Appendix

1 Hyperparameters

Table 1 summarizes the hyperparameters used in our experiments. len_{code} and len_{sum} are the sequence length of code and summary, respectively. Each covers at least 90% of the training set. $vocab_{code}$, $vocab_{sum}$, and $vocab_{ast}$ are the vocabulary size of code, summary, and AST. len_{pos} refers to the clipping distance in relative position. d_{Emb} is the dimension of the embedding layer. $heads$ and $layers$ indicate the number of layers and heads in Transformer, respectively. d_{ff} is the hidden layer dimension of feed-forward in Transformer. d_{RvNN} and $activate_f$ are dimension of hidden state and activate function in RvNN, respectively. $layer_{ff}$ is the number of the feed-forward layer in the copy component.

The values of these hyperparameters are set according to the related work (Ahmad et al., 2020; Zhang et al., 2019). We adjust the sizes of d_{Emb}, d_{ff}, and d_{RvNN} empirically. The batch size is set according to the computing memory.

	Parameter	TL-CodeSum	Funcom
Data	len_{code}	150	110
	len_{sum}	50	13
	$vocab_{code}$	50000	50000
	$vocab_{sum}$	30000	30000
	$vocab_{ast}$	10000	10000
Embedding	d_{Emb}	640	512
	len_{pos}	32	32
Transformer	d_{ff}	4096	2048
	$heads$	4	8
	$layers$	6	6
RvNN	d_{RvNN}	640	512
	$activate_f$	relu	relu
Copy	$layer_{ff}$	1	1
Training	$drop\ out$	0.2	0.2
	$optimizer$	Adam	Adam
	$batch\ size$	128	128
	$learning\ rate$	0.0001	0.0001

Table 1: Hyperparameters in our experiments.

Model	train/epoch	train/total	infer
Code2seq	41m48s	69h40m00s	0.0010s
HybridDrl	21h06m43s	633h21m30s	0.0139s
Astattgru	10m46s	07h10m40s	0.0037s
CodeAstnn	53m58s	44h04m22s	0.0090s
CAST	43m24s	26h02m24s	0.0073s

Table 2: Time cost of different AST representation models in Funcom.

Model	train/epoch	train/total	infer
Code2seq	01m51s	06h10m00s	0.0021s
HybridDrl	1h31m41s	45h50m30s	0.0514s
Astattgru	01m40s	06h36m40s	0.0073s
CodeAstnn	07m58s	26h33m20s	0.0269s
CAST	07m45s	25h50m00s	0.0358s

Table 3: Time cost of different AST representation models in TL-CodeSum.

2 Runtime of Different AST Representation Models

Tab2 and Table 3 show the time cost for differenct approaches on the two datasets. The 2nd column is the training time of one epoch. The 3rd column is the total training time. And the last column is the inference time per function. On the Funcom dataset, the baselines have different training time. Most of them range from 7 to 69 hours (except that Hybrid-DRL takes 633 hours). The inference time per function is 0.0073s for CASTS and 0.001 to 0.0139s for other baselines. Our approach has comparable time cost as the baselines. Similar performance can also be observed on TL-CodeSum. During training, HybridDrl firstly trains hybrid code representation by applying LSTM and Tree-LSTM to code token and AST. Then it generates summary based on actor-critic reinforcement learning. Therefore, it takes long time for HybridDrl to train the model.

3 Experimental Results on Deduplicated Dataset

In our experiment, we find the existence of code duplication in TL-CodeSum: around 20% code snippets in the testing set can be found in the training set. Thus, we remove the duplicated samples from the testing set and re-evaluate all approaches. Table 4 shows the result of different models in the rest testings set without duplicated samples. Our model still outperforms all baselines.

Model	Bleu-4	Meteor	Rouge-L	Cider
CodeNN	11.04	7.57	20.61	0.51
HDeepcom	10.58	7.18	20.23	0.45
Attgru	12.03	7.94	22.25	0.55
Astattgru	14.37	9.47	26.01	0.78
HybridDrl	13.27	7.26	23.47	0.61
Code2seq	13.91	7.62	21.44	0.42
CodeAstnn	23.94	14.91	36.92	1.68
NCS	23.46	15.12	37.42	1.66
CAST	**27.13**	**17.14**	**39.97**	**2.03**

Table 4: Performance of different models on deduplicated TL-CodeSum dataset.

4 Evaluation Metrics

We provide the details of the evaluation metrics we used in the experiments.

4.1 BLEU

BLEU measures the average n-gram precision between the reference sentences and generated sentences, with brevity penalty for short sentences. The formula to compute BLEU-1/2/3/4 is:

$$\text{BLEU-N} = BP \cdot \exp \sum_{n=1}^{N} \omega_n \log p_n, \tag{1}$$

where p_n (n-gram precision) is the fraction of n-grams in the generated sentences which are present in the reference sentences, and ω_n is the uniform weight 1/N. Since the generated summary is very short, high-order n-grams may not overlap. We use the +1 smoothing function (Lin and Och, 2004). BP is brevity penalty given as:

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{2}$$

Here, c is the length of the generated summary, and r is the length of the reference sentence.

4.2 ROUGE-L

Based on longest common subsequence (LCS), ROUGE-L is widely used in text summarization. Instead of using only recall, it uses F-score which is the harmonic mean of precision and recall values. Suppose A and B are generated and reference summaries of lengths c and r respectively, we have:

$$\begin{cases} P_{ROUGE-L} = \frac{LCS(A,B)}{c} \\ R_{ROUGE-L} = \frac{LCS(A,B)}{r} \end{cases} \tag{3}$$

$F_{ROUGE-L}$, which indicates the value of ROUGE-L, is calculated as the weighted harmonic mean of $P_{ROUGE-L}$ and $R_{ROUGE-L}$:

$$F_{ROUGE-L} = \frac{\left(1+\beta^2\right) P_{ROUGE-L} \cdot R_{ROUGE-L}}{R_{ROUGE-L} + \beta^2 P_{ROUGE-L}} \tag{4}$$

β is set to 1.2 as in (Zhang et al., 2020; Wan et al., 2018).

4.3 METEOR

METEOR is a recall-oriented metric that measures how well the model captures the content from the references in the generated sentences and has a better correlation with human judgment. Suppose m is the number of mapped unigrams between the reference and generated sentence with lengths c and r respectively. Then, precision, recall and F are given as:

$$P = \frac{m}{c}, \; R = \frac{m}{r}, \; F = \frac{PR}{\alpha P + (1-\alpha) R} \tag{5}$$

The sequence of mapping unigrams between the two sentences is divided into the fewest possible number of "chunks". This way, the matching unigrams in each "chunk" are adjacent (in two sentences) and the word order is the same. The penalty is then computed as:

$$\text{Pen} = \gamma \cdot \text{ frag }^{\beta} \tag{6}$$

where frag is a fragmentation fraction: frag $= ch/m$, where ch is the number of matching chunks and m is the total number of matches. The default values of α, β, γ are 0.9, 3.0 and 0.5 respectively.

