Indice generale

Descrizione e Scopo

“La Crittologia è la scienza delle scritture nascoste, una disciplina fondamentale nel mondo di oggi e nell'informatica attuale.

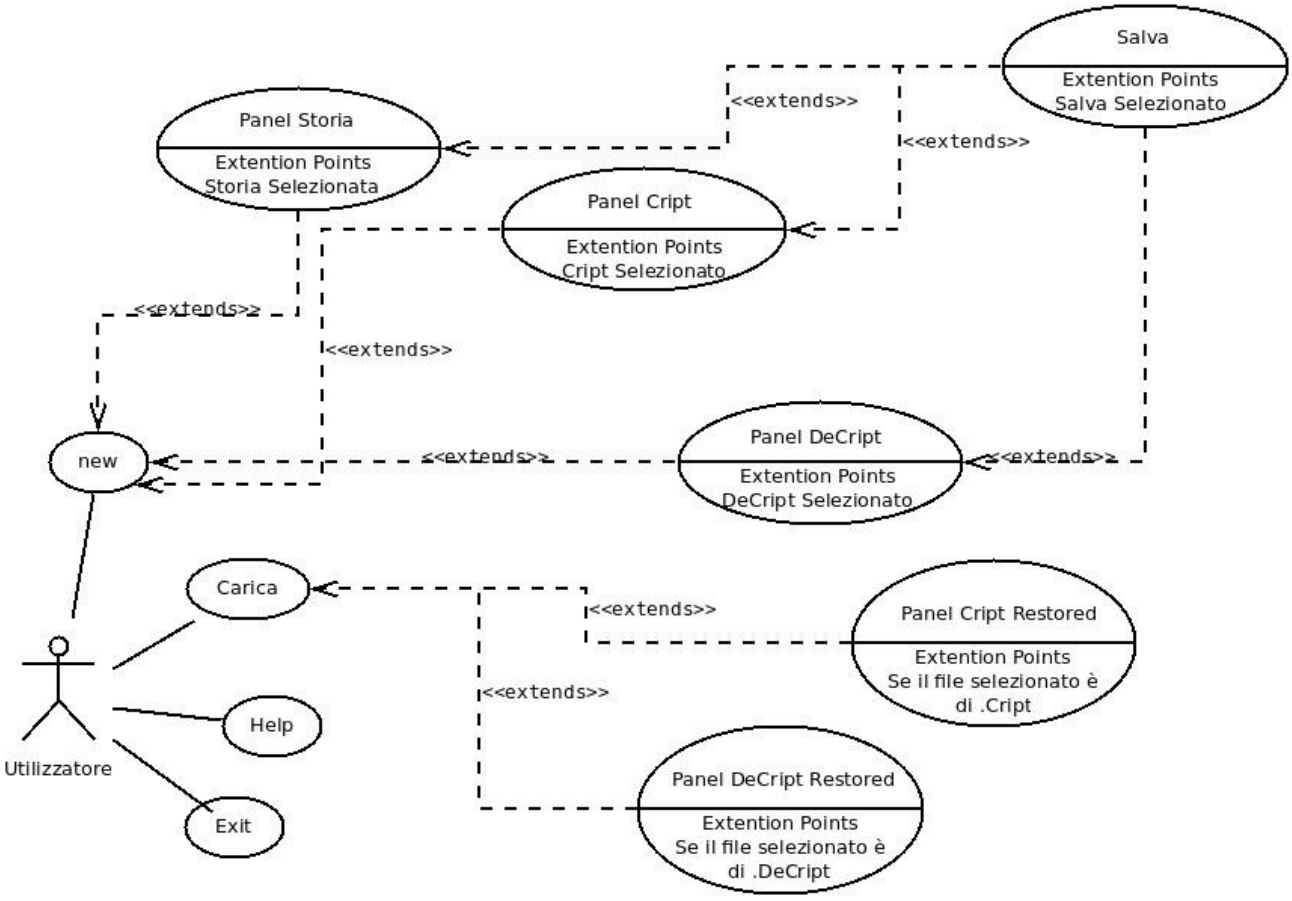
Crittografia e Crittanalisi sono le due facce della stessa medaglia, una medaglia che nel corso della storia ha dato più importanza all'una o all'altra, alternativamente.”

