

Programmazione ad oggetti  
2016-2017

# Progetto JP-Dict

di  
Francesca Lonedo  
1125508

## Scopo del progetto

JP-Dict è un'applicazione di dizionario pensata per scuole di lingua giapponese che mette a disposizione sia funzionalità di gestione degli utenti che la utilizzano, sia funzionalità di ricerca e modifica del database delle parole.

Essendo pensata per le scuole, in JP-Dict è presente una gerarchia di utenti, caratterizzata da diversi livelli di accesso alle funzionalità e alla visualizzazione dei contenuti. Sono quindi presenti amministratori, che hanno un accesso completo all'applicazione, insegnanti, a cui è permesso tutto tranne modificare gli altri utenti e studenti, a cui è permesso solo cercare all'interno del database e i cui risultati saranno filtrati a seconda del loro livello di conoscenza della lingua giapponese<sup>1</sup>.

Il dizionario mette a disposizione opzioni di ricerca sia nel database delle parole sia del database di *kanji*<sup>2</sup>. Essendo *kanji* e parole due entità intrinsecamente differenti, è stato necessario implementarli attraverso classi distinte e utilizzare due differenti contenitori per immagazzinarli.

## Breve introduzione alla lingua e al sistema di scrittura giapponese<sup>3</sup>

La lingua giapponese è una lingua agglutinante, il che significa che alla radice della parola, contenente il significato, vengono via via affissi morfemi per aggiungere informazione (tempo, negazione, indicatore di cortesia ecc.). Questa caratteristica, congiunta alla massiccia presenza di parole omofone, ha determinato lo sviluppo di un complesso sistema di scrittura che utilizza congiuntamente quattro alfabeti differenti:

- **Hiragana** (ひらがな). E' un alfabeto sillabico composto da 48 simboli di base e 25 derivati. E' utilizzato per scrivere la parte flessiva delle parole, per scrivere foneticamente parole difficili da scrivere in *kanji* (talvolta come *furigana*, scrittura in *hiragana* posta sopra ai *kanji* per indicarne la lettura) o per scrivere parole che non hanno una scrittura in *kanji* come particelle, alcuni avverbi ecc.
- **Katakana** (カタカナ). E' un alfabeto sillabico parallelo a *hiragana*, con cui condivide i fonemi rappresentati. Viene utilizzato per scrivere onomatopee, parole di origine straniera ormai assimilate dalla lingua giapponese e per traslitterare parole straniere come toponimi, nomi propri ecc.
- **Rōmaji**. E' l'alfabeto latino, utilizzato per scrivere parole straniere e sigle. E' possibile inoltre utilizzare il *rōmaji* per renderle leggibili parole giapponesi a chi ancora non conosce i *kana*<sup>45</sup>.
- **Kanji** (漢字). Sono i logogrammi di origine cinese che singolarmente o combinati tra loro vengono utilizzati per scrivere parole giapponesi. Sono caratterizzati da due differenti tipologie di lettura: *on'yomi* (音読み) è l'insieme delle letture sinogiapponesi, ossia basate sull'antica pronuncia cinese, mentre *kun'yomi* (訓読み) sono le letture giapponesi del concetto rappresentato dal carattere. A seconda del contesto un *kanji* assume una determinata lettura. I *kanji* possono essere catalogati a seconda del numero di tratti che li compongono o per il loro radicale, ossia l'insieme di tratti che forma un sottocarattere che ne conferisce l'area semantica<sup>6</sup>.

---

1 Il JLPT (Japanese Language Proficiency Test – 能力試験) è diviso in cinque livelli: N5-N4 compongono il livello base di apprendimento (初級), N3-N2 il livello intermedio (中級) e N1 è il livello avanzato (上級).

2 I *kanji* (漢字) sono i caratteri di origine cinese utilizzati nella scrittura giapponese.

3 Nel 2015 Francesca Lonedo ha conseguito la laurea in Lingue, culture e società dell'Asia e dell'Africa mediterranea (curriculum giapponese) presso l'Università Ca' Foscari di Venezia.

