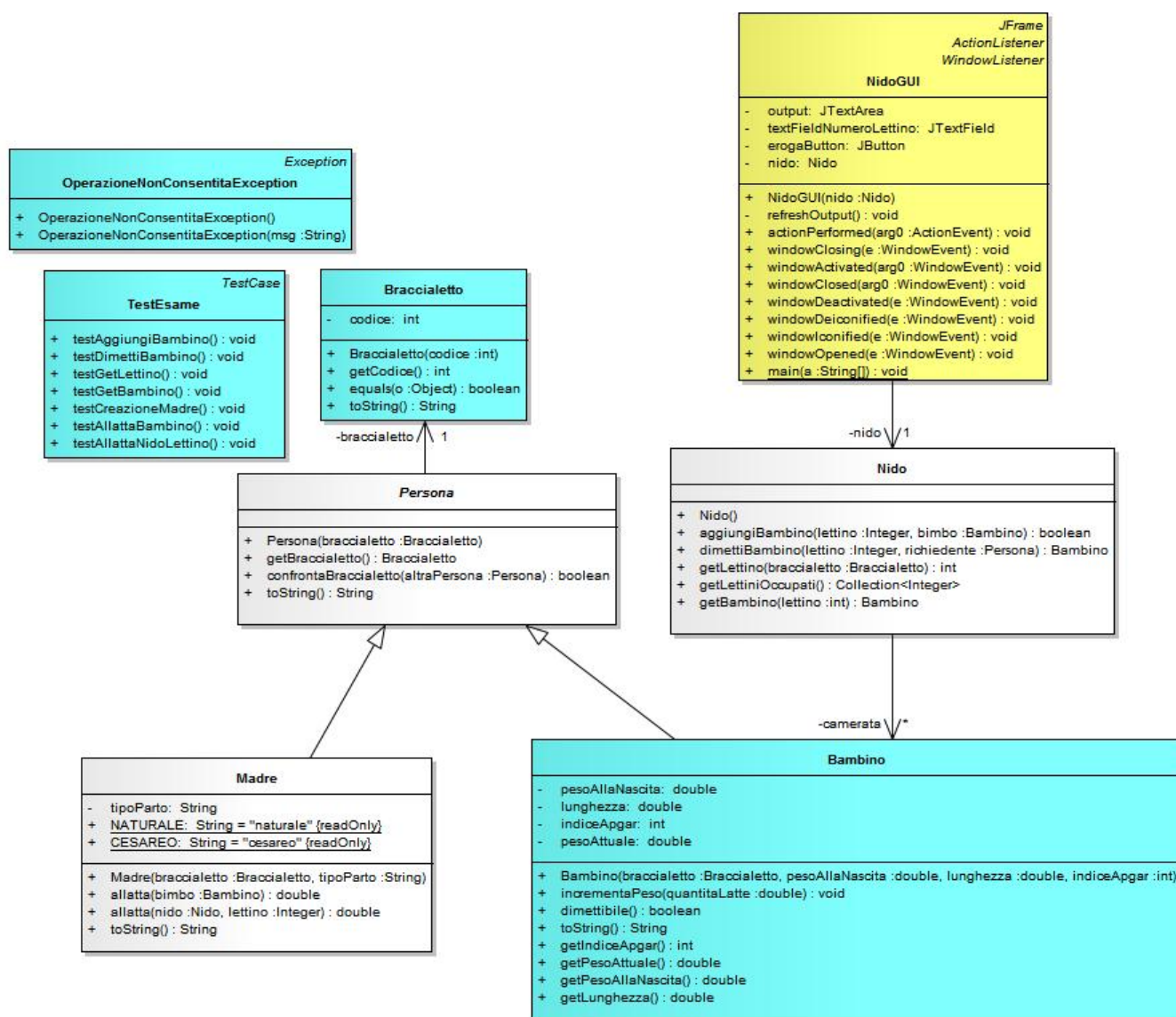


Programmazione 2

28 Giugno 2013 – Recupero Secondo Compitino

Testo parte di pratica

Il reparto Materno Infantile di un ospedale si occupa di gestire la nascita di un bambino assistendo la famiglia nei giorni immediatamente precedenti e seguenti il parto. In particolare, il personale del reparto analizza la documentazione relativa allo stato di salute della mamma e del bimbo e fornisce assistenza per le pratiche amministrative di prima necessità (ad esempio, la registrazione del bambino appena nato presso il comune di nascita o la concessione del codice fiscale). Si supponga di voler progettare e implementare una porzione di sistema informatico a supporto del reparto nello svolgimento di queste operazioni: si implementino in Java le classi descritte nel seguente diagramma UML. Le classi in azzurro sono già implementate, quelle in giallo sono parzialmente implementate, le rimanenti sono da implementare completamente.



