

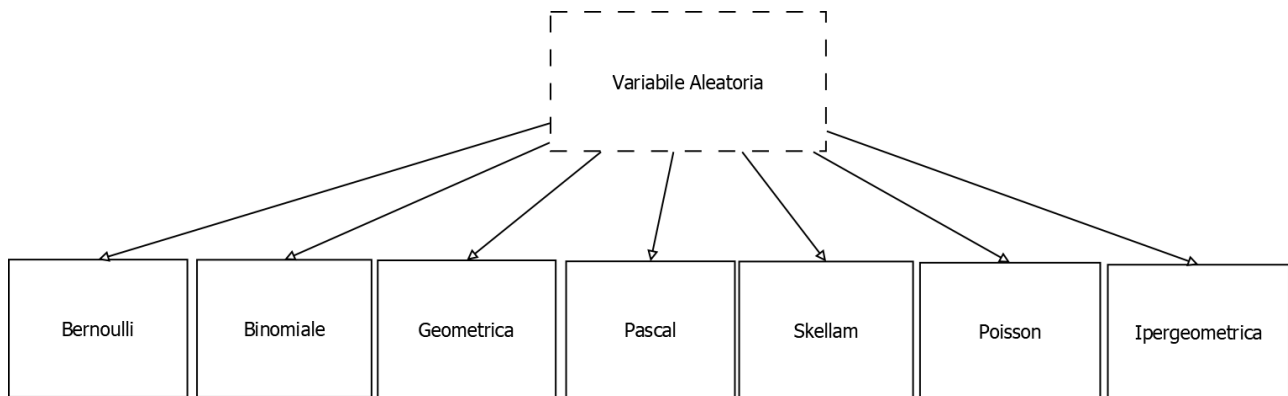
Relazione progetto Kalk

di Costantino Marco

Il tema della calcolatrice

La calcolatrice è a tema “probabilità”. Si occupa in particolare di implementare le funzioni di calcolo basilari delle variabili casuali (o aleatorie) discrete.

La gerarchia dei tipi



A monte della gerarchia troviamo una classe astratta “Variabile Aleatoria” (v.a.) che rappresenta il contratto che le classi che ereditano da v.a. dovranno rispettare.

In probabilità una variabile aleatoria è la descrizione dei risultati di un esperimento aleatorio in termini di probabilità. Diverse variabili aleatorie descrivono esperimenti diversi o danno informazioni diverse sullo stesso esperimento. Le principali operazioni effettuabili tra variabili aleatorie sono:

- Somma / Differenza / Prodotto / Quoziente

Che richiedono 2 operandi per essere svolte e

- Densità

Che richiede un parametro intero aggiuntivo e

- Media / Varianza / Moda

Che sono operazioni che necessitano di un solo operando.

Nella classe base VariabileAleatoria, sono dichiarati dei metodi per ogni una di queste operazioni, che poiché verranno implementate in maniera molto diversa in base al tipo di v.a., sono astratti. Alcuni di questi metodi però non sono disponibili in tutti i tipi di variabile aleatoria ad esempio: “Bernoulli” e “Geometrica” non implementano differenza, prodotto e quoziente oppure: “Skellam” non implementa moda, in tali casi quindi, verranno lanciate delle eccezioni. Questo per mantenere il codice del controller snello e generico.

Altre 2 operazioni, che compaiono in maniera meno consistente: “Informazione” e “Entropia” sono invece metodi propri dei vari tipi di v.a. che le implementano.

Una descrizione completa del tipo di variabile aleatoria, delle operazioni disponibili, la loro implementazione e i loro vincoli, dei parametri che le descrivono e dei vincoli di tali parametri, è presente sotto forma di commento nel file .h di ogni tipo di v.a..

Il controller

Kalk funziona come un calcolatrice scientifica portatile: un'espressione viene inserita, e una volta premuto il tasto uguale viene valutata e viene restituito il risultato dell'espressione o eventualmente un messaggio di errore. Poiché inserire direttamente le variabili aleatorie nell'espressione risulterebbe tedioso, complicherebbe la scrittura e la lettura delle espressioni e genererebbe un numero maggiore di possibili errori, nella parte alta della calcolatrice vi sono 2 box per l'input di 2 variabili aleatorie: X e Y. L'espressione poi sarà scritta in funzione di X ed Y. Una volta premuto il tasto uguale, il risultato o l'eventuale messaggio di errore sostituirà l'espressione. Il controller, implementa la valutazione di un'espressione in cui possono apparire le varie operazioni.

In particolare, rispetta l'ordine di precedenza delle operazioni e delle parentesi.