4 Con *kana* si indica prevalentemente la scrittura in *hiragana* e *katakana*.

5 Il sistema di traslitterazione utilizzato in questa relazione e da JP-Dic è lo *Hepburn*: *ch* si legge come la c in Cina, *g* si legge come g in gatto, *j* si legge come g in gioco, *sh* si legge come sc in sci, *w* si legge come u in uva, *y* si legge come i in ieri. Le vocali sono pronunciate come in italiano, *ō* rappresenta una o lunga, *ū* rappresenta una u lunga.

6 Alcuni radicali sono “uomo”, “risaia”, “acqua” ecc.

Come già accennato, in una singola frase possono coesistere tutti e quattro i sistemi di scrittura descritti:

「田中さんはANAでローマへ行きました。」  
“Tanaka san wa ANA de Roma e ikimashita”.  
“Il signor Tanaka è andato a Roma con la All Nippon Airways.”<sup>7</sup>

田中 è la scrittura in *kanji* di “Tanaka”, un nome proprio e さん è un suffisso onorifico di rispetto scritto in *hiragana*.

は è una particella, scritta in *hiragana*, e serve a marcare il soggetto.

ANA è la compagnia aerea di bandiera giapponese, All Nippon Airways.

で è una particella, scritta in *hiragana*, e in questo caso indica il mezzo.

ローマ è la traslitterazione in *katakana* di “Roma”.

へ è una particella, scritta in *hiragana*, e indica il moto a luogo.

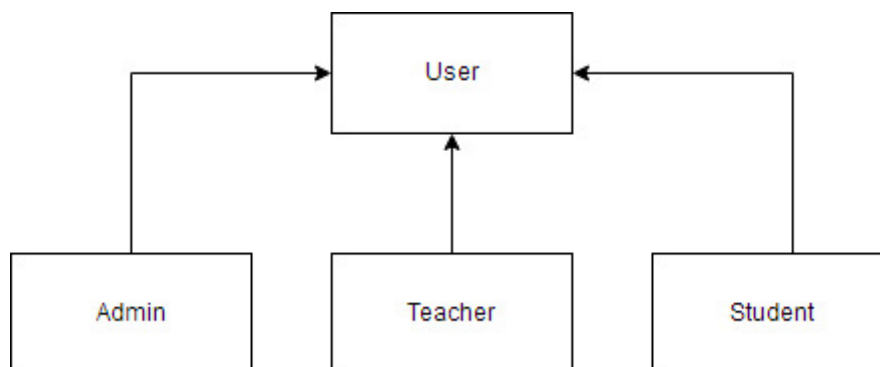
行きました è il verbo andare, 行 è la radice in *kanji* del verbo che ne indica il significato, きました è la coniugazione del verbo in *hiragana*, in questo caso il passato della forma cortese.

Per quanto riguarda la morfologia della lingua, ci sono quattro principali elementi degni di nota<sup>8</sup>:

- Verbi (*dōshi* 動詞). Possono essere transitivi (*tadōshi* 他動詞) o intransitivi (*jidōshi* 自動詞) e si differenziano a seconda della coniugazione: *ichidan* (一段), *godan* (五段) o irregolare<sup>9</sup>.
- Aggettivi (*keiyōshi* 形容詞). In giapponese vengono coniugati similmente ai verbi e sono suddivisi in due categorie a seconda di come questo avviene: *ikeiyōshi* (い形容詞) e *nakeiyōshi* (な形容詞).
- Sostantivi (*meishi* 名詞). Non hanno né genere né numero. Alcuni sono di origine straniera.
- Avverbi (*fukushi* 副詞).

## Gerarchie di tipo usate

### Utenti



La gerarchia di utenti è composta da quattro classi:

- User è una classe astratta (per via della presenza del metodo virtuale puro di clonazione) dalla quale ereditano le altre classi concrete. Mette a disposizione vari metodi, eventualmente virtuali, per il sotto oggetto tra cui costruttore, operatori di uguaglianza e disuguaglianza, setters e getters, e metodi per esportare su file xml i dati degli oggetti.

