oggetti *trasponder* di un velivolo (dispositivo ricetrasmittente che segnala la posizione del velivolo mediante un sistema di navigazione satellitare)
- L'oggetto transponder può
  - rispondere ai messaggi inviati dai computer di controllo del traffico aereo
  - fornire la posizione corrente del velivolo in risposta ad una richiesta del metodo givePosition
  - l'oggetto viene implementato tramite un thread dove un ciclo infinito nel metodo run contiene il codice per calcolare la posizione del velivolo usando i segnali dai satelliti

Oggetto transponder

```
class Transponder extends Thread {

Position currentPosition ;
Coords c1, c2 ;
Satellite sat1, sat2 ;
Navigator theNavigator ;

public Position givePosition ()
{
        return currentPosition ;
}

public void run ()
{
        while (true)
        {
                c1 = sat1.position () ;
                c2 = sat2.position () ;
                currentPosition = theNavigator.compute (c1, c2) ;
        }

}

} //Transponder
```


Oggetti concorrenti in java: Thread

- I threads in Java costituiscono il costrutto per implementare gli oggetti concorrenti
- I thread devono includere un metdo speciale, run(), che viene avviato dal sistema run-time java
- Gli oggetti attivi solitamente includono un ciclo infinito in modo da eseguire continuamente qualche computazione

Processo di progettazione orientato agli oggetti

- I processi di progettazione strutturata comportano lo sviluppo di un certo numero di modelli di sistema diversi
- Richiedono un certo sforzo per lo sviluppo ed il mantenimento di tali modelli e, per piccoli sistemi, ciò può non essere economico
- Tuttavia, per sistemi di grandi dimensioni sviluppati da diversi team i modelli di progetto costituiscono un meccanismo di comunicazione essenziale

Fasi del processo

- Si evidenziano le attività fondamentali del processo di progettazione senza far riferimento ad uno specifico processo di sviluppo
  - definizione del contesto e dei modi d'uso del sistema
  - progettazione dell'architettura del sistema
  - identificazione degli oggetti principali del sistema
  - sviluppo dei modelli del progetto
  - specifica delle interfacce degli oggetti

Esempio: sistema meteorologico

E' richiesto un sistema di previsioni meteorologiche che generi regolarmente cartine del tempo usando i dati raccolti da stazioni meteorologiche remote automatiche e da altre sorgenti dati come osservatori, satelliti e palloni meteorologici. A richiesta, le stazioni meteorologiche trasmettono i loro dati al computer d'area.

Il sistema informatico d'area convalida i dati raccolti e li integra con quelli delle diverse sorgenti. I dati integrati vengono archiviati e, utilizzando i dati di questo archivio e un database di cartine digitalizzate, viene creato un insieme di previsioni meteorologiche locali. Le cartine possono essere stampate per la distribuzione su stampanti specializzate o possono essere mostrate in una serie di formati diversi.

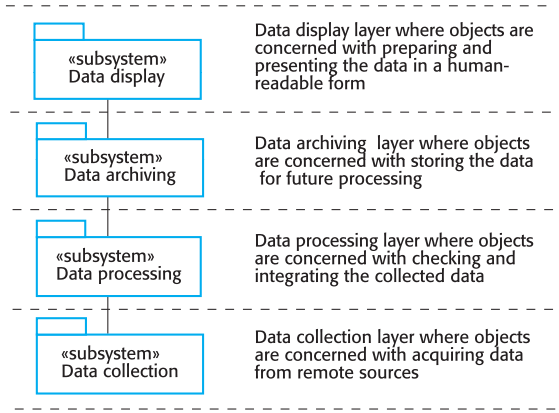
Organizzazione dell'architettura

- Una parte dell'intero sistema si occupa di raccogliere dati, una di integrare i dati dalle diverse sorgenti, un'altra di archiviare tali dati e un'altra ancora di creare mappe meteorologiche
- Si può immaginare una struttura stratificata per l'architettura di un tale sistema che riflette gli stadi di elaborazione sopra descritti
  - La stratificazione è adatta per quest'applicazione poiché ogni stadio, per le proprie operazioni, si basa sull'elaborazione del precedente

