

IIDE

Intelligent IDE (for Java Programming)

Progetto a cura di Andrea Vendrame – andrea.risk@gmail.com

L'idea di fondo

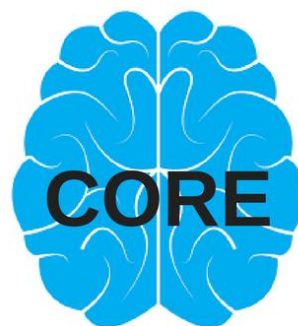
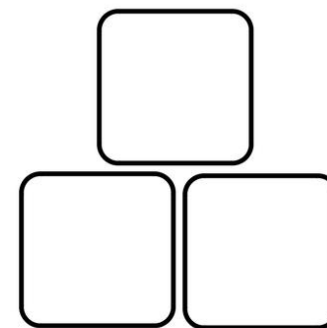
- Lo sviluppo di IIDE porterà a **programmare** in modo più semplice ed intuitivo dando anche la possibilità di implementare «al volo» l'usuale pseudocodice.
- La creazione di quello che può essere definito un ambiente di sviluppo non troppo conforme agli standard abituali si basa sull'idea di utilizzare il linguaggio italiano (nei limiti del possibile) per far eseguire a IIDE metodi i cui parametri sono dedotti dalla costruzione delle frasi utilizzate e dal contesto (simil IA).

Struttura del progetto

Database parole chiave



Moduli
integrati



Utente



Esempi di funzionamento

«Crea un file di nome foglio»

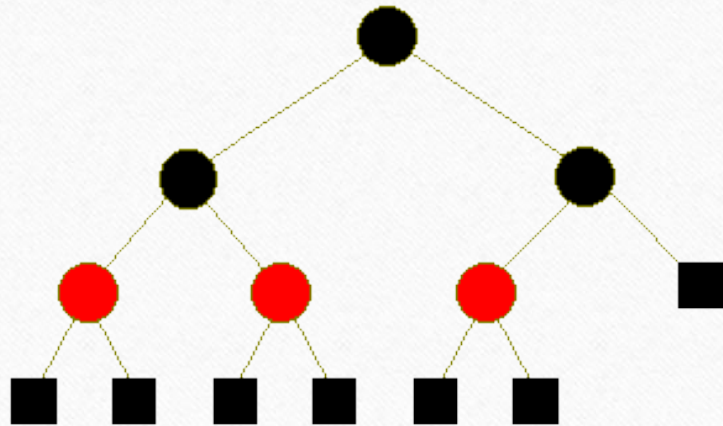
- Crea → create(tipo_param param_1, tipo_param param_2 ...)
- Un → articolo singolare → il metodo andrà ad operare su uno e un solo oggetto (il file)
- File → nome comune con significato particolare per IIDE
- Di → preposizione
- Nome → nome comune con significato particolare per IIDE
- Foglio → nome comune senza significati particolari per IIDE

Risultato: viene creato un file vuoto di nome «foglio»

Database parole chiave

Prima possibilità (Red Black Tree)