4.4 CIDER

CIDER is a consensus-based evaluation metric used in image captioning tasks. The notions of importance and accuracy are inherently captured by computing the TF-IDF weight for each n-gram and using cosine similarity for sentence similarity. To compute CIDER, we first calculate the TF-IDF weighting $g_k(s_i)$ for each n-gram ω_k in reference sentence s_i. Here ω is the vocabulary of all n-grams. Then we use the cosine similarity between the generated sentence and the reference sentences to compute CIDER$_n$ score for n-grams of length n. The formula is given as:

$$\text{CIDER}_n \left(c_i, s_i\right) == \frac{< \boldsymbol{g^n} \left(c_i\right), \boldsymbol{g^n} \left(s_i\right) >}{\left\|\boldsymbol{g}^n \left(c_i\right)\right\| \left\|\boldsymbol{g}^n \left(s_i\right)\right\|} \tag{7}$$

where $\boldsymbol{g^n}(s_i)$ is a vector formed by $g_k(s_i)$ corresponding to all the n-grams (n varying from 1 to 4). c_i is the i^{th} generated sentence. Finally, the scores of various n-grams can be combined to calculate CIDER as follows:

$$\text{CIDER} \left(c_i, s_i\right) = \sum_{n=1}^{N} w_n \, \text{CIDER}_n \left(c_i, s_i\right) \tag{8}$$

5 AST/Subtree Statistics of Different Approaches

Table 5 and Table 6 show the statistics of AST node numbers in Funcom and TL-CodeSum, respectively. For full AST, we count the number of nodes in a full AST. For CAST and CodeAstnn, we count the number of nodes in the split AST. For Code2seq, we count the number of nodes in each path. For HybridDrl, we count the number of nodes in the transformed binary tree. For Astattgru, we count the number of tokens in SBT (the flattened AST).

Table 5, 6, 7, and 8 show that subtrees of CodeAstnn have fewer nodes and more subtrees numbers than CAST. The reason is that CodeAstnn splits every statement into a subtree, leading to a set of tiny trees which are 71% smaller than ours on TL-CodeSum and 53% on Funcom. This splitting granularity may not be conducive to capture the syntactic and semantic information. Take the TL-CodeSum as an example, the full ASTs are split into 13 subtrees on average and each subtree has an average of 7 nodes using ASTNN splitting technique. In contrast, they are split into 5 subtrees on average and each subtree has an average of 15 nodes by our splitting Algorithm. It is difficult to obtain semantics by modeling such few nodes and aggregating many fragmented subtree vectors. Moreover, CodeAstnn applies a RNN-based model to aggregate the subtrees and the convergence becomes worse as the number of subtrees increases (Bengio et al., 1993).

6 Human evaluation

We conduct a human evaluation to evaluate the effectiveness of the summaries generated by our approach CAST and the other three approaches. The shows that CAST outperforms the others in all three aspects: similarity, naturalness, and informativeness. we confirmed the dominance of our approach using Wilcoxon signed-rank tests for human evaluation. The result shown in Table 9 reflects that the improvement of CAST over other approaches is statistically significant with all p-values smaller than 0.05 at 95% confidence level (except CodeAstnn in the naturalness).

References

Wasi Uddin Ahmad, Saikat Chakraborty, Baishakhi Ray, and Kai-Wei Chang. 2020. A transformer-based approach for source code summarization. In *ACL*.

Yoshua Bengio, Paolo Frasconi, and Patrice Y. Simard. 1993. The problem of learning long-term dependencies in recurrent networks. In *ICNN*.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING*.

Yao Wan, Zhou Zhao, Min Yang, Guandong Xu, Haochao Ying, Jian Wu, and Philip S. Yu. 2018. Improving automatic source code summarization via deep reinforcement learning. In *ASE*.

Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, and Xudong Liu. 2020. Retrieval-based neural source code summarization. In *ICSE*.

Jian Zhang, Xu Wang, Hongyu Zhang, Hailong Sun, Kaixuan Wang, and Xudong Liu. 2019. A novel neural source code representation based on abstract syntax tree. In *ICSE*.

Approach	Avg.	Percentiles								
		20%	30%	40%	50%	60%	70%	80%	90%	100%
Full AST	56.0	24.0	28.0	33.0	40.0	52.0	67.0	87.0	117.0	550.0
Code2seq	5.0	1.0	2.0	3.0	4.0	5.0	7.0	8.0	9.0	58.0
HybridDrl	37.0	12.0	16.0	19.0	25.0	34.0	45.0	59.0	83.0	440.0
Astattgru	219.0	72.0	88.0	112.0	148.0	200.0	268.0	360.0	494.0	2344.0
CodeAstnn	7.0	1.0	3.0	5.0	6.0	7.0	8.0	10.0	13.0	336.0
CAST	15.0	5.0	6.0	8.0	10.0	14.0	16.0	20.0	32.0	474.0

Table 5: The statistics of AST node numbers in Funcom dataset.

Approach	Avg.	Percentiles								
		20%	30%	40%	50%	60%	70%	80%	90%	100%
Full AST	126.0	37.0	50.0	66.0	83.0	104.0	132.0	175.0	261.0	6165.0
Code2seq	5.0	1.0	2.0	3.0	4.0	6.0	7.0	8.0	9.0	43.0
HybridDrl	96.0	25.0	35.0	47.0	61.0	78.0	100.0	133.0	203.0	4707.0
Astattgru	534.0	152.0	208.0	274.0	348.0	440.0	560.0	748.0	1128.0	24412.0
CodeAstnn	7.0	1.0	1.0	4.0	7.0	8.0	9.0	11.0	16.0	761.0
CAST	24.0	7.0	10.0	14.0	16.0	20.0	25.0	33.0	48.0	6155.0

Table 6: The statistics of AST node numbers in TL-CodeSum dataset.

Approach	Avg.	Percentiles								
		20%	30%	40%	50%	60%	70%	80%	90%	100%
CodeAstnn	6.0	2.0	2.0	2.0	3.0	5.0	7.0	9.0	12.0	80.0
CAST	3.0	3.0	3.0	3.0	3.0	3.0	4.0	4.0	5.0	25.0

Table 7: The statistics of subtree numbers in Funcom dataset.