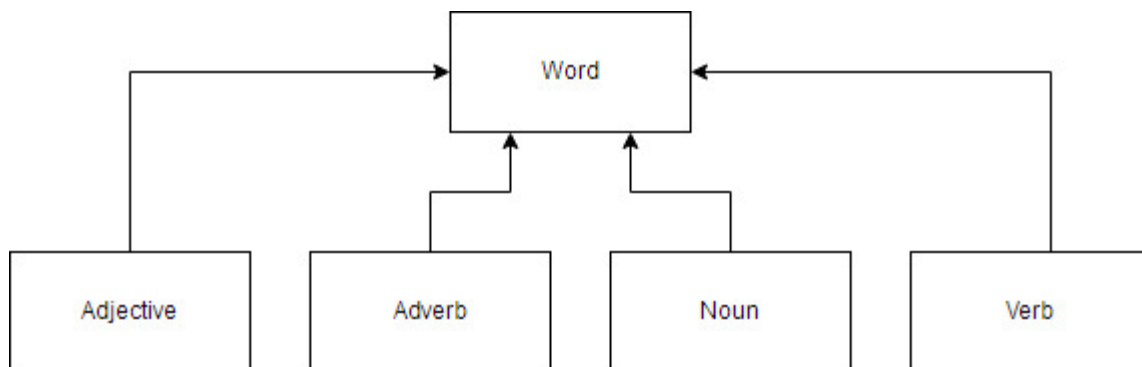
<sup>7</sup> Esempio tratto da “La lingua Giapponese” di Marcella Mariotti, Carocci Editore, Roma, 2014.

<sup>8</sup> Nella lingua giapponese sono presenti anche pronomi e particelle, ma per semplicità JP-Dict li cataloga in associazione con gli avverbi in quanto modificatori di altri elementi della frase.

<sup>9</sup> Esistono solo due verbi irregolari, “fare” する e “venire” 来る.

- Admin è una classe concreta che eredita da user e rappresenta gli amministratori del sistema. Reimplementa i metodi virtuali ereditati da user per adattarli alle proprie necessità, ma non aggiunge nuovi campi dati rispetto alla classe base. Gli utenti appartenenti a questa classe hanno pieno accesso alle funzionalità dell'applicazione: possono aggiungere, eliminare e modificare parole, *kanji* e utenti e possono visualizzare tutto il contenuto del dizionario.
- Teacher è una classe concreta che eredita da user e rappresenta gli insegnanti. Così come la classe admin, la classe user non aggiunge campi dati a quanto ereditato ma si limita a reimplementare i metodi virtuali. L'accesso all'applicazione degli utenti della classe teacher è però leggermente limitata rispetto gli utenti di classe admin: gli insegnanti possono aggiungere, eliminare e modificare parole e *kanji*, visualizzando l'intero contenuto del dizionario, ma non possono eseguire operazioni sugli utenti, dei quali visualizzeranno solo le informazioni di base come la classe di appartenenza, lo username, e se si tratta di studenti del livello del JLPT raggiunto.
- Student è una classe concreta che eredita da user e rappresenta gli studenti. Aggiunge il campo dati level per tenere traccia del livello JLPT raggiunto e reimplementa i metodi virtuali ereditati per le proprie esigenze. Aggiunge inoltre setter e getter del nuovo campo dati. Gli utenti della classe student hanno un accesso limitato all'applicazione; possono infatti unicamente utilizzare le funzionalità di ricerca nel dizionario, i cui risultati verranno filtrati a seconda del livello JLPT raggiunto: uno studente potrà visualizzare unicamente parole e *kanji* di livello pari o inferiore al proprio. Gli studenti non possono modificare il contenuto del dizionario e non possono effettuare ricerche o modifiche riguardanti gli altri utenti. Come gli altri utenti, possono modificare il proprio username e password; non possono però modificare il proprio livello del JLPT.