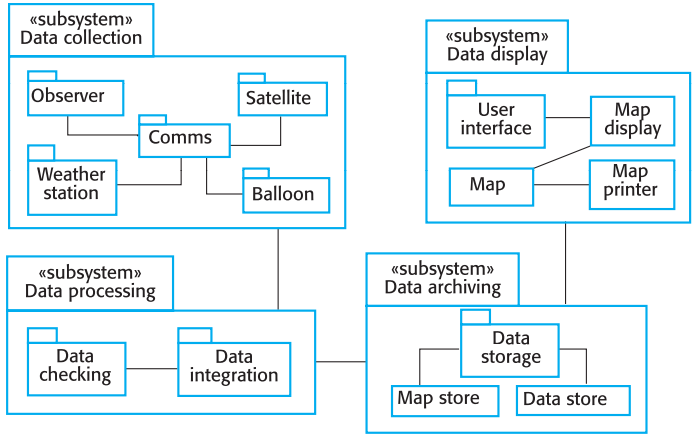
Contesto del sistema e modi d'uso

- Facciamo riferimento al sottosistema stazione meteorologica che è parte dello strato raccolta dati
- Il primo stadio della fase di progettazione consiste nello sviluppare la comprensione delle relazioni tra il software che si sta progettando ed il suo ambiente esterno
- Contesto del sistema
  - Modello statico che descrive gli altri sistemi nell'ambiente. Impiega un modello dei sottosistemi per mostrare tutti i sottosistemi coinvolti. La prossima slide mostra i sottosistemi concernenti il sistema stazione meteorologica
- Modello di utilizzo del sistema
  - E' un modello dinamico che descrive come il sistema interagisce con l'ambiente. Impiega i casi d'uso per mostrare le interazioni

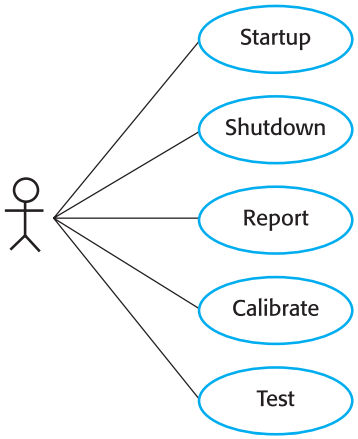
Architettura stratificata



Sottosistemi nel sistema di creazione mappe meteorlogiche



Casi d'uso per la stazione meteo



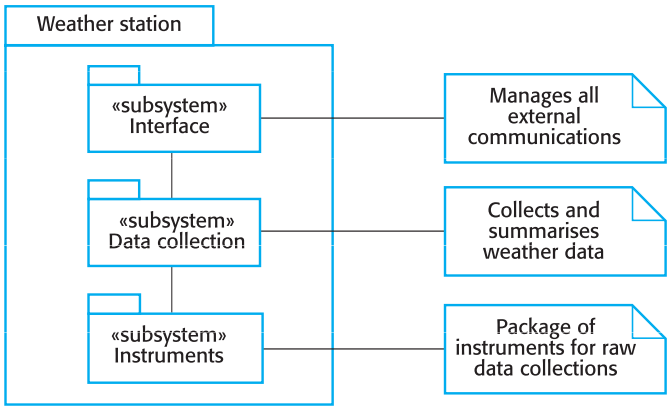
Casi d'uso per la stazione meteo

System	Weather station
Use-case	Report
Actors	Weather data collection system, Weather station
Data	The weather station sends a summary of the weather data that has been collected from the instruments in the collection period to the weather data collection system. The data sent are the maximum minimum and average ground and air temperatures, the maximum, minimum and average air pressures, the maximum, minimum and average wind speeds, the total rainfall and the wind direction as sampled at 5 minute intervals.
Stimulus	The weather data collection system establishes a modem link with the weather station and requests transmission of the data.
Response	The summarised data is sent to the weather data collection system
Comments	Weather stations are usually asked to report once per hour but this frequency may differ from one station to the other and may be modified in future.

Progettazione architetturale

- Una volta che le interazioni tra il sistema ed il suo ambiente sono state comprese, si passa alla progettazione dell'architettura del sistema
- In questo caso è appropriata un'architettura stratificata
  - strato interfaccia per la gestione delle comunicazioni
  - strato raccolta dati per gestire la strumentazione
  - strato strumentazione per la raccolta dei dati grezzi
- Non dovrebbero esserci più di sette entità in un modello architetturale

Architettura della stazione meteo



Identificazione degli oggetti

- L'identificazione degli oggetti (o classi di oggetti) è la parte più impegnativa di un progetto orientato agli oggetti
- Non c'è alcuna formula magica per l'identificazione degli oggetti. Ci si basa sull'abilità, l'esperienza e la conoscenza del dominio dei progettisti del sistema
- E' un processo iterativo, è improbabile avere un gruppo di oggetti definitivo dopo il primo passo