Così recita l'introduzione scritta nell'applicazione che ha lo scopo di fornire un ottimo strumento didattico per iniziare a studiare questa disciplina; tuttavia la semplicità d'uso di questa applicazione apre le porte anche a chi vuole solamente “giocare” mandando e ricevendo messaggi criptati.

È lecito ricordare che i cifrari implementati sono storici e facili da decodificare, anche senza conoscerne le chiavi, con gli strumenti attuali. Ma è un ottima base su cui fondare uno studio futuro molto più approfondito.

1. Progettazione: Casi D'uso

L'applicazione è descritta secondo il seguente Use Case Diagram



2. Progettazione: Scelte Ambientale

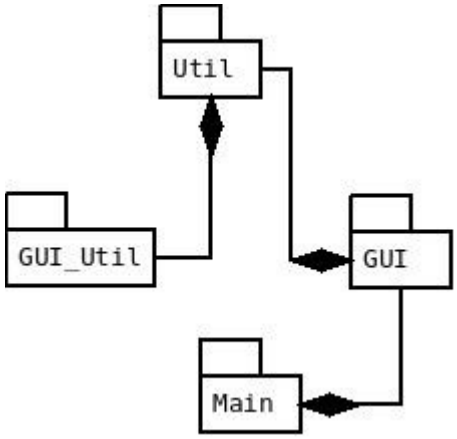
Si è deciso preventivamente, prima di sviluppare il Class Diagram, di sviluppare l'applicazione nel linguaggio di programmazione Java ed creare un eseguibile con estensione “.jar” per renderlo utilizzabile su più piattaforme senza doverlo ricompilare o apportare modifiche. Questo per favorire l'uso didattico e l'apprendimento e lo studio incrementale dei codici di criptazione stessi o della loro storia.

Questa scelta ci rende possibile di rendere non disponibili alcuni costrutti non rappresentabili in Java, come per esempio l'ereditarietà multipla, che renderà leggermente più veloce il passaggio dal Class Diagram alle classe.

3. Progettazione: Struttura delle classi

L'applicazione consta di tre package:

- GUI: che contiene le principali componenti di interfaccia utente e il main
- Util: che contiene le componenti che svolgono le principali funzioni che l'applicazione è in grado di supportare
- GUI_Util: che contiene le componenti di interfaccia utente che fanno da tramite con le impostazioni che un cifrario deve settare necessariamente per poter procedere alla criptazione o decriptazione di un codice



Package GUI

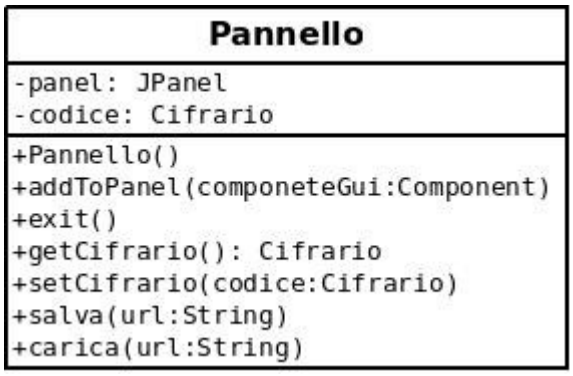
Questo package cotiene la classe “Main”, la classe “GUI_Primaria”, la classe “Pannello” e le sue sottoclassi “Cript”, “DeCript” e “Storia”, costituiscono le componenti basilari dell'interfaccia grafica.

La classe Pannello

La classe concreta“Pannello” implementa la parte di interfaccia utente che l'utente dovrà utilizzare, nonostante è una classe concreta non vengono mai creati oggetti di tipo pannello, ma viene

utilizzata per mettere a fattor comune operazioni che verranno utilizzate dalle sue sottoclassi.

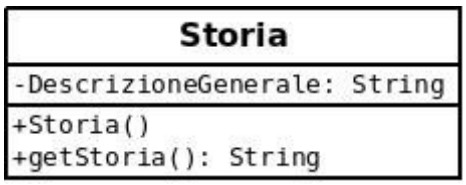
Essa non è astratta perché necessita di un costruttore.