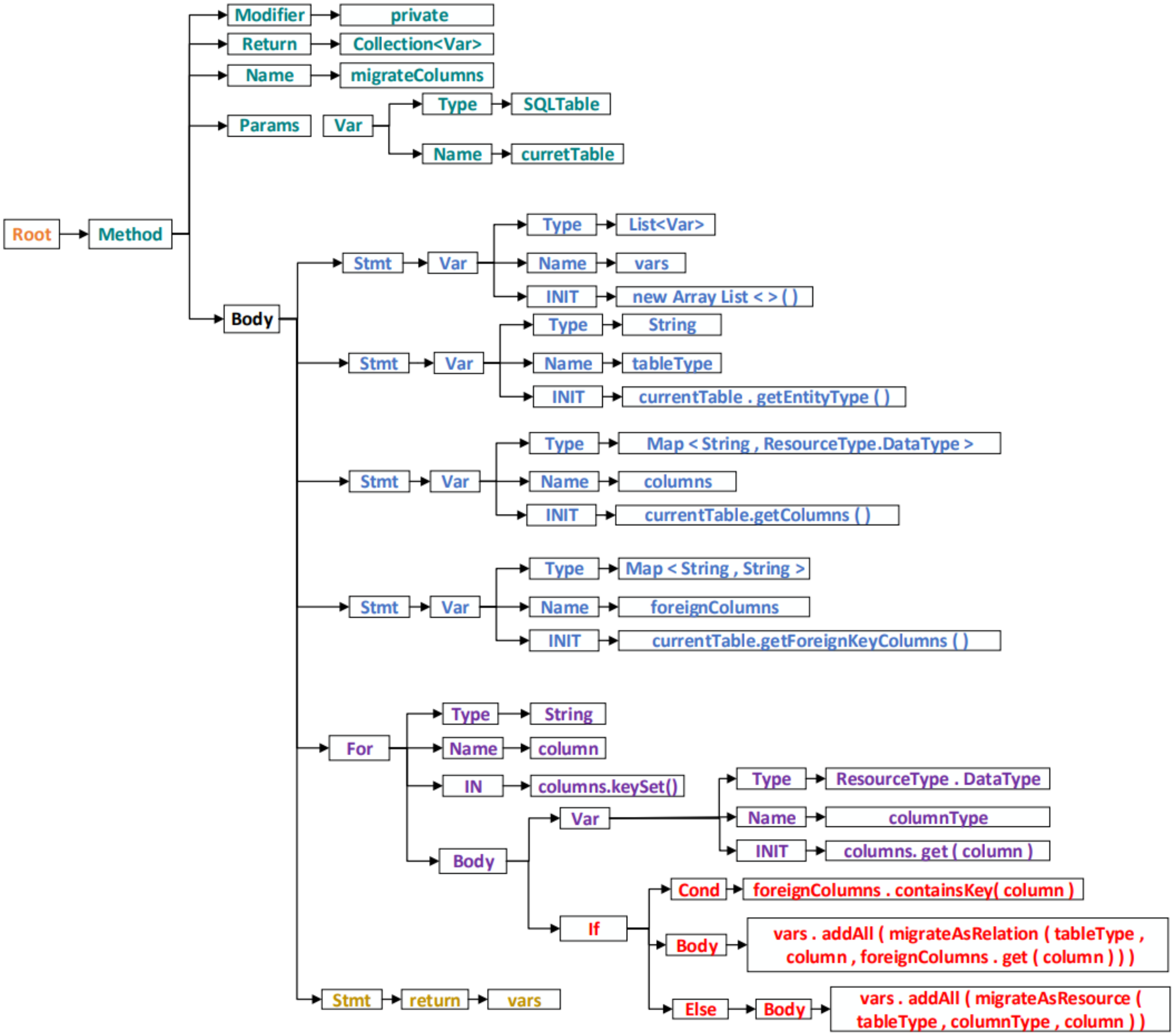
Approach	Avg.	Percentiles								
		20%	30%	40%	50%	60%	70%	80%	90%	100%
CodeAstnn	13.0	3.0	5.0	6.0	8.0	11.0	14.0	18.0	28.0	593.0
CAST	5.0	3.0	3.0	3.0	4.0	5.0	6.0	7.0	9.0	124.0

Table 8: The statistics of subtree numbers in TL-CodeSum dataset.

Model	Informativeness	Naturalness	Similarity
CAST	$1.83e^{-6}$	$5.15e^{-7}$	$8.71e^{-09}$
Astattgru	$5.23e^{-4}$	$7.49e^{-3}$	$3.53e^{-6}$
NCS	$1.34e^{-2}$	$4.66e^{-1}$	$2.86e^{-5}$

Table 9: Statistics significance p-value of CAST over other methods in human evaluation.

Full AST of Figure 1


Root
Method
Modifier
private
Return
Collection<Var>
Name
migrateColumns
Params
Var
Type
SQLTable
Name
curretTable
Body
Stmt
Var
Type
List<Var>
Name
vars
INIT
new Array List < > ( )
Stmt
Var
Type
String
Name
tableType
INIT
currentTable . getEntityType ( )
Stmt
Var
Type
Map < String , ResourceType.DataType >
Name
columns
INIT
currentTable.getColumns ( )
Stmt
Var
Type
Map < String , String >
Name
foreignColumns
INIT
currentTable.getForeignKeyColumns ( )
For
Type
String
Name
column
IN
columns.keySet()
Body
Var
Type
ResourceType . DataType
Name
columnType
INIT
columns. get (column)
If
Cond
foreignColumns . containsKey(column)
Body
vars . addAll (migrateAsRelation (tableType , column , foreignColumns . get (column)))
Else
Body
vars . addAll (migrateAsResource (tableType , columnType , column))
Stmt
return
vars

