

passwordKalkulator

Progetto per l'esame di Programmazione ad oggetti
A.A. 2017-2018

Giovanni Peron
1137766

Sommario

Descrizione generale	3
Analisi delle gerarchie.....	3
Gerarchie del modello logico	3
Gerarchie della parte grafica	4
Analisi del codice polimorfo	5
Manuale utente dell'applicazione	6
Analisi delle ore di lavoro	8
Note	8
Note per la compilazione.....	8

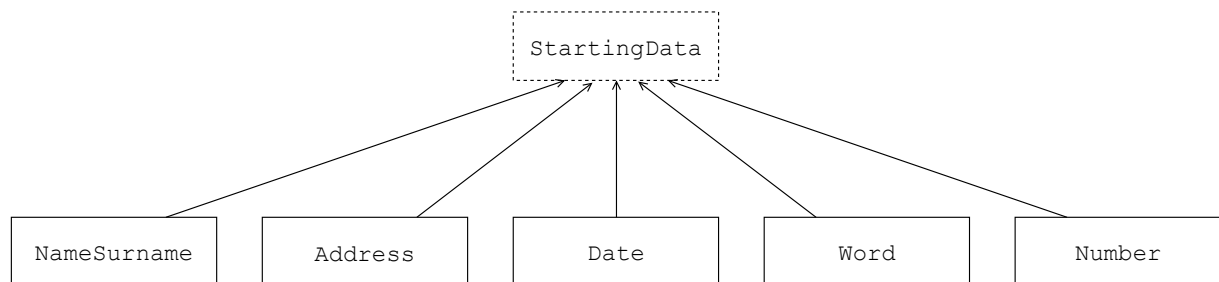
Descrizione generale

Il progetto consiste in un generatore di password. L'utente deve prima inserire dei dati, dopodiché può scegliere due tra i dati inseriti per calcolare la propria password, utilizzando tre diversi metodi di combinazione. Inoltre, prima di eseguire il calcolo è possibile scegliere il livello della password che si vuole ottenere, un livello classifica il tipo di password che si vuole ottenere in base ad una lunghezza minima e un insieme di caratteri che la password dovrà contenere.

Analisi delle gerarchie

Gerarchie del modello logico

Il progetto utilizza due gerarchie differenti la prima rappresenta i vari tipi di dati che l'utente inserisce per generare la password, la seconda rappresenta le password che possono essere calcolate.

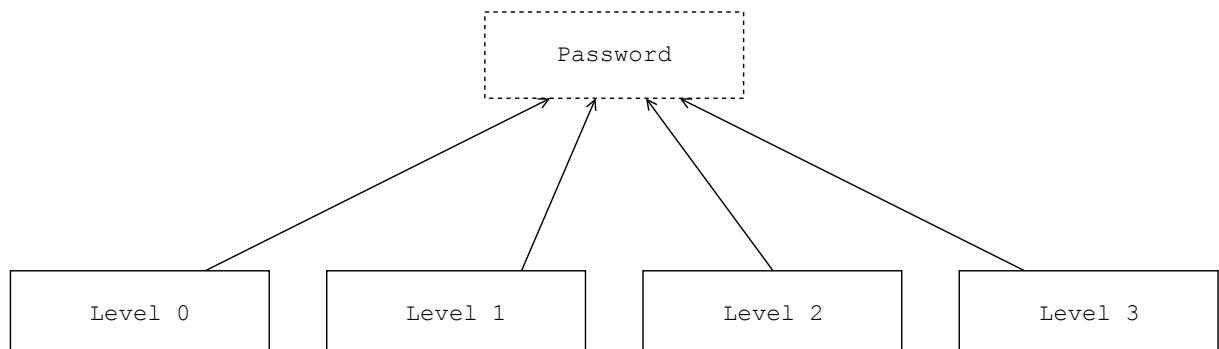


La classe base della prima gerarchia è `StartingData`. Essa è una classe che rappresenta tutti i tipi di dati che possono essere inseriti per generare una password. `StartingData` contiene un solo campo dati:

- `int id`, nel caso servisse identificare il tipo di dato rappresentato, 1=`NameSurname` 2=`Address` 3=`Date` 4=`Word` 5=`Number`.