## Parole



La gerarchia di parole è composta da quattro classi:

- Word è una classe base astratta (grazie al metodo virtuale puro di clonazione) dalla quale ereditano le classi concrete della gerarchia. Mette a disposizione vari metodi virtuali e non per il sotto oggetto, tra cui costruttore, setters e getters, operatori di uguaglianza e disuguaglianza e metodi per l'esportazione dei dati di un oggetto in un file xml.
- Adjective è una classe concreta che eredita da word e rappresenta un aggettivo. Aggiunge un campo dati booleano a tenere traccia del tipo di aggettivo (*nakeiyōshi* o *ikeiyōshi* rappresentati da un unico campo booleano in quanto uno esclusivo dell'altro) e reimplementa i metodi ereditati per le proprie necessità. Aggiunge inoltre il getter e il setter per il nuovo campo dati.
- Adverb è una classe concreta che eredita da word e rappresenta un avverbio. Questa classe è utilizzata per memorizzare nel dizionario anche altri elementi morfologici come le particelle, e per questo viene aggiunto il nuovo campo dati booleano particle, con relativi getter e setter. La classe inoltre reimplementa quanto ereditato per le proprie esigenze.

- **Noun** è una classe concreta che eredita da **word** e rappresenta un sostantivo. Aggiunge un campo dati booleano per tenere traccia se la parola sia di origine straniera, ne implementa **setter** e **getter** e reimplementa quanto ereditato.
- **Verb** è una classe concreta che eredita da **word** e rappresenta un verbo. La forma del verbo rappresentata è ovviamente la forma del dizionario o piana (*jishokei* 辞書系), e i due campi dati booleani aggiunti tengono nota della coniugazione del verbo<sup>10</sup> (*ichidan* 一段 o *godan* 五段) e della transitività. Sono resi disponibili i **setter** e **getter** necessari e viene reimplementato quanto ereditato per le nuove esigenze.

## Utilizzo di codice polimorfo

### Gerarchia utenti

La classe base **user** mette a disposizione un distruttore virtuale, in modo da richiamare il distruttore corretto quando necessario, e un metodo di clonazione virtuale puro. Sono inoltre implementati i seguenti metodi virtuali:

- `void exportUserData(QXmlStreamWriter &) const;` si occupa di esportare i dati relativi ad un oggetto in formato xml. La classe **student** lo reimplementa in modo da adattarlo alla presenza di un nuovo campo dati mentre le classi **admin** e **teacher** nella loro reimplementazione chiamano il metodo della classe base. Viene utilizzato dal metodo `bool fileExport();` della classe **userContainer** (`usercontainer.cpp`) per esportare in xml i dati degli utenti.
- `StdString getInfo() const;` crea una stringa con le informazioni relative all'utente come classe di appartenenza, username ecc. Viene reimplementata dalle classi che lo ereditano. Viene utilizzato dal metodo `void showUser(user*);` della classe **userSearchPanel** (`usersearchpanel.cpp`) per inizializzare il campo dati degli oggetti **QListWidgetItem** da visualizzare nella **QListWidget** con le informazioni relative agli utenti.

### Gerarchia parole

Così come la gerarchia utenti, la gerarchia di parole mette a disposizione un distruttore virtuale e un metodo di clonazione virtuale puro. Sono inoltre virtualizzati i seguenti metodi:

- `void exportWordData(QXmlStreamWriter &) const;` si occupa di esportare in formato xml i dati relativi ad una parola e viene reimplementato da ogni classe che lo eredita per adattarlo ai campi dati aggiunti. Viene utilizzato dal metodo `void fileExport();` della classe **dictionary** (`dictionary.cpp`) per esportare i dati delle parole nel file xml.
- `QString getInfo() const;` restituisce una stringa contenente le informazioni di base della parola come categoria morfologica, scritture ecc. Ogni classe che lo eredita lo reimplementa per le proprie esigenze. Viene utilizzato dal metodo `void search();` della classe **workpanel** (`workpanel.cpp`) per inizializzare il campo dati degli oggetti **QListWidgetItem** da visualizzare nella **QListWidget**.