AST Splitting Algorithm

Algorithm 1 AST Splitting Algorithm

Input: The full AST of the code snippet, FT

Output: a set of subtrees, $ST = \{T_1, T_2, T_3, ...\}$

```
$OvT.root \leftarrow Root$
$parentStack.push(Root)$
splitTree($FT.Method$)
function splitTree($FT$)
    $ST \leftarrow []$
    $Subtrees \leftarrow$ visitMethodDeclaration($FT$)
    for $tree \in Subtrees$ do
        $result \leftarrow$splitTree($tree$)
        $ST \leftarrow ST \cup result$
    end for
    $ST \leftarrow OvT \cup ST$
    return $ST$
end function

function visitMethodDeclaration($tree$)
    $Subtrees \leftarrow []$
    if $tree.root$ isinstance $Method$ then
        $SigT \leftarrow$genMethSigTree($tree$)
        $Subtrees \leftarrow Subtrees \cup SigT$
        $OvT.addVertex(SigT)$
        $OvT.addEdge(parentStack.peek(), SigT)$
        $parentStack.push(SigT)$
        $tree = tree.Body$
        $OvT.addVertex(MethBody)$
        $OvT.addEdge(parentStack.peek(), MethBody)$
        $parentStack.push(MethBody)$
    end if
    return visitChildren($tree, Subtrees$)
end function

function visitChildren($tree, Subtrees$)
    int $n \leftarrow tree.getChildCount()$
    for $i = 1$ to $n$ do
        $t \leftarrow node.getChild(i)$
        switch ($t$)
        case 1: $t$ instanceof $IfStatement$:
            $st \leftarrow genIfTree(t)$
            $Subtrees \leftarrow Subtrees \cup st$
        ......
        case 9: $t$ instanceof $TryStatement$:
            $st \leftarrow genTryTree(t)$
            $Subtrees \leftarrow Subtrees \cup st$
        case 10: $t$ instanceof $Statement$:
            if $t.parent$ is $Root$
                $st \leftarrow genStatementsBlockTree(t)$
                $Subtrees \leftarrow Subtrees \cup st$
            end if
        end switch
    end for
    return $Subtrees$
end function
```

AST Splitting Rule Specification

```
----------------------------------------------------------------------
Input AST    =    Root [Method [Modifier [t*], Return [t], Name [t],
(Params [(Var [Type[t], Name [t]])+] ,)? Body [StmtT*]]
----------------------------------------------------------------------
StmtT        =    SimpleT | BlockT
OvT          =    Root [MethSig [MethBody [(StmtsT_r | BlockT_r)* ]]]
SigT         =    MethSig [Modifier [t*], Return [t], Name [t], (Params
[(Var [Type[t], Name [t]])+])?]
StmtsT       =    StatementsBlock [SimpleT +]
BlockT       =    IfT | ForT | WhileT | DoWhileT | SwitchT |LabelT |
SynchT | TryWithResourceT | TryT
SimpleT      =    VariableDeclaration | Expression | ReturnStatement
StmtsT_r     =    rootOf(StmtsT)
BlockT_r     =    rootOf(BlockT)

IfT          =    If [Cond[Expression], Body[StmtT +] (,Else[StmtT +])?]
ForT         =    for[forControl,Body[StmtT +]]|for-each[enForControl,
Body[StmtT +]]
forControl   =    (Init[VariableDeclaration | Expression])? ,
(Cond[Expression])?, (Update[Expression+])?
enForControl=   Modifier[t*] ,Type[t] , Name[t],In [t]
WhileT       =   While[Cond [Expression], Body [StmtT +] ]
DoWhileT     =   DoWhile[Cond [Expression], Body [StmtT +] ]
LabelT       =   Lable[Name(t), StmtT ]
SynchT       =   synchronized[Expression, Body[StmtT +]]
SwitchT      =   Switch[Expression, StmtGroup* switchLabel*]
StmtGroup    =   switchLabel+ ,blockStatement+
switchLabel =   case[Body[StmtT +]] |  default[Body[StmtT +]]
TryWithResourceT = Try[Resources,Body[SimpleT+],(Catch[StmtT+])*,
(Finally[StmtT+])?]
Resources    =   [Resource ,( Resource)*]
Resource     =   Modifier[t]* ,Type [t], Name[t], INIT['=' Expression]
Try          =   try[Body[SimpleT+], Catch[StmtT+],(Finally[StmtT+])?]
VariableDeclaration = Modifier[t*] ,Type[t] , Name[t], INIT['='
Expression]
ReturnStatement     = Return expression
t           =    identifier

expression  =
    | this| super | t | break | continue
    | expression, bop='.'
      ( t
      | methodCall
      | this
      | NEW nonWildcardTypeArguments? innerCreator
      | super superSuffix
      | explicitGenericInvocation
      )
    | expression '[' expression ']'
    | methodCall
    | NEW creator
    | Type[t],expression
    | expression, postfix=('++' | '--')
    | prefix=('+'|'-'|'++'|'--') ,expression
    | prefix=('~'|'!'), expression
    | expression, bop=('*'|'/'|'%') expression
    | expression, bop=('+'|'-') expression
    | expression ,('<' '<' | '>' '>' '>' | '>' '>') ,expression
    | expression, bop=('<=' | '>=' | '>' | '<'), expression
    | expression ,bop=INSTANCEOF , Type[t]
    | expression ,bop=('==' | '!=') ,expression
    | expression, bop='&', expression
    | expression, bop='^', expression
    | expression ,bop='|' ,expression
    | expression, bop='&&' ,expression
    | expression, bop='||', expression
bop=('=' | '+=' | '-=' | '*=' | '/=' | '&=' | '|=' | '^=' | '>>=' |
'>>>=' | '<<=' | '%=')
--------------------------------------------------------------------
Subtrees   =    {OvT} U {SigT} U {StmtsT*} U {BlockT*}
----------------------------------------------------------------------
```