Strutture del database



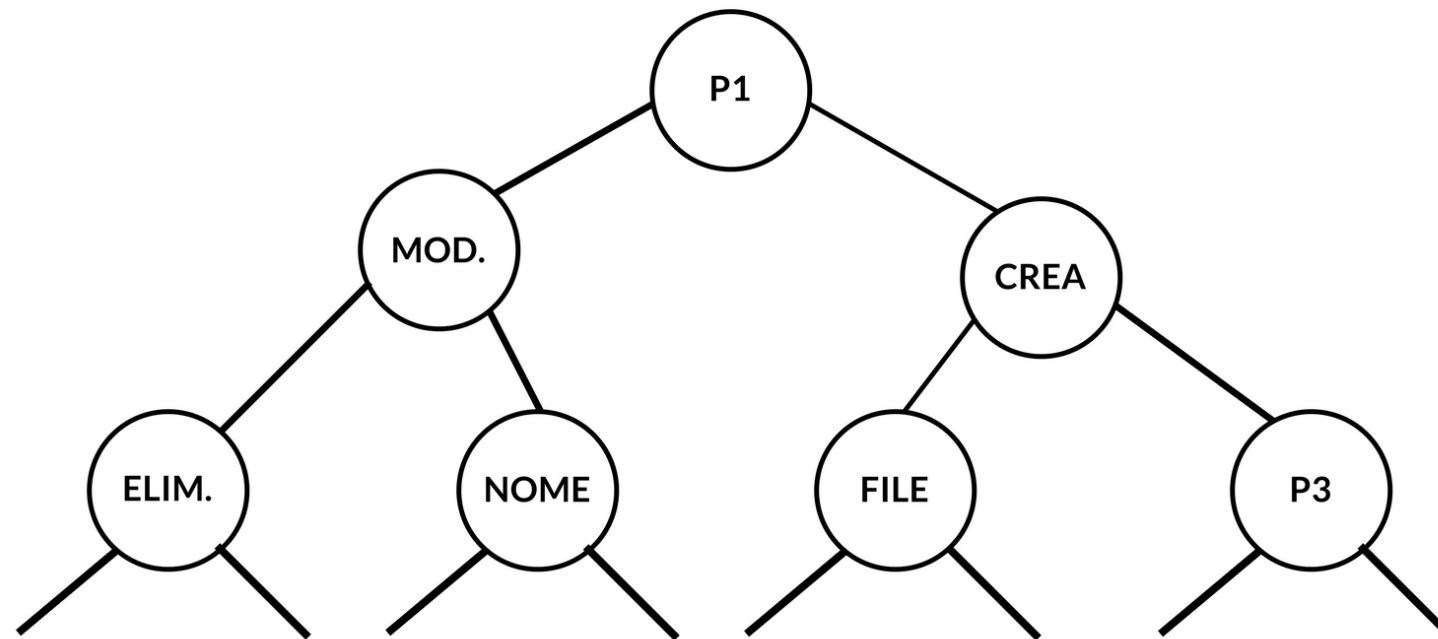
Red-Black Tree

- Ogni nodo rappresenta una parola chiave.
- Ogni parola ha associata una chiave univoca che corrisponde ad un numero intero.
- Inserimento, ricerca e cancellazione sono ottimizzati: $\Theta(\log n)$.
- I nodi hanno almeno un campo «proprietà» di tipo `String[]`.
- Altre specifiche ancora da definire.