La classe Storia

La classe “Storia” estende la classe “Pannello” e implementa la parte di interfaccia grafica che l'utente utilizzerà quando vorrà esplorare la storia e la descrizione del metodo di criptazione o decriptazione.

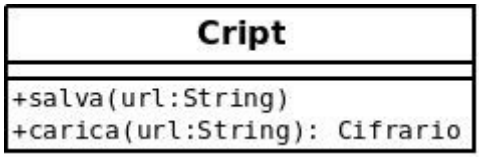
La classe “Storia” può essere in associazione con un cifrario oppure con nulla, ottenendo in output o la storia e la spiegazione del metodo selezionato, oppure un discorso generale sulla Criptologia.



La classe Cript

La classe “Cript” estende la classe “Pannello” e implementa la parte di interfaccia grafica che l'utente utilizzerà per poter criptare un testo con uno specifico codice.

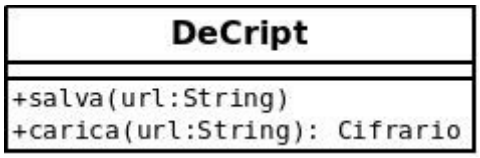
La classe “Cript” può è sempre in associazione con un cifrario, per offrire in output la criptazione del messaggio inserito nel cifrario selezionato.



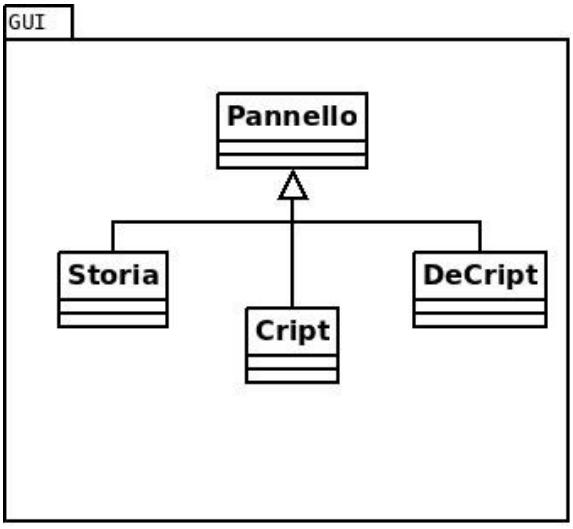
La classe DeCript

La classe “DeCript” estende la classe “Pannello” e implementa la parte di interfaccia grafica che l'utente utilizzerà per poter decriptare un testo specificandone la corretta impostazione.

La classe “DeCript” può è sempre in associazione con un cifrario, per offrire in output la decriptazione del messaggio inserito nel cifrario selezionato.



Schema totale del Packcage

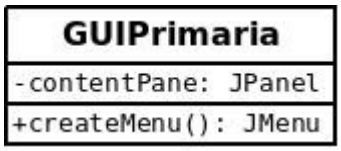


Package Main

Questo package contiene le componenti software più vicine all'utente e il metodo main.

La classe GUI_Primaria

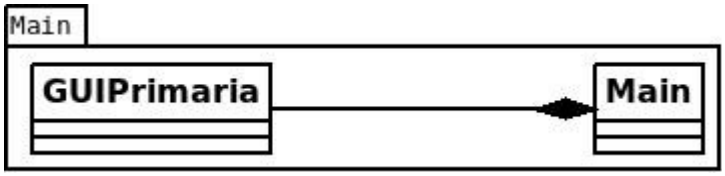
Costituisce la schermata di base che un utente può vedere, contiene la barra del menu e un oggetto di tipo “JtabbedPane” che aggiunge alla schermata varie istanze della classe “Pannello”.



La classe Main

È composta dal solo metodo “main”, che costruisce un oggetto di tipo GUI_Primari e la setta nella condizione di visibile.

Schema totale del Package



Package Util

Questo package contiene le classi di utilità, in grado di criptare e decriptare codici. Queste classi rappresentano il fulcro su cui si muove la logica dell'applicazione.

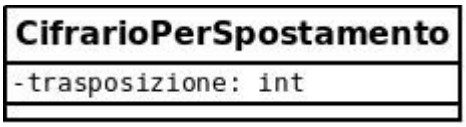
La classe Cifrario

La classe astratta “Cifrario” è la classe che mette a fattor comune tutto ciò che un qualunque cifrario ha bisogno. Tramite questa classe si implementa il pattern Abstract Factory.

