

Sapy

Metodologie di programmazione

Package: it.uniroma1.sapy

Class: Sapy

Attraverso sapy , il punto di accesso del programma,viene interpretato il file .sapy attraverso l'apposito comando

Package: it.uniroma1.sapy.espressione

lo scopo è di fornire i mezzi per valutare le diverse espressioni

Class: Espressione

SuperClasse astratta dove risiedono i metodi per calcolare le espressioni: int_exp(), bool_cmp(), bool_exp()

Class: MemoriaVariabile

Classe nella quale sono contenute le variabili con i rispettivi valori

Class: Variabile

Classe che rappresenta le variabili e interpreta l'espressione assegnata alla variabile

Class: EspressioneBooleana

Classe che attraverso la funzione calcolaValore() ,calcola il valore dell'espressione

Class: EspressioneAritmetica

Classe che attraverso la funzione calcolaValore() ,calcola il valore dell'espressione

Class: MemoriaGoto

Classe nella quale sono contenute le istruzioni con le rispettive etichette utili al Goto

Exception: EspressioneNonValida

Se un espressione non è valida , non si riesce a calcolare, allora genera un errore : Espressione non valida

Package: it.uniroma1.sapy.lexer

Fornisce i mezzi per effettuare la trasformazione in una lista di token contiene le varie classi per il riconoscimento e la creazione dei Token ad es. TokenAnd .. etc.

Class: Lexer

Fornisce gli strumenti per la creazione della lista dei token e viene verificata l'esistenza del file

Class: Token

Superclasse astratta dei Token

Package: it.uniroma1.sapy.parsing

Fornisce gli funzionalità per l'interpretazione della sequenza e la sua successiva trasformazione in un ProgrammaEseguibile

Class: Parser

Attraverso una lista di Token viene creato un ProgrammaEseguibile. Vengono parsati tutti i token e create le varie istruzioni

Exception: TokenNonIntepretabile

Se non viene riconosciuto il token dal parser viene emessa un eccezione

Package: it.uniroma1.sapy.runtime

Dato in input un sorgente sotto forma di stringa viene richiamato il parser le istruzioni ottenuto vengono eseguite una ad una

Class: Interprete

Viene richiamato il parser e creato un oggetto ProgrammaEseguibile e viene richiamata l'apposita funzione esegui()

Class: ProgrammaEseguibile

Viene eseguita la lista di istruzione le quali sono autosufficienti nel giusto ordine stando attento all'esecuzione dei GOTO

Package: it.uniroma1.sapy.runtime.istruzioni

Contiene le varie Istruzioni

Class: Istruzione

Superclasse Astratta delle istruzioni

Class: IstruzioneAssegnazione

Classe implementazione della istruzioneAssegnazione attraverso esegui verrà aggiunta la variabile alla MemoriaVariabile

Class: IstruzioneFOR

Classe implementazione della istruzioneFOR attraverso esegui dove viene iterata la variabile del ciclo

Class: IstruzioneGOTO

Classe implementazione della istruzioneGOTO viene implementato il salto da un istruzione all'altra a seconda dell'etichetta specificata

Class: IstruzioneIF

Classe implementazione della istruzioneIF viene implementata la istruzione IF

Class: IstruzionePrint

Classe implementazione della istruzionePrint viene implementata l'istruzione Print

Class: IstruzioneInput

*Classe implementazione della istruzioneInput viene implementata
l'istruzione Input*

Class: IstruzioneEND

*Classe implementazione della istruzioneEND viene implementata
l'istruzione END*

Mario Corradi 1407477