Approcci per l'identificazione

- Analisi grammaticale della descrizione in linguaggio naturale del sistema
- Utilizzo di entità tangibili del dominio di applicazione, per esempio, velivoli, ruoli come manager, eventi come le richieste ecc..
- Utilizzo di un approccio comportamentale in cui i progettisti identificano gli oggetti sulla base di chi partecipa e con quale comportamento
- Analisi basata sugli scenari. Sono identificati gli oggetti, gli attributi e di metodi in ogni scenario

Descrizione della stazione meteo

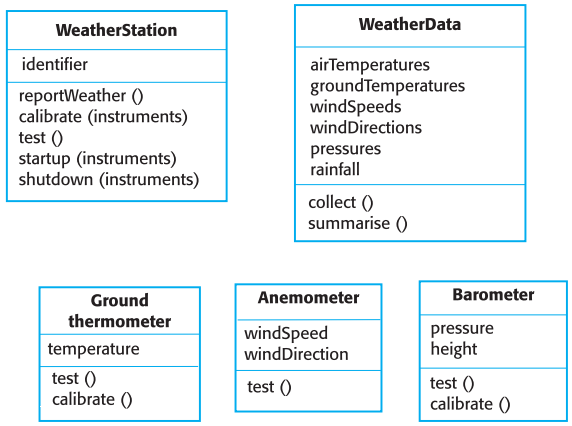
Una stazione meteo è un pacchetto di strumenti controllati via software per raccogliere dati, eseguire alcune elaborazioni e trasmettere questi dati per ulteriori elaborazioni. Le strumentazioni includono i termometri dell'aria e del suolo, un anemometro, un barometro ed un misuratore di pioggia. I dati sono raccolti periodicamente.

Quando è impostato un comando per trasmettere i dati meteo, la stazione meteo elabora e riassume i dati raccolti. I dati riassunti sono trasmessi al computer che realizza mappe quando è ricevuta una richiesta

Classi di oggetti della stazione meteo

- Termometro da suolo, Anemometro, Barometro
  - Oggetti del dominio dell'applicazione che corrispondono ad oggetti "hardware" in relazione con la strumentazione del sistema
- Stazione Meteo
  - L'interfaccia di base della stazione meteo con il suo ambiente. Riflette, pertanto, le interazioni identificate nel modello dei casi d'uso.
- Dati Meteo
  - Incapsulano i dati riepilogati dagli strumenti

Classi dioggetti della stazione meteo



Ulteriori oggetti e rifinitura degli oggetti

- Si usa la conoscenza del dominio per identificare altri oggetti e operazioni
  - Le stazioni meteo dovrebbero avere un identificatore unico
  - Le stazioni meteo sono collocate in posti remoti cosicché le strumentazioni possono riportare dati errati; gli errori devono essere riportati in maniera automatica. Pertanto sono richiesti attributi ed operazioni per un auto-check
- Oggetti attivi o passivi
  - In questo caso, gli oggetti sono passivi e raccolgono i dati a richiesta piuttosto che in modo autonomo. Ciò comporta una certa flessibilità a discapito del tempo di elaborazione del controllore

Modelli di progetti

- I modelli di progetti mostrano gli oggetti e le classi di oggetti e le relazioni tra queste entità
- I modelli statici descrivono la struttura statica del sistema in termini di classi di oggetti e relazioni
- I modelli dinamici descrivono le interazioni dinamiche tra gli oggetti

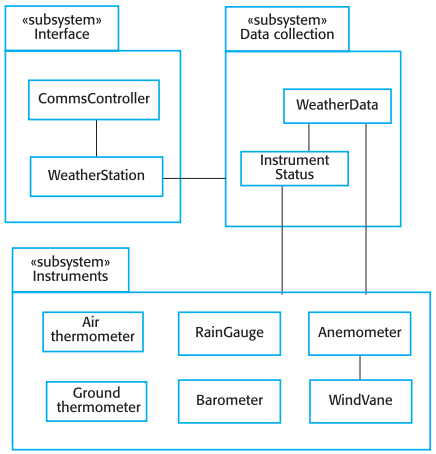
Esempi di modelli di progetto

- Modelli di sottosistema che mostrano il raggruppamento logico degli oggetti in sottosistemi coerenti
- I modelli di sequenza che mostrano la sequenza delle interazioni tra gli oggetti
- I modelli di macchine a stati che mostrano come i singoli oggetti cambiano il loro stato in risposta ad eventi specifici

Modelli di sottosistema

- Mostrano come il progetto è organizzato in un gruppo di oggetti logicamente connessi
- In UML, questi sono mostrati usando i package. Questo è un modello logico. L'organizzazione reale degli oggetti nel sistema può essere differente