**note: The Body and Block are treated the same in implementation.

100 Examples in Funcom

```
-----------------------------------------
code
        protected void addResourseIDPropertyDescriptor(Object object) {
                itemPropertyDescriptors.add
                        (createItemPropertyDescriptor
                                (((ComposeableAdapterFactory)adapterFactory).getRootAdapterFactory(),
                                 getResourceLocator(),
                                 getString("_UI_RootPanel_resourseID_feature"),
                                 getString("_UI_PropertyDescriptor_description",
"_UI_RootPanel_resourseID_feature", "_UI_RootPanel_type"),
                                 DictionaryPackage.Literals.ROOT_PANEL__RESOURSE_ID,
                                 true,
                                 false,
                                 false,
                                 ItemPropertyDescriptor.GENERIC_VALUE_IMAGE,
                                 getString("_UI_SecurityPropertyCategory"),
                                 null));
        }
Ground Truth: this adds a property descriptor for the res ourse id feature
CodeNN : this adds a property descriptor for the res feature
HDeepcom : this adds a property descriptor for the res id feature
Attnndgru : this adds a property descriptor for the res id feature
NCS :this adds a property descriptor for the res acompanhamento id feature
Code2seq : this adds a property descriptor for
HybridDrl: this adds a property descriptor for the identifier feature
Astattdgru: this adds a property descriptor for the res id feature
CodeAstnn : add a resource locator to the res try id
CAST  : this adds a property descriptor for the res ourse id feature
---------------------------------------
code
        private JTable getGrdFabricantes()
        {
                if (grdFabricantes == null)
                {
                        grdFabricantes = new Tabela();
                        grdFabricantes.setFillsViewportHeight(true);

grdFabricantes.setPreferredScrollableViewportSize(grdFabricantes.getPreferredSize());
                }

                return grdFabricantes;
        }
Ground Truth: this method initializes grd fabricantes
CodeNN : this the initializes
HDeepcom : this method initializes
Attnndgru : this method initializes
NCS :this method initializes grd {
Code2seq : this method initializes
HybridDrl: this method initializes j table
Astattdgru: this method initializes
CodeAstnn : this method initializes usg <unk>
CAST  : this method initializes grd fabricantes
---------------------------------------
code
        public Node getChild(int index) {
                return children[index];
        }
Ground Truth: returns the child at the given position
CodeNN : returns the child of the index index
HDeepcom : returns the child at the specified index index
Attnndgru : returns the child at the specified index
```

```
NCS :returns the child at the specified index
Code2seq : returns the child at the specified
HybridDrl: returns the child at the specified index
Astattdgru: returns the child at the specified index
CodeAstnn : returns the child at the specified index
CAST  : returns the child at the given index
---------------------------------------
code
        public void setChild(int index, Node c) {
                children[index] = c;
        }
Ground Truth: sets the child at the given position
CodeNN : returns the child of the index
HDeepcom : sets the child child at
Attnndgru : sets the child at the specified index
NCS :sets the child at the specified index
Code2seq : sets the child of the node
HybridDrl: sets the node at the given index
Astattdgru: sets the child node at the specified index
CodeAstnn : sets the child of the node at the specified index
CAST  : sets the child at the given index
---------------------------------------
code
     public void setMotif_id(java.lang.String motif_id) {
        this.motif_id = motif_id;
    }
Ground Truth: sets the motif id value for this motif result
CodeNN : returns the related to the column
HDeepcom : sets the id value for this id
Attnndgru : sets the id value for this
NCS :sets the motif id value for this custom tab
Code2seq : sets the id value for this
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the id value for this
CodeAstnn : sets the <unk> id value for this get <unk>
CAST  : sets the motif id value for this motif
---------------------------------------
code
     public int countValuesForArgument(String name) {
        List<?> list = options.get(name);
        if (list == null) return 0;
        return list.size();
    }
Ground Truth: counts the number of values for a given argument
CodeNN : returns the list of the the
HDeepcom : counts the number of of the the the
Attnndgru : returns the number of times the specified name is in the list
NCS :get the number of values for an argument
Code2seq : returns the number of options in
HybridDrl: returns the number of the option with the
Astattdgru: returns the number of values in the argument
CodeAstnn : returns the number of values for the given argument
CAST  : count the number of values for a given argument
---------------------------------------
code
     private void removeConnections(List<GraphConnector> connections) {
        for (GraphConnector conn : new LinkedList<GraphConnector>(connections)) {
            conn.disconnect();
        }
    }
Ground Truth: disconnects a list of connections from their endpoints
```

```
CodeNN : this the the connections
HDeepcom : removes all connections connections connections connections connections
Attnndgru : removes the given connections from the list
NCS :disconnects a list of connections with the target
Code2seq : disconnects all connections from the given
HybridDrl: disconnect the list of connections
Astattdgru: removes all connections from the network
CodeAstnn : removes all the connections from the list of connections
CAST  : disconnects a list of connections from their endpoints
---------------------------------------
code
 public void setNewSource(WorkflowEntityNode connectionSource) {
        if (connectionSource == null) {
                throw new IllegalArgumentException();
        }
        setLabel("move connection startpoint");
        newSource = connectionSource;
        newTarget = null;
}
Ground Truth: set a new source endpoint for this connection
CodeNN : this the source source
HDeepcom : sets the new source source
Attnndgru : sets the new source
NCS :set a new source endpoint for this connection when execute is invoked
Code2seq : sets the source node of the
HybridDrl: sets a new connection endpoint
Astattdgru: sets the new source endpoint for this connection
CodeAstnn : sets the new source
CAST  : set a new source endpoint for this connection
---------------------------------------
code
                public void onKeyUp(final KeyUpEvent event) {
                        if (event.getNativeKeyCode() == KeyCodes.KEY_ENTER) {
                                sendNameToServer();
                                playerName = nameField.getText();
                        }
                }
Ground Truth: fired when the user types in the name field
CodeNN : sets when the event the the
HDeepcom : called when the up up
Attnndgru : fired when the user types in the key
NCS :return the value of the name field
Code2seq : called when a key is pressed
HybridDrl: called when the player is
Astattdgru: called when a key is released
CodeAstnn : called when a key is pressed
CAST  : fired when the user types in the name field
---------------------------------------
code
        public DialUI
        getUI() { return (DialUI)ui; }
Ground Truth: gets the l f object that renders this component
CodeNN : sets the dial that dial dial
HDeepcom : returns the l dial dial this this this this this this this
Attnndgru : returns the ui object associated with this
NCS :returns the current look and feel l f object that renders this component
Code2seq : returns the ui object that is
HybridDrl: returns the ui object for this component
Astattdgru: returns the current dial ui
CodeAstnn : returns the dial ui