L'algoritmo che permette ciò è ricorsivo e funziona a grandi linee così:

Data un'espressione, l'algoritmo la valuta cercando l'operazione a priorità minima. Quando la trova, esegue tale operazione tra i risultati della valutazione dell'espressione precedente e successiva a quell'operazione.

Esempio:

$\text{Valuta}[(x+y)*y] = \text{Valuta}[x+y] * \text{Valuta}[y] = \text{Valuta}[x] + \text{Valuta}[y] * y = x+y*y$

Grazie allo stack di attivazione delle chiamate ricorsive, le operazioni vengono eseguite dalla più alta alla più bassa in priorità.

Il metodo ritorna un puntatore polimorfo a VariabileAleatoria e il caso base dell'algoritmo è X o Y.

In tale algoritmo avviene il principale uso del polimorfismo nella calcolatrice. La ricerca di errori nei parametri dell'espressione è risultato della classi che ereditano da VariabileAleatoria. Controller invece è responsabile di individuare errori nella sintassi dell'espressione.

Controller ha anche il compito di calcolare e ritornare il risultato di operazioni mono operando. Poiché queste non ritornano tutte lo stesso tipo il risultato è ritornato sotto forma di QString. Poiché tali operazioni non sono disponibili omogeneamente tra i tipi di v.a. è necessario effettuare RTTI per tali calcoli.

La gui

La gui presenta un combobox per la selezione del tipo di variabile aleatoria che si vuole utilizzare.

Ogni volta che viene selezionata un tipo di variabile aleatoria, i tasti per le operazioni non disponibili vengono disattivati. È ancora possibile scrivere da tastiera l'operazione non disponibile nell'input box, ma un errore verrà segnalato una volta premuto il tasto "=". Gli input box per X e Y accettano una descrizione parametrica delle variabili aleatorie. I parametri necessari e i vincoli su tali parametri sono descritti in ogni file .h dei vari tipi di v.a.. L'input box K serve a raccogliere un aggiuntivo parametro intero necessario per il calcolo della densità. L'inserimento di tale parametro non è necessario se non si calcola la densità. Poiché le operazioni +, -, *, / sono definite soltanto tra variabili aleatorie, non è possibile applicarle tra v.a. e numeri. Poiché le operazioni mono operando restituiscono numeri o coppie di numeri, è possibile richiamarle solamente come ultima operazione, applicata su una variabile aleatoria.

Esempio di espressione lecita: $E(x+y)$ (dove E è la media della v.a. risultante dalla somma di x e y)

Esempio di espressione illecita: $E(x)+y$ (poiché $E(x)$ è un numero e la somma tra numeri e v.a. non è definita, l'operazione non è lecita.)

Il tasto "C" non fa altro che cancellare ciò che è scritto nell'input box.

La seguente è una tabella che riassume i parametri necessari per descrivere le v.a.:

Bernoulli	$p \in \mathbb{R} \ p \in [0, 1]$
Binomiale	$n \in \mathbb{N} \ n > 0, \ p \in \mathbb{R} \ p \in [0, 1]$
Geometrica	$p \in \mathbb{R} \ p \in [0, 1]$
Pascal	$p \in \mathbb{R} \ p \in [0, 1], \ n \in \mathbb{N} \ n > 0$
Poisson	$\lambda \in \mathbb{R}, \ \lambda > 0$
Skellam	$\lambda_1, \lambda_2 \in \mathbb{R}, \ \lambda_1, \lambda_2 > 0$
Ipergeometrica	$n \in \mathbb{N} \ n > 0, \ h \in \mathbb{N} \ h \geq 0, \ r \in \mathbb{N} \ r > 0, \ n > h, \ n > r$

I parametri andranno semplicemente inseriti nell'input box nell'ordine corretto, separati da una virgola.