## Manuale utente

JP-Dict utilizza tre file xml per salvare i dati utilizzati dall'applicazione: `userdata.xml` per memorizzare i dati relativi agli utenti, `kanjidata.xml` per i dati relativi ai *kanji* e `dictionarydata.xml` per i dati relativi alle parole del dizionario. Qualora non venga trovato un file `userdata.xml` (ad esempio al momento dell'inizializzazione dell'applicazione), viene automaticamente creato un nuovo file contenente un amministratore di default con username: `admin` e password: `admin`.

Dopo il login l'utente visualizza il pannello principale, che viene modificato dinamicamente a seconda del livello di accesso dello stesso.

---

<sup>10</sup> Per semplicità, JP-Dict cataloga come *godan* i due verbi non regolari della lingua Giapponese.

Per gli studenti è disponibile solo la funzione di ricerca di base: dopo aver selezionato se cercare tra le parole (in modalità giapponese-inglese o inglese giapponese) o tra i *kanji* (eventualmente cercando tra i radicali o tra le letture), le informazioni relative a quanto cercato vengono visualizzate in un pannello sulla sinistra mentre l'elenco completo dei risultati appare in una lista sulla destra; questa accumula la cronologia dei risultati ottenuti fino a quando l'utente non decida di "pulire" cliccando sul pulsante clean. A seguito della selezione di un elemento dalla lista dei risultati, questo viene visualizzato per intero nel pannello sulla sinistra. Per amministratori e insegnanti viene aggiunta la possibilità di filtrare i risultati della ricerca per livello del JLPT: se è attiva una selezione per keyword questa viene filtrata per livello mentre se si sta eseguendo una ricerca tra i *kanji* è possibile visualizzare l'elenco completo dei *kanji* di un livello del JLPT<sup>11</sup>. Vengono aggiunti inoltre i pulsanti edit e delete, che rispettivamente aprono il pannello di modifica delle parole/*kanji* dell'elemento selezionato o lo eliminano. Compaiono inoltre un pulsante per accedere al pannello per aggiungere parole, uno per aggiungere *kanji* e uno per accedere al pannello di amministrazione utenti. Questo è fondamentalmente un pannello di ricerca dove i risultati sono visualizzati in una lista. La ricerca può avvenire tramite inserimento di una keyword oppure è possibile richiedere un elenco completo di tutti gli insegnanti oppure di tutti gli studenti (filtrabili per livello del JLPT). Selezionare un utente dalla lista tramite doppio click ha effetti diversi a seconda che l'utente a usare l'applicazione sia un insegnante o un amministratore: il primo semplicemente visualizza i dati dell'utente in una nuova finestra mentre per il secondo è reso disponibile un pannello di modifica ed eliminazione.

## Ore impiegate per la realizzazione

Progettazione modello: 1h

Progettazione Gui: 4h

Codifica modello: 15h

Codifica Gui: 35h

Debug: 6h

Test: 4h

Il monte ore disponibile è stato sforato poiché per l'implementazione del dizionario è stato necessario l'utilizzo di due differenti contenitori, uno per le parole e uno per i *kanji*. Come spiegato nell'introduzione alla lingua, *kanji* e parole sono due entità intrinsecamente differenti e per tanto sarebbe stata una forzatura inserirle all'interno di una gerarchia che permettesse l'utilizzo di un unico contenitore per immagazzinarle.

Inoltre, essendo i *kanji* parte fondamentale dell'apprendimento della lingua, è stato ritenuto opportuno non escluderli dall'applicazione, nonostante il lavoro aggiuntivo da essi richiesto.

Sistema operativo di sviluppo: Windows Vista Home Premium Service Pack 2 (32 bit).

Versione Qt: Qt Creator 4.2.0 (4.2.0) Based on Qt 5.7.1 (MSVC 2015, 32 bit)

Compilatore: MinGW 5.3.0 32-bit2

---

<sup>11</sup> I livelli dei *kanji* sono standardizzati dalla fondazione che si occupa della certificazione linguistica, dunque è stata resa disponibile una funzionalità che permetta di visualizzarli per livello. Questo non è stato invece reso disponibile per le parole in quanto il livello di appartenenza non è ufficiale e potrebbe variare a discrezione degli istituti didattici.