CAST  : returns the l f object that renders this component
```

```
---------------------------------------
code
        public void
        setUI(DialUI ui) { super.setUI(ui); }
Ground Truth: sets the l f object that renders this component
CodeNN : sets the look feel feel feel l l l l l l
HDeepcom : sets the l renders that this renders renders
Attnndgru : sets the look and feel l f object that renders this component
NCS :sets the look and feel l f object that renders this component
Code2seq : sets the ui object that this
HybridDrl: sets the ui
Astattdgru: sets the value of the code dial ui code object
CodeAstnn : sets the look and feel l f f f f f f f
CAST  : sets the l f object that renders this component
---------------------------------------
code
        public void
        updateUI() { setUI((DialUI)UIManager.getUI(this)); }
Ground Truth: resets the ui property to a value from the current look and feel
CodeNN : sets the ui f feel
HDeepcom : resets the ui property
Attnndgru : updates the ui
NCS :sets the ui property to a value from the current look and feel
Code2seq : sets the ui object to be
HybridDrl: updates the look and feel
Astattdgru: resets the ui property to the current value from the current
CodeAstnn : updates the ui with the given ui
CAST  : resets the ui property to a value from the current look and feel
---------------------------------------
code
     public void setShaded(boolean shade) {
        jShadeButton.setSelected(shade);
    }
Ground Truth: sets the shaded attribute of the j shade panel object
CodeNN : sets the the button
HDeepcom : sets the of the
Attnndgru : set the of the button
NCS :method to set whether or not to shade the shaded channel
Code2seq : set the shade of the button
HybridDrl: sets the value of the
Astattdgru: sets the
CodeAstnn : sets the <unk>
CAST  : sets the shaded attribute of the j shade button object
---------------------------------------
code
     public void setPiece(Piece piece, Field field) {

        List column = (List) columns.get(field.getColumn());

        column.set(field.getRow(), piece);

    }
Ground Truth: sets the piece attribute of the board object
CodeNN : returns the field the the the
HDeepcom : sets the piece of the the
Attnndgru : sets the piece of the given piece
NCS :sets the piece at the given position
Code2seq : sets the piece of the given
HybridDrl: sets the piece at the given piece
Astattdgru: sets the piece
CodeAstnn : sets the piece
```

```
CAST  : sets the piece attribute of the board object
---------------------------------------
code
   private void pressedAbort() {
    pfPasswd.setText("");
    bAborted = true;
    setVisible(false);
  }
Ground Truth: called when the abort button is pressed
CodeNN : method the the pressed
HDeepcom : called method abort
Attnndgru : method called when the user pres ses the cancel button
NCS :this method is called when the abort button is pressed
Code2seq : this method is called when the
HybridDrl: sets the text
Astattdgru: this method is called when the user clicks the abort button
CodeAstnn : called when abort button is pressed
CAST  : method called when the abort button is pressed
---------------------------------------
code
        public String nextLine() {
                if (!hasNextLine())
                        return null;
                return data[++lineNum];
        }
Ground Truth: returns the next line in the file
CodeNN : returns the next line line the line
HDeepcom : returns the next line
Attnndgru : returns the next line in the iteration
NCS :returns the next line in the input
Code2seq : returns the next line in the
HybridDrl: returns the next line
Astattdgru: returns the next available line in the input stream
CodeAstnn : returns the next line in the buffer
CAST  : returns the next line in the file
---------------------------------------
code
     public String getCurrentGameKey() {
        return game.getName();
    }
Ground Truth: gets the current game key
CodeNN : sets the game name the the game
HDeepcom : returns the current of
Attnndgru : returns the current game key
NCS :returns the name of the current game
Code2seq : returns the name of the game
HybridDrl: returns the name of the game
Astattdgru: returns the game key
CodeAstnn : returns the current game key
CAST  : gets the current game key
---------------------------------------
code
   public void resizeColumns(int columns) {
    if (numColumns == columns)
      return;

    if (columns <= 0) {
      throw new IndexOutOfBoundsException("Cannot set number of columns to "
        + columns);
    }
    if (numColumns > columns) {
      // Fewer columns. Remove extraneous cells.
      for (int i = 0; i < numRows; i++) {
        for (int j = numColumns - 1; j >= columns; j--)
          removeCell(i, j);
      }
    } else {
      // More columns. add cells where necessary.
      for (int i = 0; i < numRows; i++) {
        for (int j = numColumns; j < columns; j++)
          insertCell(i, j);
      }
    }
    numColumns = columns;
  }
Ground Truth: resizes the grid to the specified number of columns
CodeNN : returns the columns of columns the
HDeepcom : resizes the columns of the columns columns columns columns
Attnndgru : resizes the columns to fit the columns in the table
NCS :resizes the columns with the specified number of columns
Code2seq : sets the number of columns in
HybridDrl: sets the number of columns
Astattdgru: resizes the columns
CodeAstnn : resizes the given number of columns
CAST  : resize the grid to the specified number of columns
---------------------------------------
code
   public void cancel() {
    if (isRepeating)
      clearInterval(timerId);
    else
      clearTimeout(timerId);
    timers.remove(this);
  }
Ground Truth: cancels this timer
CodeNN : returns the timer timers
HDeepcom : cancels the timer
Attnndgru : cancels the timer
NCS :cancel the timer
Code2seq : removes all the timers from the
HybridDrl: cancel the
Astattdgru: cancels the timer
CodeAstnn : cancels the timer
CAST  : cancels this timer
---------------------------------------
code
   protected Element createCell() {
    return DOM.createTD();
  }
Ground Truth: creates a new cell
CodeNN : create a new of
HDeepcom : creates the cell cell
Attnndgru : creates the cell
NCS :create a new cell to be used for rendering the contents of the
Code2seq : creates a new td element
HybridDrl: returns the element of the
Astattdgru: creates the cell element
CodeAstnn : creates a cell
CAST  : creates a new cell
---------------------------------------
code
```

```
   protected void removeRow(int row) {
    int columnCount = getCellCount(row);
    for (int column = 0; column < columnCount; ++column) {
      cleanCell(row, column);
    }
    DOM.removeChild(bodyElem, DOM.getChild(bodyElem, row));
  }
Ground Truth: removes the specified row from the table
CodeNN : this the number of the the
HDeepcom : removes the row from the the
Attnndgru : removes a row from the table
NCS :this method removes the specified row from the table
Code2seq : removes the specified row from the
HybridDrl: removes a row from the table
Astattdgru: removes the row at the given row
CodeAstnn : removes a row from the table
CAST  : removes the specified row from the table
---------------------------------------
code
     public boolean intersects(Rectangle2D r) {
        return gp.intersects(r);
    }
Ground Truth: tests if the interior of this code shape code intersects the
CodeNN : returns whether the rectangle this this rectangle rectangle
HDeepcom : returns if the interior intersects the intersects intersects
Attnndgru : returns true if the given rectangle intersects this rectangle
NCS :tests if the interior of this code triangle code intersects the interior
Code2seq : checks if the rectangle intersects the
HybridDrl: returns true if the given rectangle intersects the
Astattdgru: returns true if this rectangle intersects the given rectangle
CodeAstnn : returns true if the given rectangle intersects the current
CAST  : tests if the interior of this code shape code intersects the interior of
---------------------------------------
code
     public void mouseWheelMoved(MouseWheelEvent e) {