`StartingData` è povera di campi dati in quanto è una classe astratta il cui compito principale è fornire una via per operare sui dati delle sue sottoclassi in modo polimorfo. Per questo motivo `StartingData` possiede tre metodi virtuali puri `getData`, `checkLength` e `validate`.

Le classi derivate da `StartingData` sono `NameSurname`, `Address`, `Date`, `Word` e `Number`. Esse rappresentano i tipi effettivi di dati inseribili, ognuna di esse possiede dei campi dati di tipo `QString` che memorizzano le informazioni inserite.



La seconda gerarchia della parte logica, come accennato, è dedicata alla rappresentazione delle password calcolabili. La classe base della gerarchia è `Password`, la quale possiede due campi dati:

- `QString pwd`; effettiva password calcolata
- `int len`; la lunghezza della password calcolata

Da Password vengono derivate le sottoclassi Level0, Level1, Level2 e Level3. Esse definiscono i livelli messi a disposizione per classificare le password. Tutte e quattro contengono i seguenti campi dati:

- QString description; la descrizione del livello
- const int lenMin; specifica la lunghezza minima che le password appartenenti al livello devono rispettare

Le password calcolate da passwordGenerator possono quindi essere:

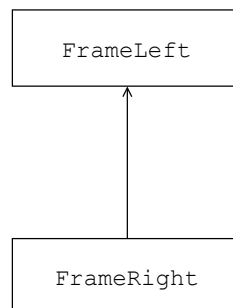
- di livello 0 contenenti solo lettere minuscole o solo numeri e con una lunghezza minima pari a 4 caratteri;
- di livello 1 contengono lettere minuscole e numeri; devono avere una lunghezza minima pari a 6 caratteri;
- di livello 2 formate da lettere minuscole, maiuscole e numeri; possiedono una lunghezza minima pari a 8 caratteri;
- di livello 3 contenenti lettere minuscole, maiuscole, numeri e alcuni caratteri speciali; aventi una lunghezza minima pari a 10 caratteri.

Le operazioni corrispondono a tre metodi: combination1, combination2, e combination3. Questi combinano i dati specificati dall'utente in modo casuale per generare una password. Questi metodi variano a seconda delle specifiche dettate dal livello scelto per la password, ad esempio:

Se l'utente inserisce parola: rosso e data: 23/07/1990, imposta il livello della password a 1 e digita *Parola* [tasto_Genera1] *Data* [tasto_=] otterrà una password del tipo oss719. Se il livello fosse stato 2 sSO199osS.

Gerarchie della parte grafica

La parte grafica possiede una gerarchia nella quale la classe base FrameLeft rappresenta la parte sinistra della finestra di utilizzo del programma. Mentre la classe derivata FrameRight rappresenta la parte destra.



La finestra principale del programma viene creata istanziando un oggetto della classe Grafica. Nel costruttore di grafica vengono assemblati i vari gruppi che compongono la finestra, in questo caso il gruppo di destra e quello di sinistra. Nel momento in cui viene istanziato l'oggetto FrameRight, per inizializzare tutte le componenti della parte destra, viene richiamato il costruttore di FrameLeft che inizializza anche la parte sinistra della finestra. Una volta ottenuto l'oggetto di tipo FrameRight viene utilizzato per invocare i metodi createLeftGroup (appartenente a FrameLeft) e createRightGroup (metodo proprio di FrameRight). Questi che restituiscono rispettivamente un puntatore al layout del gruppo sinistra e un puntatore al layout del gruppo di destra, che verranno utilizzati nel costruttore di Grafica per assemblare il layout della finestra principale.

Analisi del codice polimorfo

StartingData possiede tre metodi virtuali puri:

```
virtual QString getData() const=0;
virtual bool checkLength()=0;//non const modifico in Number e in Address
virtual void validate()=0;
```

Il metodo `getData` restituisce una stringa che rappresenta i dati inseriti, viene utilizzato nei metodi combination della gerarchia Password.

`checkLength` controlla che i dati inseriti siano abbastanza lunghi per il calcolo di una password, (potrebbe essere un metodo costante ma, nelle classi Address e Number, vengono aggiunte delle cifre di riempimento nel caso un numero non rispetti la lunghezza minima). Il metodo `checkLength` viene utilizzato nello slot `clickedBEqual` della classe `FrameRight`. Ovvero nel momento in cui viene premuto il pulsante uguale, prima di invocare uno dei metodi combination per calcolare la password, viene eseguito un controllo sulla lunghezza di `StartingData *d1` e `StartingData *d2` che sono i parametri da passare a combination.