Il modello dei dati del sistema è realizzato tramite le classi **Persona**, **Madre**, **Bambino**, **Braccialetto** e **Nido**. La prima descrive una persona, la seconda modella una mamma, caratterizzata dal tipo di parto sostenuto ("naturale" o "cesareo"). La classe **Bambino** rappresenta un bambino appena nato, caratterizzato dal peso alla nascita, dalla lunghezza alla nascita e da un intero compreso fra 0 e 10 denominato indice di Apgar che stabilisce il livello di salute del bimbo. Completa la descrizione un attributo **pesoAttuale** che specifica l'aumento di peso del bimbo a seguito di ogni poppata. **Madre** e **Bambino** sono collegati tra loro per mezzo di un **Braccialetto** (associazione con la

classe **Persona**), assegnato in sala parto immediatamente dopo la nascita, caratterizzato da un solo parametro che definisce il numero progressivo del bambino nel registro delle nascite in ospedale (ad esempio, il bimbo con braccialetto 816 sarà l'816-mo bimbo nato nell'ospedale dall'inizio dell'anno). In questo modo, non si possono effettuare scambi di bimbi tra genitori diversi. La classe **Nido** descrive il luogo in cui vengono custoditi i neonati dopo la nascita. Al nido i bambini vengono riconosciuti tramite il braccialetto e possono essere ritirati dai genitori solo se il braccialetto al polso della mamma presenta lo stesso numero di quello del bambino. A ogni bambino viene assegnato un lettino, identificato da un progressivo numerico. Le madri possono allattare i bimbi al Nido. La classe **TestEsame**, parzialmente implementata, permette di verificare il corretto funzionamento di alcuni metodi. Tale classe deve essere completata con un ulteriore metodo come descritto in seguito.

Il diagramma è completo e non richiede ulteriori metodi e attributi, ad eccezione di ciò che occorre affinché l'associazione tra Nido e Bambino possa essere realizzata rispettando le caratteristiche descritte sopra.

Classe **Persona** (da implementare)

- Rappresenta una Persona in termini di informazioni necessarie alla corretta identificazione di una famiglia: l'associazione con un'istanza della classe Braccialetto.
- Definisce un costruttore per inizializzare correttamente il suo attributo; il costruttore lancia un'eccezione `OperazioneNonConsentitaException()` se il braccialetto passato come parametro è null;
- Definisce un metodo `get` per il Braccialetto;
- Definisce un metodo `confrontaBraccialetto(altraPersona)` che consente di verificare l'appartenenza di una persona alla stessa famiglia dell'oggetto passato come parametro, tramite confronto dei loro braccialetti.
- Effettua l'override del metodo `toString()`, ritornando il codice del braccialetto della persona;

Classe **Braccialetto** (fornita):

Rappresenta il braccialetto di riconoscimento che viene applicato al bimbo appena nato e ai suoi genitori subito dopo il parto. L'unico attributo è il codice numerico attribuito al bambino secondo le modalità descritte in precedenza. Si veda il codice per dettagli.

Classe **Bambino** (fornita):

Estende la classe **Persona**, definendo un attributo che rappresenta il peso alla nascita, uno che rappresenta la lunghezza, uno che specifica l'indice di Apgar e uno che rappresenta il peso dopo l'allattamento. Il metodo `dimettibile()` permette di stabilire se il bambino possa tornare a casa dall'ospedale (ritorna true) o meno (ritorna false). Si veda il codice per dettagli.

Classe **Nido** (da implementare)

- Rappresenta il nido di un ospedale associato ad una collezione di oggetti **Bambino**. Assegna a ogni bambino un lettino, specificato da un intero che definisce la posizione del bambino nella collezione. Un bambino non cambia mai lettino per tutta la permanenza al nido. Ogni lettino può essere occupato da un solo bambino e viene liberato quando un bambino viene dimesso.
- Definisce un costruttore che istanzia opportunamente la collezione per l'implementazione del nido;
- Definisce un metodo `aggiungiBambino(lettino, bimbo)` che permette di aggiungere il bambino specificato dal secondo parametro collocandolo nel lettino specificato dal primo parametro, verificando che il bambino non sia già presente nella camerata e che il lettino non sia già occupato; se l'operazione va a buon fine ritorna true, false altrimenti;
- Definisce un metodo `dimettiBambino(lettino, richiedente)` che verifica che il bambino assegnato al lettino passato come parametro sia presente al nido e possa essere dimesso. Verifica anche che l'oggetto **Persona** passato come parametro sia un genitore del bambino. In caso affermativo, restituisce il **Bambino** liberando il lettino assegnatogli, in caso contrario ritorna null;
- Definisce un metodo `getLettiniOccupati()` che ritorna una collezione contenente i lettini attualmente assegnati a bambini;