Esempio di ricerca di parole chiave data una frase in input

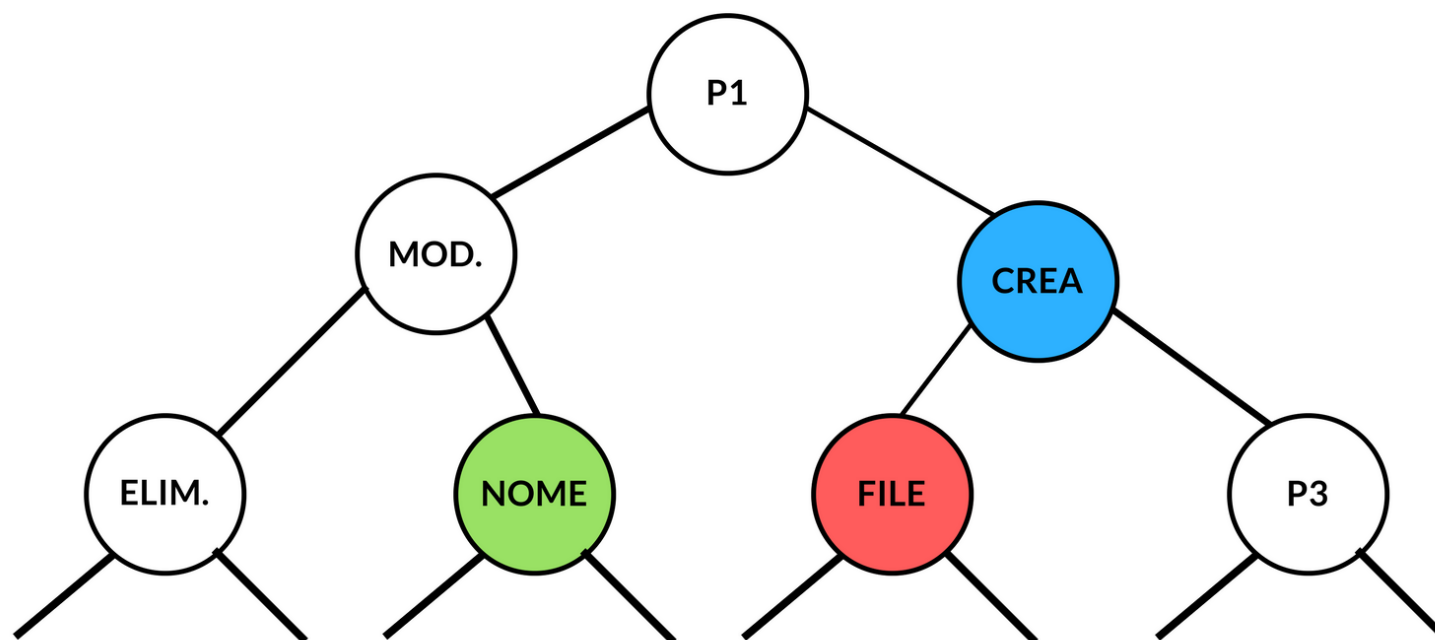
Nota: ogni parola trovata restituisce un insieme di proprietà
ottenibile come array di stringhe (String[])