Il metodo `validate` ha il dovere di eliminare gli eventuali spazi dalle stringhe in cui sono salvati i dati e di controllare che i caratteri inseriti siano coerenti con il tipo di dato effettivo, ad esempio nome e cognome di una persona non possono contenere dei numeri. Il metodo `validate` viene utilizzato nel metodo `saveData` della classe `FrameLeft`. `validate` non viene utilizzato in modo polimorfo perché, nella parte della GUI, `FrameLeft` possiede tra i suoi campi dati un puntatore per ogni sottoclasse della gerarchia `StartingData`; per cui l'invocazione avviene già con l'oggetto appropriato senza il bisogno di ricorrere al polimorfismo. È stato scelto comunque di mantenere `validate` virtuale nel caso si abbia il bisogno di usare il polimorfismo.

I metodi virtuali nella gerarchia Password sono:

```
virtual QString combination1(const startingData &d1, const startingData &d2)=0;
virtual QString combination2(const startingData &d1, const startingData &d2)=0;
virtual QString combination3(const startingData &d1, const startingData &d2)=0;
virtual QString getDescription() const=0;
```

`combination1`, `combination2`, `combination3` sono i metodi che ricevono i dati forniti dall'utente e generano la password a partire da essi, rispettando le restrizioni del livello in cui sono definiti. Vengono utilizzati nello slot corrispondente al click del pulsante uguale, `clickedBEqual` nella classe `FrameRight`; in base all'operazione e al livello scelti viene invocato il giusto metodo combination, che ritornerà il valore della password calcolata da mostrare sul display.

Il metodo `getDescription` ritorna la descrizione del livello a cui appartiene l'oggetto d'invocazione. Viene utilizzato nella classe `FrameRight` quando viene allocato il puntatore `QLabel *levelDesc` che mostra la descrizione del livello scelto.

Manuale utente dell'applicazione

All'apertura dell'applicazione salta all'occhio la divisione tra la parte sinistra per l'inserimento dei dati e la parte destra per la generazione della password.

The screenshot shows the 'passwordKalkulator' application window. It is divided into two main sections: 'Inserimento dati' (Data Entry) on the left and 'Generazione password' (Password Generation) on the right. The 'Inserimento dati' section contains input fields for 'Nome' (Name), 'Cognome' (Surname), 'Indirizzo' (Address), 'Via:' (Street), 'N.Civico:' (House Number), 'Comune:' (Municipality), 'Data' (Date), 'Parola' (Word), and 'Numero' (Number). The 'Generazione password' section features a large green box for the generated password, a dropdown menu to 'Scegli il livello della password che vuoi ottenere...' (Choose the password level you want to obtain...), and a grid of buttons for generating passwords based on different criteria: 'Nome Cognome', 'Indirizzo', 'Parola', 'Data', 'Numero', and 'C'. A 'Genera' button is also present for each criterion. A note explains that a password of level 0 contains only lowercase letters or numbers and has a minimum length of 4 characters.

La prima cosa da fare è inserire dei dati, non è necessario compilare tutti i campi, ma per procedere al calcolo della password bisogna inserire almeno due tipi di dato. Le informazioni inserite devono essere valide, ovvero:

- tutti i campi devono avere una lunghezza almeno pari a tre (esclusi quelli di tipo numerico: numero, n. civico e data).
- le informazioni inserite devono essere coerenti, per esempio un nome non può contenere valori numerici o caratteri speciali, un numero non può contenere lettere o caratteri speciali.

Qualora ci fosse qualcosa di errato comparirà una segnalazione di errore:

The two screenshots show the application interface with error messages. The left screenshot shows the 'Nome' field containing 'giova' and the 'Cognome' field containing 'st'. The generated password is 'Nome Cognome + Nome Cognome'. The right screenshot shows the 'Nome' field containing 'gio4va' and the 'Cognome' field containing 'storti'. The generated password is 'Nome Cognome'. A red error message at the bottom of the right screenshot states: 'Nome non valido, sono consentite solo lettere' (Name is not valid, only letters are allowed).