- Definisce un metodo `getBambino (int lettino)` che ritorna il Bambino assegnato al lettino passato come parametro;
- Definisce un metodo `getLetto(bambino)` che ritorna il numero del lettino occupato dal bambino in possesso del bambino passato come parametro. Se tale bambino esiste, ritorna il lettino corrispondente, altrimenti ritorna -1;

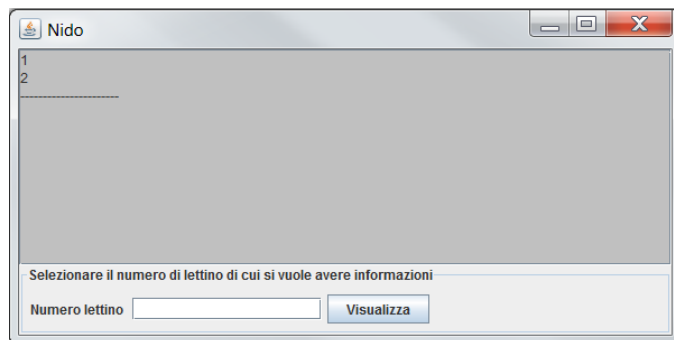
Classe **Madre** (da implementare)

- Estende la classe `Persona`, definendo un attributo che specifica il tipo di parto e due costanti `NATURALE` e `CESAREO` per il controllo del tipo di parto;
- Definisce un costruttore per inizializzare l'attributo; il costruttore solleva l'eccezione `OperazioneNonConsentitaException()` se il tipo parto è diverso da "naturale" o "cesareo";
- Definisce un metodo `allatta(bimbo)` che permette di simulare l'operazione di allattamento del bambino passato come parametro. Il metodo controlla che il bimbo passato sia effettivamente il figlio della mamma (attraverso il metodo `confrontaBambino()`), in caso contrario lancia l'eccezione `OperazioneNonConsentitaException()`; determina la quantità di latte da erogare come prodotto del peso alla nascita del bambino per la lunghezza, diviso 10 se il tipo parto è "cesareo" o diviso 5 se il tipo parto è "naturale". Infine, aggiorna il parametro peso del bambino incrementandolo della quantità di latte erogata. Restituisce la quantità di latte erogata;
- Definisce un metodo `allatta(nido, lettino)` che cerca all'interno del nido il bambino corrispondente al lettino passato come parametro. Si comporta come il metodo `allatta(bimbo)`;
- Effettua l'override del metodo `toString()`, includendo il tipo parto.

Le classi sopra descritte devono essere testate mediante la classe **TestEsame** che contiene un insieme di test già implementati.

Solo dopo aver verificato la corretta esecuzione di tutti i test è possibile svolgere la parte inerente le interfacce grafiche descritte di seguito.

L'interfaccia grafica realizzata dalla classe `NidoGUI` si presenta come segue:



L'interfaccia grafica visualizza i numeri di lettino attualmente occupati e permette agli utenti la visualizzazione delle informazioni del bambino corrispondente inserendo il numero nel textField *Numero Lettino* e premendo il pulsante *Visualizza*. A tal fine, la classe `NidoGUI` è stata dichiarata *listener* per eventi di tipo `ActionEvent`, e il pulsante *Visualizza* è stato impostato per inviare gli eventi di tipo `ActionEvent` all'istanza della classe `NidoGUI`.

Lo studente implementi il metodo `actionPerformed` in modo tale che quando si clicca sul pulsante *Visualizza*, siano visualizzate le informazioni del bambino il cui numero di lettino è stato inserito dall'utente, oppure la scritta "Letto non occupato" se il numero inserito non corrisponde ad alcun bimbo nel nido. Le informazioni devono essere visualizzate immediatamente dopo la lista dei numeri di lettino e la linea tratteggiata, sovrascrivendo di volta in volta informazioni di "visualizzazioni" precedenti.