        if (e.getModifiersEx() == MouseEvent.ALT_DOWN_MASK)
            thumbSizeModifier += (double) e.getWheelRotation() / 10.0;
        else
            increaseZoom(((double) e.getWheelRotation()) / 10d);
        repaint();
    }
Ground Truth: invoked when the mouse wheel is rotated
CodeNN : returns when wheel wheel wheel
HDeepcom : mouse mouse mouse mouse mouse mouse mouse
Attnndgru : mouse wheel moved event
NCS :handle mouse wheel movement
Code2seq : invoked when the mouse wheel has
HybridDrl: invoked when a mouse button is clicked
Astattdgru: handles mouse wheel moved event
CodeAstnn : invoked when a mouse wheel has been moved on a component
CAST  : invoked when the mouse wheel is moved
---------------------------------------
code
        private JPanel getJPanelFondo() {
                if (jPanelFondo == null) {
                        jPanelFondo = new JPanel();
                        jPanelFondo.setLayout(new BorderLayout());
                        //jPanelFondo.setPreferredSize(new Dimension(400, 40));
                        jPanelFondo.setName("Fondo");
                        jPanelFondo.add(getJPanelBotones(), BorderLayout.NORTH);
                }
                return jPanelFondo;
        }
Ground Truth: this method initializes j panel fondo
CodeNN : this method initializes j panel
HDeepcom : this method initializes j panel
Attnndgru : this method initializes j panel
NCS :this method initializes j panel {
Code2seq : this method initializes j panel
HybridDrl: this method initializes j panel
Astattdgru: this method initializes j panel
CodeAstnn : this method initializes j panel clr
CAST  : this method initializes j panel fondo
---------------------------------------
code
        private JPanel getJPanelComitente() {
                if (jPanelComitente == null) {
                        jPanelComitente = new JPanel();
                        jPanelComitente.setLayout(new BorderLayout());
                        jPanelComitente.setPreferredSize(new Dimension(400, 100));
                        jPanelComitente.setName("Comitente");
                        jPanelComitente.add(getJPanelBuscarComitente(), BorderLayout.NORTH);
                        jPanelComitente.add(getJPanelDatosComitente(), BorderLayout.SOUTH);
                }
                return jPanelComitente;
        }
Ground Truth: this method initializes j panel comitente
CodeNN : this method initializes j panel
HDeepcom : this method initializes j panel
Attnndgru : this method initializes j panel
NCS :this method initializes j panel {
Code2seq : this method initializes j panel
HybridDrl: this method initializes j panel
Astattdgru: this method initializes j panel
CodeAstnn : this method initializes j panel <unk>
CAST  : this method initializes j panel comitente
---------------------------------------
code
        private JPanel getJPanelFondo() {
                if (jPanelFondo == null) {
                        jPanelFondo = new JPanel();
                        jPanelFondo.setLayout(new BorderLayout());
                        jPanelFondo.setPreferredSize(new Dimension(400, 100));
                        jPanelFondo.setName("Fondo");
                        jPanelFondo.add(getJPanelDatos(), BorderLayout.NORTH);
                        jPanelFondo.add(getJPanelBotones(), BorderLayout.EAST);
                }
                return jPanelFondo;
        }
Ground Truth: this method initializes j panel fondo
CodeNN : this method initializes j panel
HDeepcom : this method initializes j panel
Attnndgru : this method initializes j panel
NCS :this method initializes j panel {
Code2seq : this method initializes j panel
HybridDrl: this method initializes j panel
Astattdgru: this method initializes j panel
CodeAstnn : this method initializes j panel clr
CAST  : this method initializes j panel fondo
---------------------------------------
code
        private JTextField getJTextFieldComitente() {
                if (jTextFieldComitente == null) {
                        jTextFieldComitente = new JTextField();
                        jTextFieldComitente.setEditable(false);
                        jTextFieldComitente.setFont(new Font("Dialog", Font.BOLD, 12));
                        jTextFieldComitente.setForeground(Color.blue);
                        jTextFieldComitente.setColumns(20);
                }
                return jTextFieldComitente;
        }
Ground Truth: this method initializes j text field comitente
CodeNN : this method initializes j text field
HDeepcom : this method initializes j text field
Attnndgru : this method initializes j text field
NCS :this method initializes j text field {
Code2seq : this method initializes j text field
HybridDrl: this method initializes j text field
Astattdgru: this method initializes j text field
CodeAstnn : this method initializes j text field <unk>
CAST  : this method initializes j text field comitente
---------------------------------------
code
     public int getOriHeight() {

        return m_oriHeight;
    }
Ground Truth: returns the ori height
CodeNN : returns the height of the
HDeepcom : returns the height height
Attnndgru : returns the height of the
NCS :returns the meaning of the meaning of ori
Code2seq : returns the height of the node
HybridDrl: returns the number of the
Astattdgru: returns the height
CodeAstnn : returns the height of the height of the image
CAST  : returns the ori height
---------------------------------------
code
     public void playTestCase(String className) throws Exception {

        Class testCaseClass = Class.forName(className);
        TestRunner.run(testCaseClass);

    }
Ground Truth: play the test case
CodeNN : test the for
HDeepcom : play a test case case
Attnndgru : plays the test case
NCS :run the specified test case
Code2seq : run a test case
HybridDrl: sets the class name of the
Astattdgru: play a wav file
CodeAstnn : plays a test case
CAST  : play the test case
---------------------------------------
code
     public HashCodeBuilder append(long value) {
        iTotal = iTotal * iConstant + ((int) (value ^ (value >> 32)));
        return this;
    }
Ground Truth: p append a code hash code code for a code long code
CodeNN : returns a hash code value this
```

```
HDeepcom : append the code code code code code code code code code code
Attnndgru : append a code hash code code for the given value
NCS :appending a code hash code code for a code long code
Code2seq : add a hash code to the
HybridDrl: appends a long value to the
Astattdgru: append the code hash code code for the given code long code
CodeAstnn : appends the specified long value to this hash code
CAST  : append a code hash code code for a code long code
---------------------------------------
code
     public HashCodeBuilder append(short value) {
        iTotal = iTotal * iConstant + value;
        return this;
    }
Ground Truth: p append a code hash code code for a code short code
CodeNN : sets the value of the code
HDeepcom : append the code code code code code code code code code code
Attnndgru : append a code hash code code for the given code short code
NCS :appending a code hash code code for a code short code
Code2seq : append a code hash code code
HybridDrl: appends the string representation of the
Astattdgru: append the code builder code code
CodeAstnn : appends the given short to the end of the hash code
CAST  : append a code hash code code for a code short code
---------------------------------------
code
     public HashCodeBuilder append(char value) {
        iTotal = iTotal * iConstant + value;
        return this;
    }
Ground Truth: p append a code hash code code for a code char code
CodeNN : sets the code value code code
HDeepcom : append the code code code code code code code code code code
Attnndgru : append a char value
NCS :appending a code char code code for a code char code
Code2seq : append a code hash code code
HybridDrl: appends the given value to the string
Astattdgru: append the given char value to the computation of the given
CodeAstnn : appends the specified char to the end of this hash code
CAST  : append a code hash code code for a code char code
---------------------------------------