Dopo aver inserito delle informazioni valide l'utente deve scegliere il livello della password desiderata, per capire quale possa essere quello più adeguato può affidarsi alla descrizione.

Quando l'utente è certo del livello scelto può procedere a digitare sul tastierino le operazioni per calcolare la password, queste sono del tipo:

dato1 [tasto_operazione] *dato2* [tasto_=].

Ad esempio, è possibile generare una password di livello 2 utilizzando i campi dati *Parola* e *Numero* con l'operazione Genera 1 in questo modo:

Parola [tasto_Genera1] *Numero* [tasto_=].

Con *Parola* settata a “rosso” e *Numero* settato a “15” una possibile password risultante è rOs015Oss.

The image shows two screenshots of the 'passwordKalkulator' application. Both screenshots show the 'Inserimento dati' (Data Entry) section on the left and the 'Generazione password' (Password Generation) section on the right. In the 'Inserimento dati' section, the following values are entered: Nome: Pietro, Cognome: Dalbon, Indirizzo: B. Mogno, Via: B. Mogno, N.Civico: 7, Comune: Camposampiero, Data: 25/06/97, Parola: rosso, and Numero: 15. In the 'Generazione password' section, 'Livello 2' is selected. The application displays a description: 'Una password di livello 2 contiene lettere minuscole, maiuscole e numeri; possiede una lunghezza minima pari a 8 caratteri'. The 'Genera 1' button is pressed, resulting in the password 'rOs015Oss' displayed in a green box.

Nel caso l'utente sbagliasse la digitazione di qualche parametro durante l'utilizzo può premere il pulsante Canc del tastierino [tasto_C], per resettare lo stato del calcolo e poter rieseguire la digitazione.

Di seguito riporto altri due esempi di generazione di una password.

Esempio generazione di una password di livello 1 utilizzando i dati *Nome Cognome* e *Data* con l'operazione Genera 2, digitazione effettuata:

Nome Cognome [tasto_Genera2] *Data* [tasto_=].

Con *Nome Cognome* settato a “Pietro Dalbon” e *Data* settata a “25/06/1997” il risultato è i5e6t2.

The image shows two screenshots of the 'passwordKalkulator' application. Both screenshots show the 'Inserimento dati' (Data Entry) section on the left and the 'Generazione password' (Password Generation) section on the right. In the 'Inserimento dati' section, the following values are entered: Nome: Pietro, Cognome: Dalbon, Indirizzo: B. Mogno, Via: B. Mogno, N.Civico: 7, Comune: Camposampiero, Data: 25/06/97, Parola: rosso, and Numero: 15. In the 'Generazione password' section, 'Livello 1' is selected. The application displays a description: 'Una password di livello 1 contiene lettere minuscole e numeri; possiede una lunghezza minima pari a 6 caratteri'. The 'Genera 2' button is pressed, resulting in the password 'i5e6t2' displayed in a green box.

Esempio generazione di una password di livello 3 utilizzando i dati *Nome Cognome* e *Numero* con l'operazione Genera 3, digitazione effettuata:

Nome Cognome [tasto_Genera3] *Numero* [tasto_=].

Con *Nome Cognome* settato a “Pietro Dalbon” e *Data* settata a “15” il risultato è b-5oi1!NE0t.



Analisi delle ore richieste

Fasi progettuali	Ore richieste
Analisi preliminare del problema	5 ore
Progettazione modello	3 ore
Progettazione GUI	2 ore
Apprendimento libreria Qt	5 ore
Codifica modello	9 ore
Codifica GUI	18 ore
Debugging	4 ore
Testing	3 ore
TOTALE	49 ore

Note

Il progetto è stato sviluppato in un calcolatore con le seguenti specifiche:

- OS: Windows 10 Home (versione 1803 – build 17134.48);
- Informazioni su Qt: QtCreator 4.5.0 based on Qt 5.10.0 (MSVC 2015, 32 bit);
- Compilatore Java: Java 9.0.1.

Note per la compilazione

Per compilare il progetto si deve utilizzare il file passwordKalkulator.pro, come riportato nei seguenti comandi:

```
qmake passwordKalkulator.pro
make
```