La classe CifrarioPerSpostamento

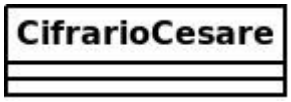
La classe “CifrarioPerSpostamento” è una classe concreta che estende “Cifrario” e concretizza i metodi astratti della sua super classe introducendo una variabile “trasposizione” necessaria alle sue impostazioni.

Funzionamento: in cifrario per trasposizione è un cifrario che trasforma una lettera in un'altra lettera successiva nell'alfabeto spostata di n posizioni.



La classe CifrarioCesare

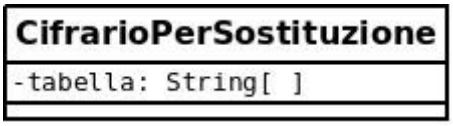
La classe “CifrarioCesare” estende la classe “CifrarioPerSpostamento” e realizza nient'altro che un particolare caso di cifrario per trasposizione in cui il valore di “trasposizione” è 1 e tutti i simboli che non sono lettere restano invariati.



La classe CifrarioPerSostituzione

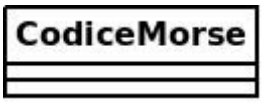
La classe “CifrarioPerSostituzione” è una classe concreta che estende “Cifrario” e concretizza i metodi astratti della sua super classe introducendo una tabella che stabilisce le regole secondo cui verranno trasformati i caratteri.

Funzionamento: il cifrario per sostituzione è un cifrario che trasforma un carattere in un altro carattere secondo una convenzione tra gli utilizzatori del codice.



La classe CodiceMorse

La classe “CodiceMorse” è una classe che estende “CifrarioPerSostituzione” e implementa un particolare metodo di cifrario per sostituzione molto usato soprattutto in passato, trasforma ogni lettera in una sequenza di “.” e “_”.