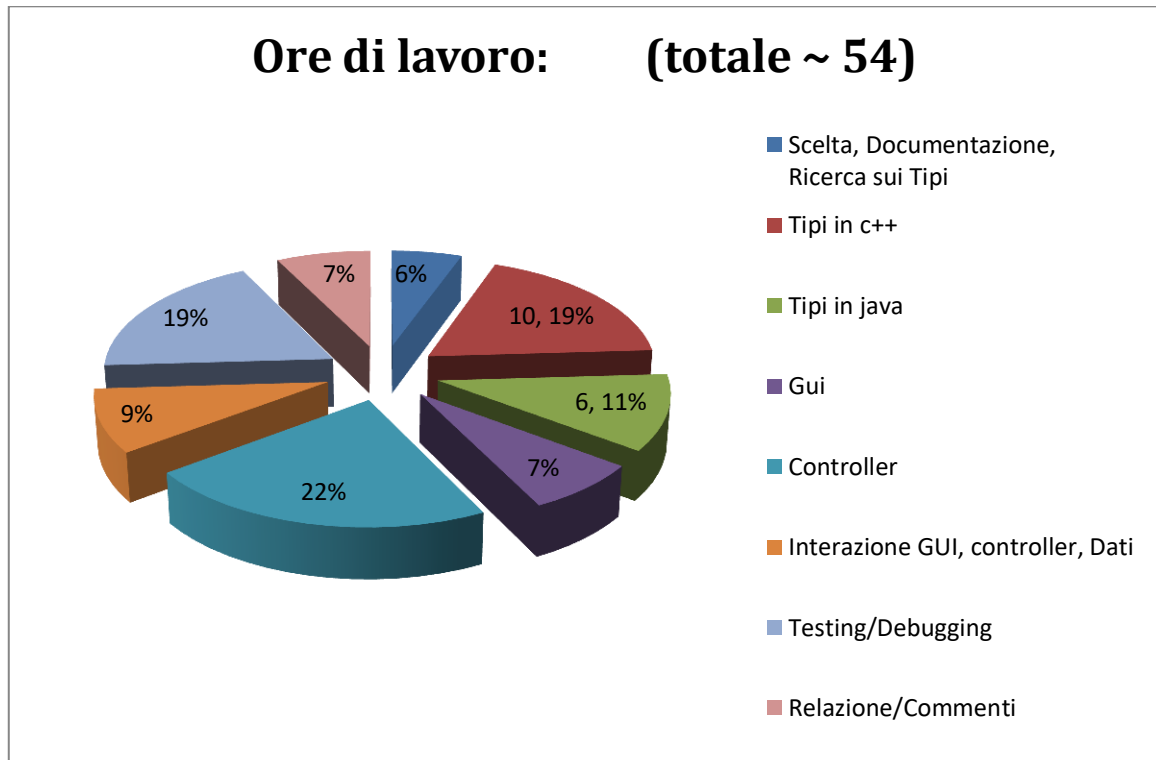
La rappresentazione di numeri con parte decimale fa uso del "." non della virgola.

Ambiente di sviluppo, .pro, ore di lavoro, considerazioni

Circa metà del progetto è stato sviluppato sulla macchina virtuale fornita, con QT 5.5.1

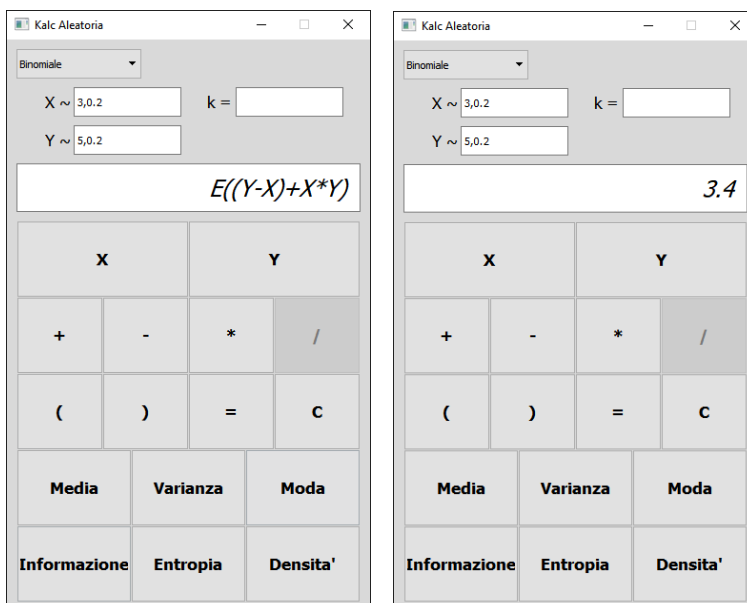
Per ragioni di prestazioni della macchina virtuale, il progetto è stato finito su Windows 10 sempre con QT 5.5.1.

Il progetto per essere compilato necessita del file **Kalk.pro** consegnato insieme al resto dei file del progetto.



I tipi di dato scelti non hanno riservato imprevisti particolari, mentre l'implementazione del controller si è rivelata non banale. Credo che l'esercizio dello scrivere un algoritmo che valuti un'espressione sia utile e molto interessante. È stato interessante inoltre notare, come nella realizzazione del progetto, separare controller, gui e dati fosse un'esigenza spontanea e quasi obbligata per avere un codice risultante chiaro.

Esempio di sessione della calcolatrice:



(screenshot raccolti su Windows 10)