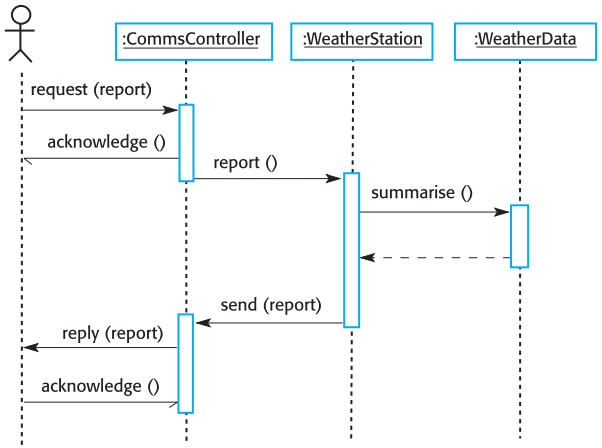
Sottosistemi della stazione meteo



Modelli di sequenza

- I modelli di sequenza mostrano la sequenza delle interazioni tra oggetti che hanno luogo
- La prossima slide mostra la sequenza di interazioni quando un sistema di mappatura esterno richiede i dati alla stazione meteorologica

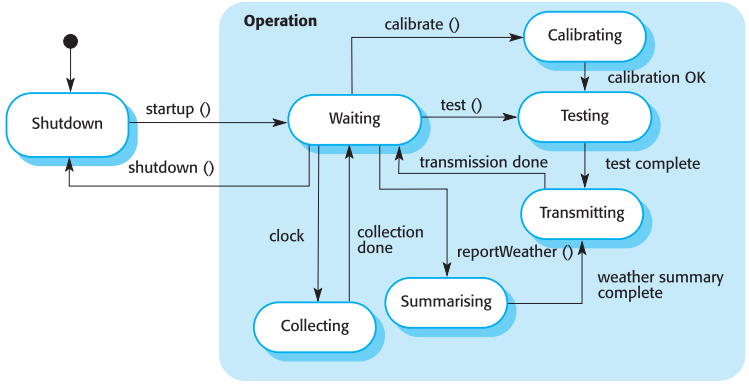
Sequenza raccolta dati



Macchina a stati

- Mostra come gli oggetti rispondono differenti richieste di servizio e le transizioni di stato innescate da tali richieste
  - Se lo stato dell'oggetto è Shutdown allora esso risponde ad un messaggio Startup()
  - Nello stato di attesa l'oggetto attende altri messaggi
  - Se è ricevuto il messaggio reportWeather() il sistema si sposta nello stato riassunto
  - Se è ricevuto un messaggio calibrate() il sistema si sposta nello stato calibrazione
  - Se viene ricevuto un segnale dall'orologio il sistema si sposta allo stato raccolta dove raccoglie le informazioni dalla strumentazione

Diagramma degli stati della stazione meteo



Specificha delle interfacce oggetto

- Le interfacce degli oggetti è necessaria affinché gli oggetti e i sottosistemi possano essere progettati in parallelo
- I progettisti dovrebbero evitare di dettagliare la rappresentazione dell'interfaccia in fase di progettazione nascondendola all'interno dell'oggetto stesso
- Gli oggetti possono avere numerose interfacce che rappresentano dei punti di vista dei metodi forniti

Interfaccia della stazione meteo

```
interface WeatherStation {

  public void WeatherStation () ;

  public void startup () ;
  public void startup (Instrument i) ;

  public void shutdown () ;
  public void shutdown (Instrument i)
;

  public void reportWeather ( ) ;

  public void test () ;
  public void test ( Instrument i ) ;

  public void calibrate ( Instrument i)
;

  public int getID () ;

} //WeatherStation
```


Evoluzione della progettazione

- Nascondere le informazioni negli oggetti significa che le modifiche fatte ad un oggetto non influenzano altri oggetti in modo non predicibile
- Supponiamo di dover aggiungere la funzionalità di controllo dell'inquinamento alla stazione meteo. Si campiona l'aria e si calcola la quantità di agenti inquinanti nell'atmosfera.
- Le letture dell'inquinamento vengono trasmesse assieme ai dati meteorologici

Cambiamenti richiesti

- Aggiungere una classe di oggetti chiamata Air quality come parte della WeatherStation
- Aggiungere un'operazione reportAirQuality a WeatherStation. Modificare il software di controllo per raccogliere le letture sull'inquinamento
- Aggiungere gli oggetti che rappresentano gli strumenti di monitoraggio dell'inquinamento
  - ossido di nitrati, fumo e benzene

Monitoraggio inquinamento