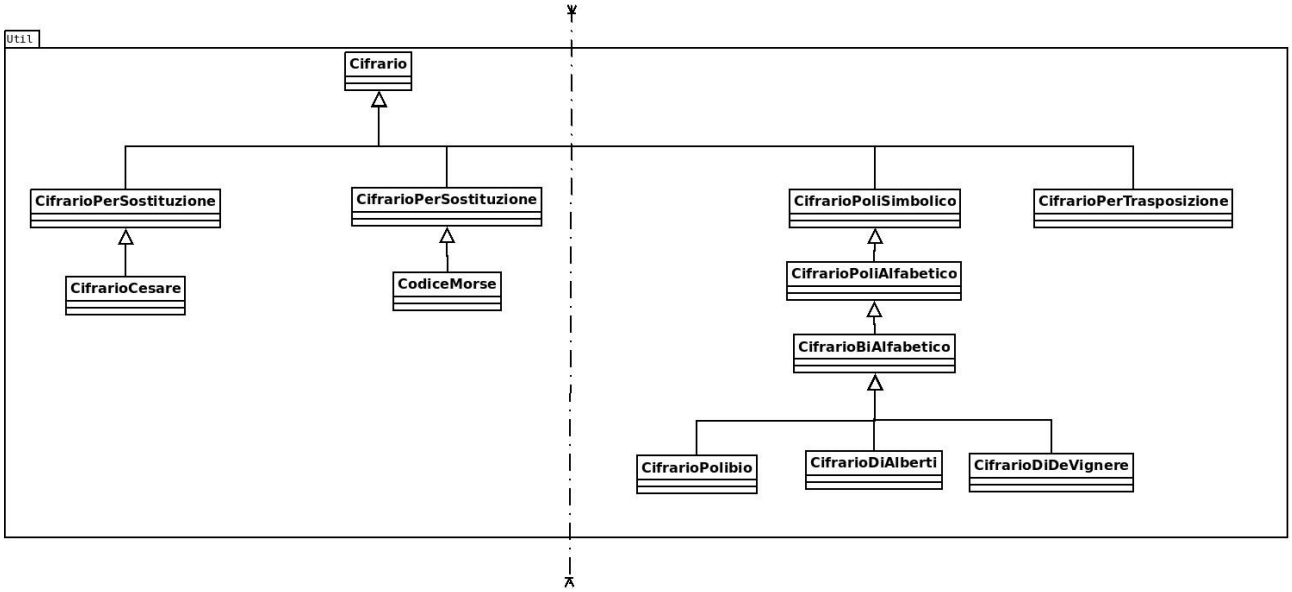
Classi Previste nelle future release

- La classe CifrarioPerTrasposizione: utilizza una striscia di caratteri che arrotolata su uno scitale (un tubo il cui diametro è la chiave per la decriptazione) diventa leggibile
- La classe CifrarioPoliSimbolico: trasforma un carattere in una serie di simboli
- La classe CifrarioPoliAlfabetico: trasforma un carattere in una serie di caratteri
- La classe CifrarioPolibio: trasforma un carattere in due caratteri in base ad una corrispondenza tabellare particolare
- La classe CifrarioDiAlberti: trasforma un carattere in due caratteri in base ad una corrispondenza tabellare particolare diversa da quella di Polibio
- La classe cifrarioDiDeVignere: trasforma un carattere in due caratteri in base a una chiave lunga a piacere (anche molto lunga es. un libro)

* è inoltre previsto nelle future release di implementare il salva e il carica.

Schema totale del Package

*a sinistra della linea tratteggiata sono presenti le classi da implementare, a destra alcune di quelle da implementare nella prossima release.



Package Util_GUI

Questo package contiene classi che permettono la creazione di menu grafici che permettono di settare opportunamente i vari tipi di cifrari.

La classe Opzioni

La classe “Opzioni” costituisce la classe padre di tutte le varie opzioni particolari per ogni tipologia di cifrario. Mette a fattor comune l'interfaccia opzioni di base.

La classe OpzioniCifrarioPerSpostamento

La classe “OpzioniCifrarioPerSpostamento” estende la classe “Opzioni” e realizza le opzioni particolari della corrispondente classe “CifrarioPerSpostamento”, fornendo un JSpinner per settare il valore di spostamento.

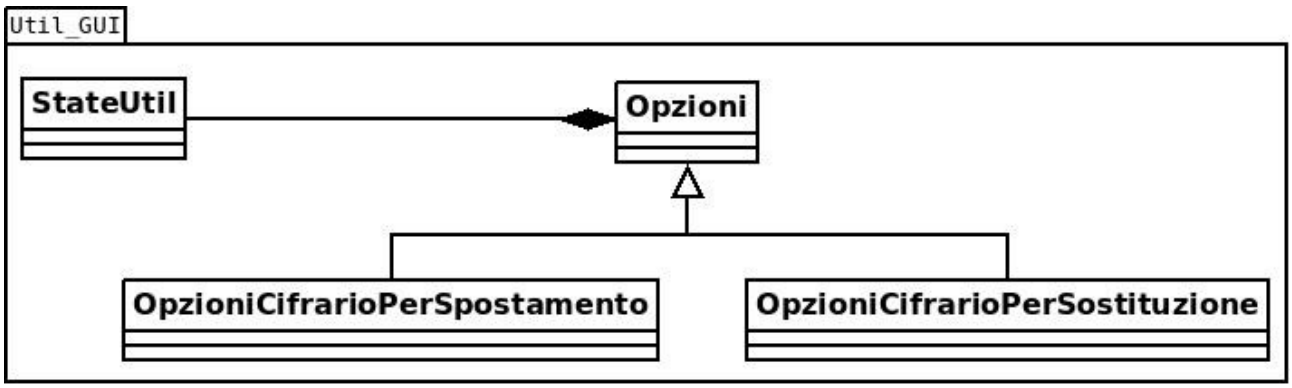
La classe OpzioniCIfrarioPerSostituzione

La classe “OpzioniCifrarioPerSostituzione” estende la classe “Opzioni” e realizza le opzioni particolari della corrispondente classe “CifrarioPerSostituzione”.