Database parole chiave



Database parole chiave

Input: crea un file di nome orazio



Database parole chiave

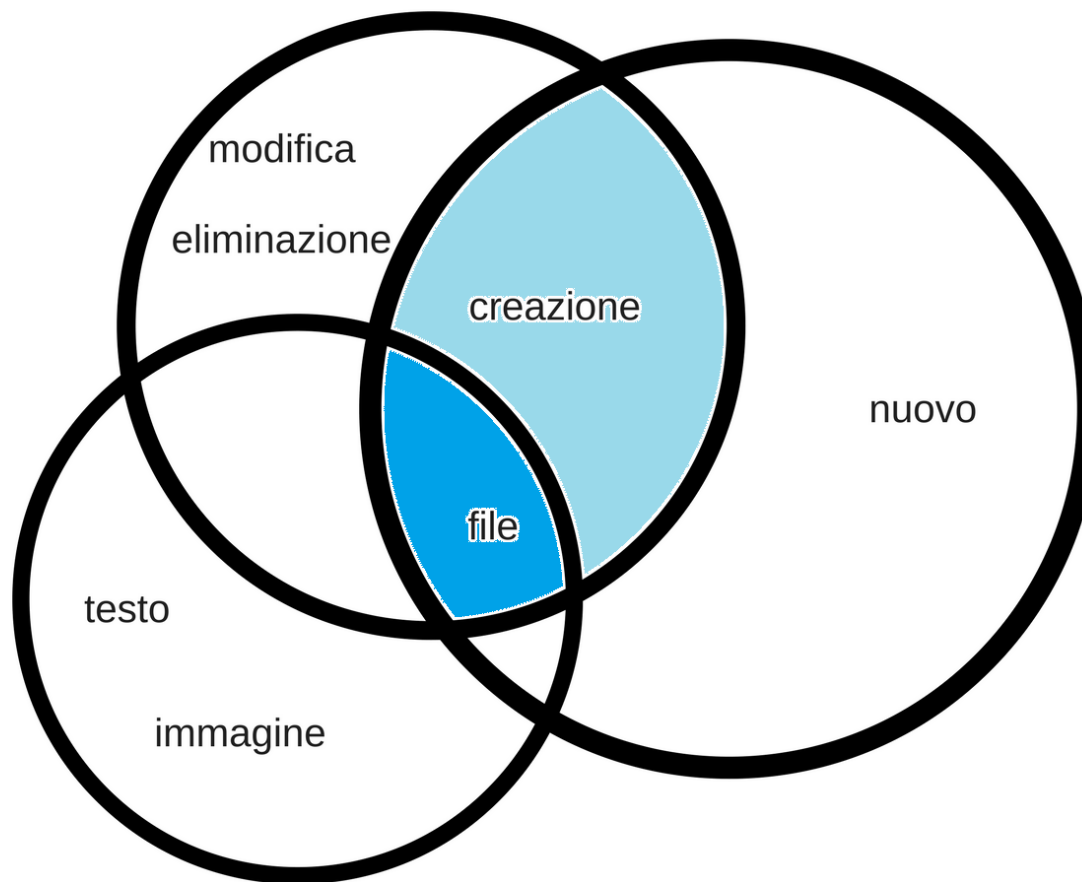
Input: crea un file di nome orazio

|
creazione, file,
nuovo, ...

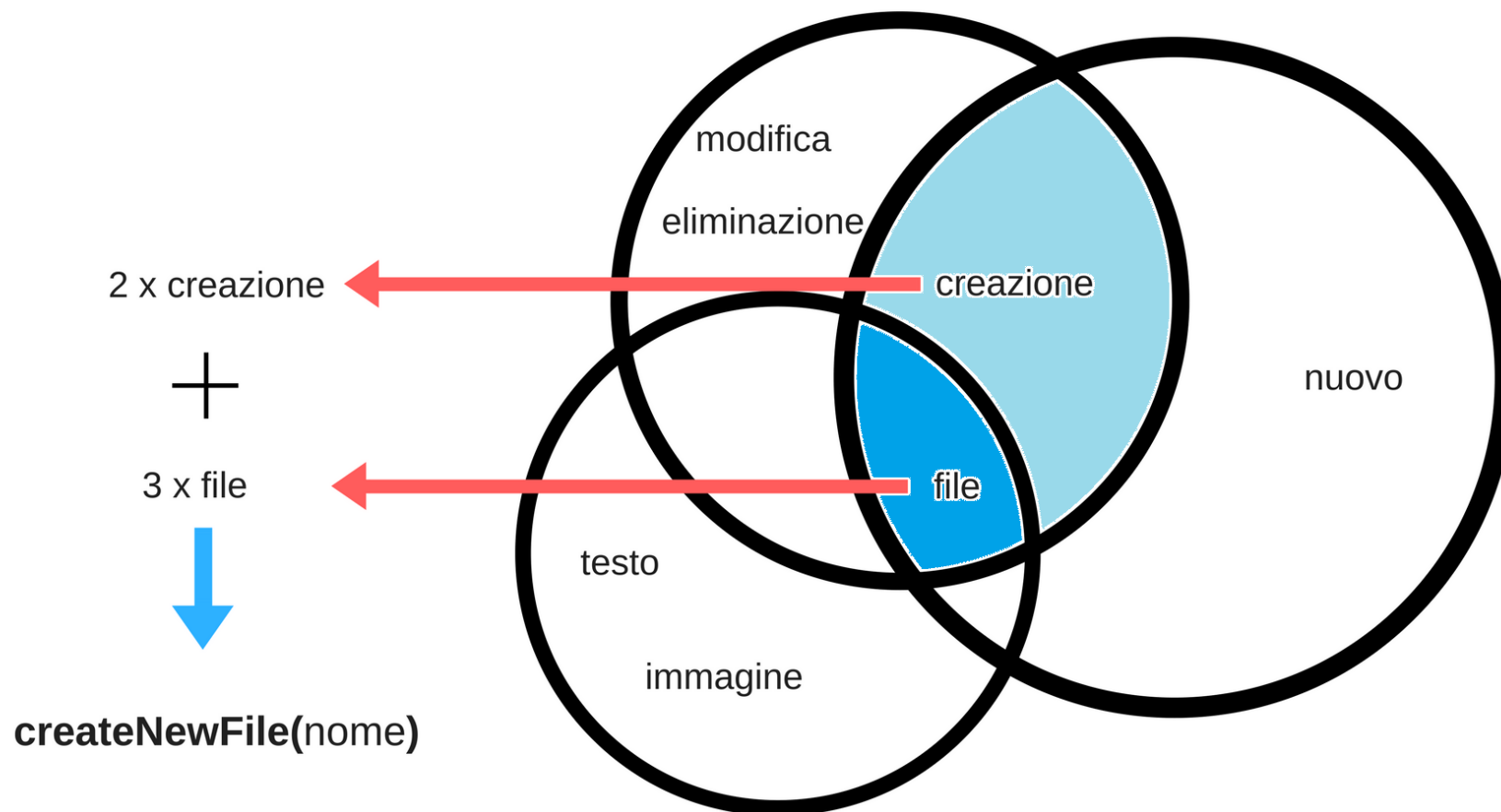
|
file, creazione,
modifica,
eliminazione, ...

