**MakePartitions:**

-La classe prende in input N pagine sulle quali genera le xPath, dopodiché calcola tutte le differenti regole.

-Successivamente crea un associazione tra ogni pagina e le sue regole, poi raggruppa le pagine che contengono le stesse regole

-Vengono poi eliminati gli xPath equivalenti, e selezionato l’xPath caratteristico di ogni ClasseDiPagine come la più corta tra le regole.

-La **generazione delle partizioni** è affidata al metodo trovaTutteLePossibiliPartizioni i quali parametri e valori di ritorno sono commentati nel codice (la partizione è rappresentata nel codice con la classe “InsiemeDiClassi”).

Questo metodo è **ricorsivo** e partendo da UNA ClasseDiPagine alla volta trova tutte le possibili partizioni.

Il metodo è abbastanza complesso ma trova tutte le possibili combinazioni.

-Vengono poi eliminate le partizioni UGUALI che vengono inevitabilmente generate.

-Infine PER OGNI PARTIZIONE viene controllato di quale partizione è raffinamento e di quale no. Il metodo per controllare la relazione di raffinamento si basa su due partizioni: quella “principale” e quella da “controllare”. Questa condizione deve essere sempre valida: Un insieme di pagine contenuto nella ClasseDiPagine della partizione “da controllare” deve essere sottoinsieme di un insieme di un insieme di pagine contenuto nella ClasseDiPagine della partizione “principale” per far si che ci sia una relazione di raffinamento. (**presente una classe di test per questo metodo)**

Tutte le operazioni all’inizio del metodo sono manipolazioni delle collezioni delle classi volte a rendere semplice il controllo della condizione di sopra.

**XPATH EQUIVALENTI (presente una classe di test)**

L’eliminazione delle regole xPath equivalenti si basa sull’uso della classe “VettoreDiValori” il quale metodo equals() è stato riscritto:

questa classe contiene un array indicizzato dalle pagine (pos1->pag1, pos2->pag2 ecc che viene riempito tramite il metodo estrai Nodi() e in particolare tramite il metodo evaluateXPath() il quale testa la regola xPath (xPath inserito nel costruttore della classe) e salva per ogni pagina i nodi che estrae la regola per ogni pagina.

**public** NodeList[] estraiNodi() **throws** XPathExpressionException {

**final** NodeList[] nodiEstratti = **new** NodeList[n\_Pag];

**for**(Page p : **this**.classe.getPages()) {

NodeList node = *evaluateXPath*(p.getDocument(), **this**.xPath);

//inserisco i valori estratti dalla regola nelle posizioni dell'array corrispondenti all'id pagina, che è una stringa e va convertito in int

nodiEstratti[Integer.*parseInt*(p.getId())]= node;

**for**(**int** i=0; i<nodiEstratti.length;i++) {

}

}

**return** nodiEstratti;

}

**LAUNCH CONFIG**

Sono presenti nel progetto 4 configurazioni di lancio per osservare l’OUTPUT dell’algoritmo.

Si trovano in fondo al progetto (vicino al pom.xml)