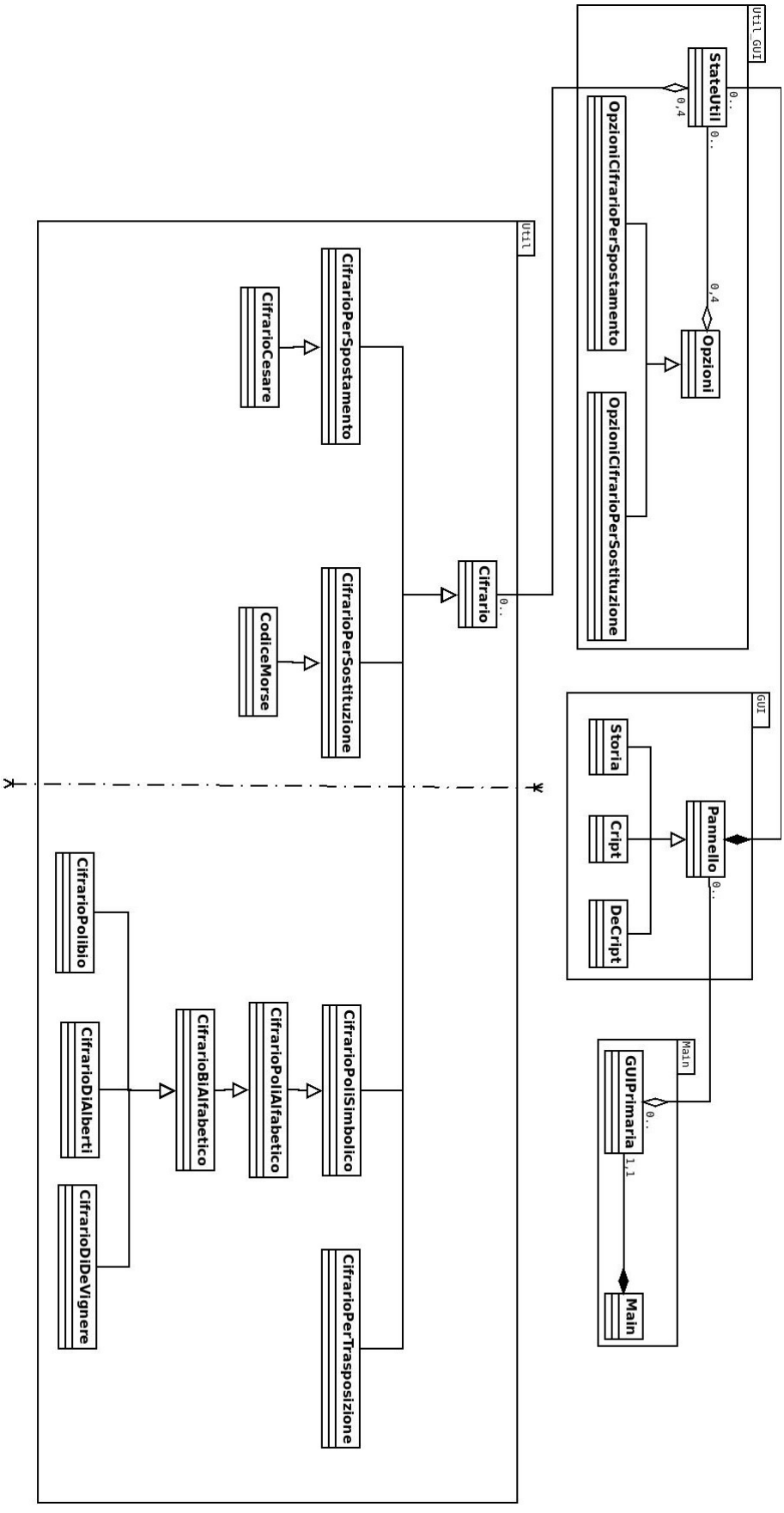
La classe StateUtil

La classe “StateUtil” è uno strumento per controllare assieme la classe “Cifrario” con la classe “Opzioni” corrispondente e ne gestisce il cambiamento di stato.