|
file, testo,
immagine, ...

Database parole chiave



Database parole chiave

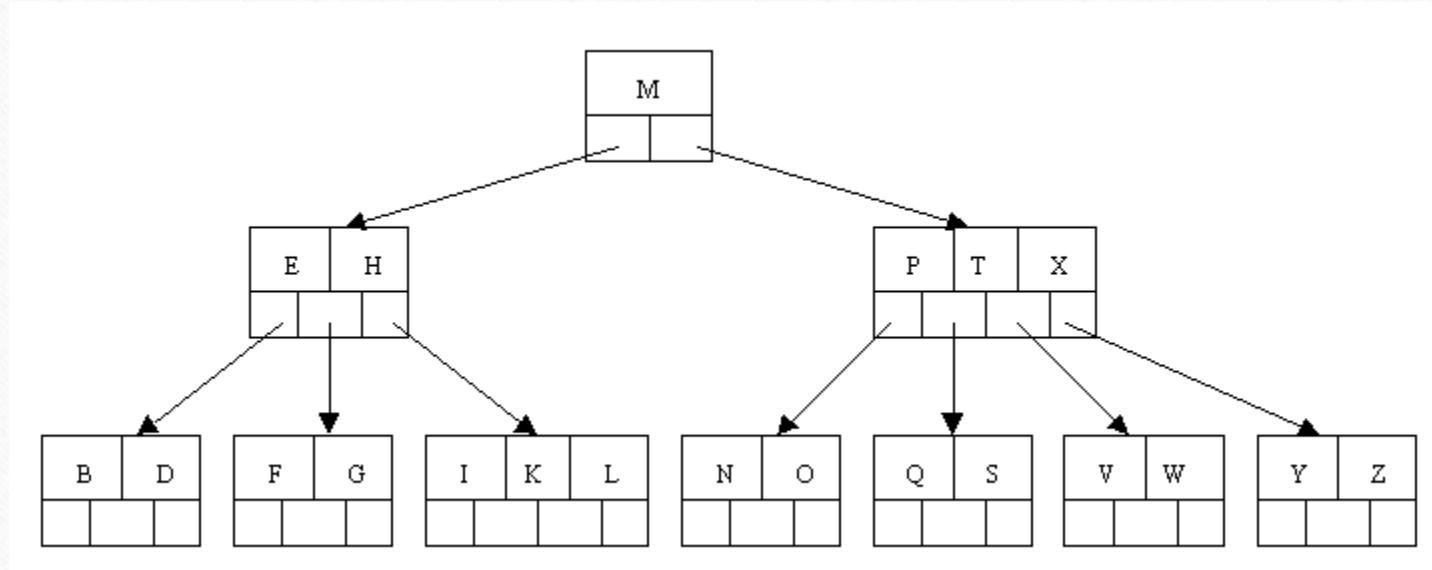


Seconda possibilità (B-Tree)

Perché i B-Tree?