Schema totale del Package



Visione totale della progettazione

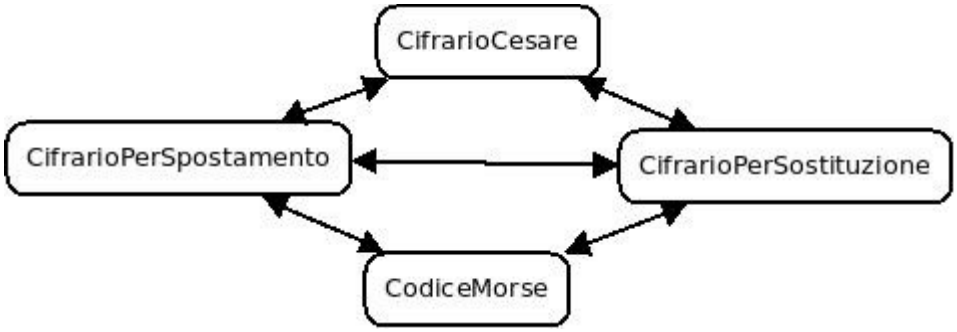


4. Progettazione: Comportamento

In questo capitolo sarà spiegato in parte il comportamento delle varie componenti software.

Generale

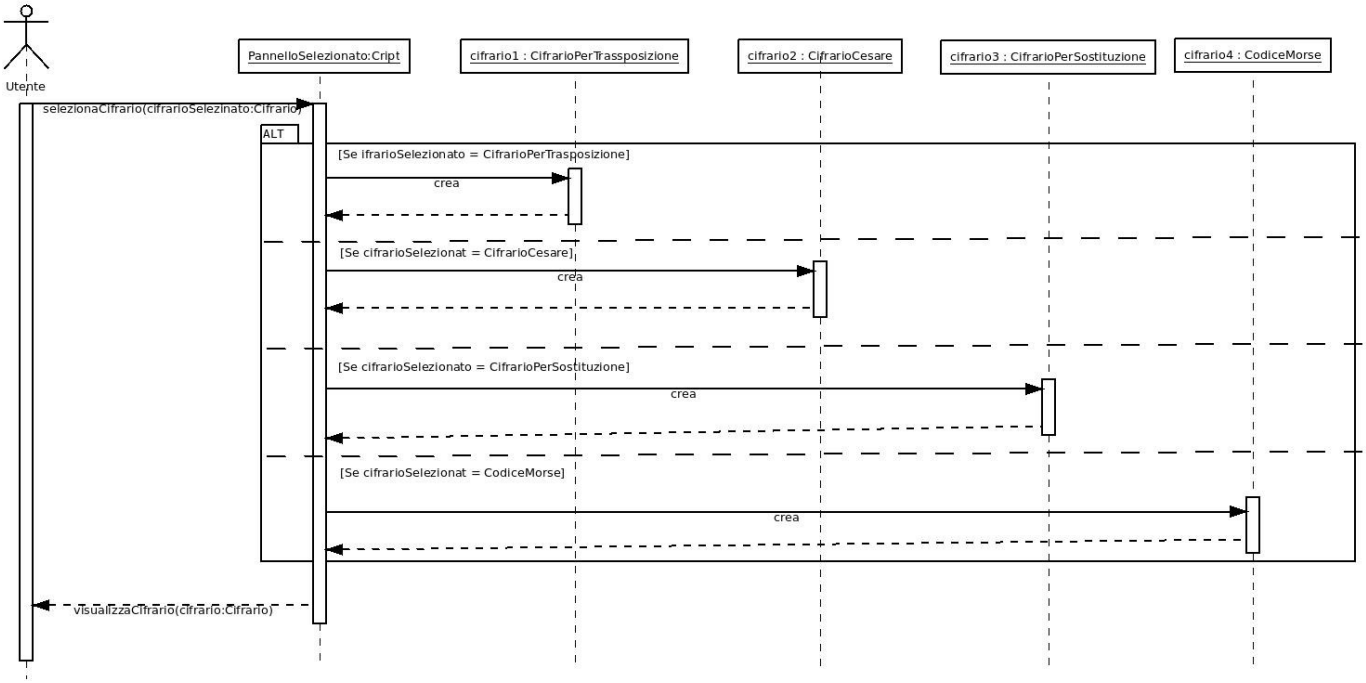
Il programma in generale deve essere in grado di permettere il cambiamento di metodo di criptazione e quindi di oggetto , cioè un cambiamento di stato a discrezione dell'utente secondo il seguente schema a seguito dell'azione di scelta dell'utente:



Svolgimenti classici

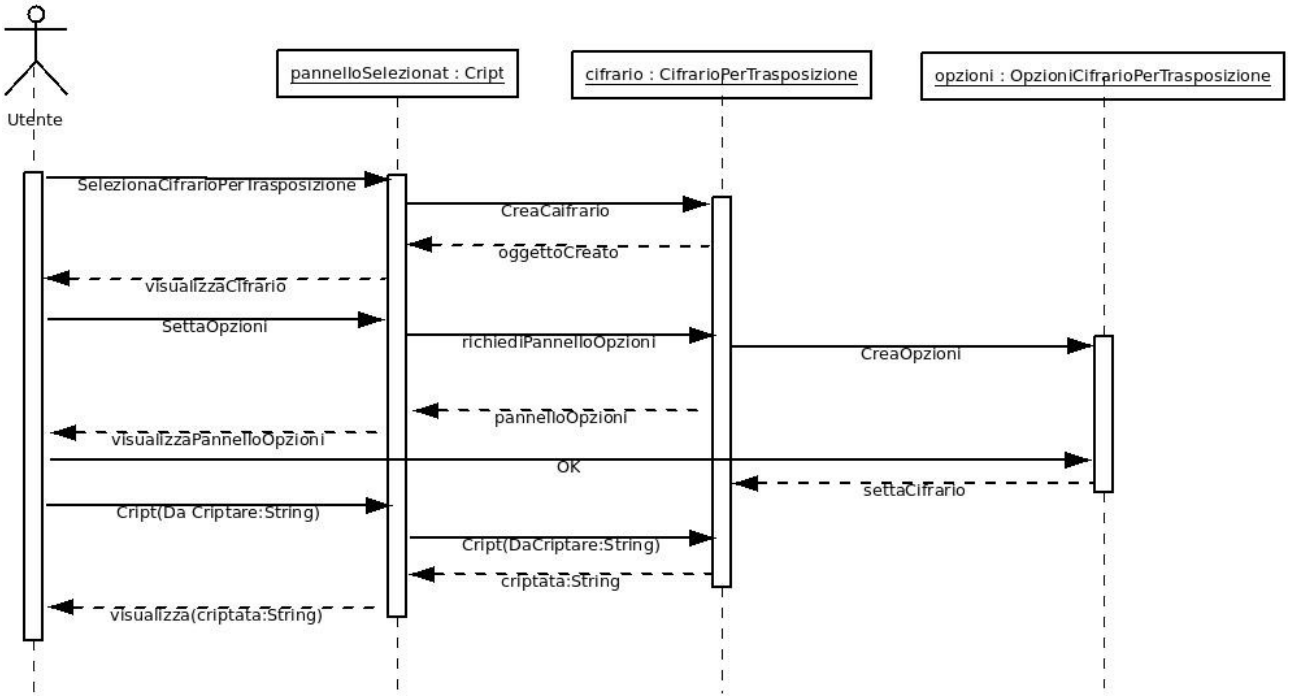
In questo capitolo si mostreranno degli svolgimenti normali delle seguenti componenti software durante l'interazione di un utente.

Cambia Cifrario



*Anche per il pannello deCript il funzionamento è analogo

Cripta e Decripta



*L'esempio del Cifrario per trasposizione è del tutto generico, analogamente avviene lo stesso procedimento per gli altri cifrari e per la decriptazione.

5. Progettazione: Implementazione

In questo capitolo si chiarisce come implementare nello specifico alcune scelte progettuali.

Vocabolario dei Pattern

È stato utilizzato il pattern Abstract Factory per l'implementazione dei metodi del package “Util” che forniscono gli oggetti che criptano e decriptano il testo, e anche per i metodi del package “GUI” come emerge dal class diagram .

Altre scelte progettuali

La classe “StateUtil” gestisce i tipi di cifrari e l'opzione corrispondente, evita che si creino e cancellino oggetti cifrario memorizzando i cifrari con le rispettive opzioni in apposite strutture dati in modo da utilizzare un pattern state.

Glossario dei Software di supporto

- Tutti i diagrammi sono stati realizzati con il software Dia
- L'applicazione è stata sviluppata sul sistema operativo Linux Mint 17.3 Rosa
- Per redigere le relazioni è stato utilizzato libre office
- L'applicazione verrà sviluppata in Java 1.8
- Per realizzare il logo è stato utilizzato il software Gimp