- Il B-Tree principale conterrà le parole della lingua italiana fungendo da dizionario.
- Altri B-Tree secondari conterranno le parole chiave dei vari linguaggi di programmazione.
- I nodi dell'albero contengono poche informazioni e quindi più nodi possono essere caricati in RAM.
- Dato il gran numero di parole da memorizzare non è possibile utilizzare altri tipi di strutture dati.

Esempio di B-Tree



Struttura nodo del B-Tree

Campi e tipologia

1. **Chiave** – Intero
2. **Numero di chiavi salvate nel nodo attualmente** – Intero ($t-1 \leq n \leq 2t-1$)
3. **Array di chiavi** salvate in ordine non decrescente ($k_1 \leq k_2 \leq \dots \leq k_n$)
4. **Leaf** – campo booleano che indica se il nodo è una foglia
5. **Array di puntatori ai nodi figli** – String[] di dimensione $n+1$
6. **Numero** intero associato al risultato **dell'analisi** della parola rappresentata dal nodo.
7. **ID** creatore del nodo – Intero
8. **Metodo associato** (SE E SOLO SE il nodo è associato ad un verbo) – String
9. **Categoria speciale** – Intero
10. //Un intero per una possibile futura specifica
11. //Una stringa per una possibile futura specifica

Motivazioni e specifiche delle scelte progettuali del nodo

1. La chiave è un intero calcolato in base ad una funzione definita in modo tale che ogni parola p_1 diversa da p_2 abbia un intero corrispondente diverso e univoco \rightarrow dato un intero è possibile ricostruire la parola associata.
2. Il puntatore ad un altro nodo, dato che i nodi sono salvati come file in memoria secondaria, deve essere un percorso valido del filesystem.
3. /
4. Array di stringhe come per il punto 2.
5. Il numero intero indica nello specifico il risultato dell'analisi grammaticale della parola italiana associata alla chiave (ad esempio la parola «cane = nome comune di animale maschile singolare ecc...» ha come categoria x_1 e «il = articolo determinativo maschile singolare» avrà come categoria x_2)
6. Nel caso sia possibile che altri utenti aggiungano nodi al database è necessario dividerli in affidabili e root attraverso un meccanismo di login che affida ad ogni utente un identificativo intero (o una stringa) biunivoco.
7. Ad esempio la parola «crea» avrà associato il metodo `create(param1, param2)`.
8. La parola «file» ha un significato particolare in Java e deve essere quindi segnalata tale particolarità.

Project Packages



WordAnalysis

Questo package contiene le classi che a loro volta contengono i metodi utilizzati per codificare le parole dal vocabolario italiano e le loro proprietà.

Presa una parola appartenente al vocabolario italiano le sue proprietà codificate sono:

1. La parola stessa;
2. La sua analisi grammaticale;
3. // Altre proprietà ancora da definire...

File

Questo package contiene le classi che a loro volta contengono i metodi utilizzati per gestire in modo più agile e semplice la lettura, la scrittura e altre operazioni sui file.

Math

Questo package contiene le classi che a loro volta contengono i metodi utilizzati per eseguire alcune funzioni matematiche, come la valutazione di un polinomio, basato sulla struttura di una parola.

Test

Questo package contiene le classi che a loro volta contengono i metodi utilizzati per eseguire procedure di test sulla struttura dati del progetto e sui suoi metodi.

Progetto in fase di sviluppo...