code
     public HashCodeBuilder append(boolean value) {
        iTotal = iTotal * iConstant + (value ? 0 : 1);
        return this;
    }
Ground Truth: p append a code hash code code for a code boolean code
CodeNN : returns a the code the
HDeepcom : append a boolean value code code code
Attnndgru : append a code hash code code for the code result set code
NCS :appending a code hash code code for a code boolean code
Code2seq : append a code hash code code
HybridDrl: appends the string value to the
Astattdgru: append a boolean value
CodeAstnn : appends the given boolean to the end of the hash code
CAST  : append a code hash code code for a code boolean code
---------------------------------------
code
     public HashCodeBuilder append(char[] array) {
        if (array == null) {
            iTotal = iTotal * iConstant;
        } else {
            for (int i = 0; i < array.length; i++) {
                append(array[i]);
            }
        }
        return this;
    }
Ground Truth: p append a code hash code code for a code char code array
CodeNN : returns the array array
HDeepcom : append a array code code code code code code code code code
Attnndgru : append the given char array to the given array
NCS :appending a code hash code code for a code char code array
Code2seq : append a code hash code code
HybridDrl: appends an array of characters to the
Astattdgru: append the code hash code code for an array
CodeAstnn : appends the given array of characters to this hash code
CAST  : append a code hash code code for a code char code array
---------------------------------------
code
     public HashCodeBuilder append(boolean[] array) {
        if (array == null) {
            iTotal = iTotal * iConstant;
        } else {
            for (int i = 0; i < array.length; i++) {
                append(array[i]);
            }
        }
        return this;
    }
Ground Truth: p append a code hash code code for a code boolean code array
CodeNN : returns a array code code
HDeepcom : append append code code code code code code code code code code
Attnndgru : append the code hash code to the given string
NCS :appending a code hash code code for a code boolean code array
Code2seq : append a code hash code code
HybridDrl: appends an array of array to the array
Astattdgru: append a boolean array
CodeAstnn : appends the specified array to this hash code
CAST  : append a code hash code code for a code boolean code array
---------------------------------------
code
     public void deselectAll() {
        checkWidget();
        this.table.deselectAll();
    }
Ground Truth: deselects all selected items in the receivers list
CodeNN : returns the items in the the receiver receiver receiver receiver receiver receiver
HDeepcom : deselects all selected items in the receivers
Attnndgru : deselects all selected items in the receiver
NCS :deselects all items in the receivers list
Code2seq : deselects all items
HybridDrl: deselects all selected items
Astattdgru: deselects all selected items in the receiver
CodeAstnn : deselects all the items in the receiver
CAST  : deselects all selected items in the receivers list
---------------------------------------
code
     public boolean getEditable() {
        checkWidget();
        return this.text.getEditable();
    }
```

```
Ground Truth: gets the editable state
CodeNN : returns the editable editable
HDeepcom : returns code editable code if the is
Attnndgru : returns code true code if the receiver is editable
NCS :returns the editable state
Code2seq : returns code true code if the
HybridDrl: returns true if the receiver is text
Astattdgru: returns whether or not the text is editable
CodeAstnn : returns code true code if the receiver is editable
CAST  : gets the editable state
---------------------------------------
code
     public Point getSelection() {
        checkWidget();
        return this.text.getSelection();
    }
Ground Truth: returns a code point code whose x coordinate is the start of the
CodeNN : returns the selection selection the the
HDeepcom : returns code point code code the is is
Attnndgru : returns the selection which is the receivers
NCS :gets the selection
Code2seq : returns the selection point in the
HybridDrl: returns a point describing the receivers selection
Astattdgru: returns the selection which is the first selected
CodeAstnn : returns the selection
CAST  : returns a code point code whose x coordinate is the start
---------------------------------------
code
        private E extract() {
                final E[] items = this.items;
                E x = items[takeIndex];
                items[takeIndex] = null;
                takeIndex = inc(takeIndex);
                --count;
                return x;
        }
Ground Truth: extracts element at current take position advances and signals
CodeNN : increment the item from the the the
HDeepcom : extracts the from from from the the
Attnndgru : extracts the item from the array
NCS :extract element from queue
Code2seq : removes the element from the list
HybridDrl: returns the element at the
Astattdgru: extracts the element from the list
CodeAstnn : extracts the item from the list
CAST  : extracts element at current take position advances and signals
---------------------------------------
code
     public void setGroupNameCol(java.lang.String groupNameCol) {
        this.groupNameCol = groupNameCol;
    }
Ground Truth: setter for property group name col
CodeNN : sets the column name column column column column
HDeepcom : sets the group group col value for this column group
Attnndgru : sets the group name col value for this group name
NCS :sets the group name col value for this report definition field
Code2seq : setter for property group name
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the group name col value for this group name type
CodeAstnn : sets the group name col value for this group name type
CAST  : setter for property group name col
```

```
---------------------------------------
code
     public void setJoinRoleCol(java.lang.String joinRoleCol) {
        this.joinRoleCol = joinRoleCol;
    }
Ground Truth: setter for property join role col
CodeNN : sets the property related to for column column role
HDeepcom : sets the join role column value for this join role
Attnndgru : sets the join role col value for this role
NCS :sets the join role col value for this report aggregate
Code2seq : setter for property join role
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the join role col value for this join role
CodeAstnn : sets the join role col value for this join role type
CAST  : setter for property join role col
---------------------------------------
code
     public void setRoleNameCol(java.lang.String roleNameCol) {
        this.roleNameCol = roleNameCol;
    }
Ground Truth: setter for property role name col
CodeNN : sets the column column column column column column
HDeepcom : sets the role column column value for this role col
Attnndgru : set the value related to the column role name role
NCS :set the value related to the column role name col
Code2seq : setter for property column name
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the role name col value for this role type
CodeAstnn : sets the role name col value for this user
CAST  : setter for property role name col
---------------------------------------
code
     public long getChildrenCardinality() {
        return childrenCardinality;
    }
Ground Truth: returns the children cardinality
CodeNN : returns the children of
HDeepcom : returns the cardinality cardinality
Attnndgru : returns the number of children of this node
NCS :returns the cardinality of the children of this node
Code2seq : returns the cardinality of this node
HybridDrl: returns the value of the
Astattdgru: returns the cardinality of the children
CodeAstnn : returns the number of children
CAST  : returns the children cardinality
---------------------------------------
code
   public void setDimVisible(boolean isDimVisible) {
    this.isDimVisible = isDimVisible;
  }
Ground Truth: sets the is dim visible
CodeNN : sets the the of
HDeepcom : sets the the of
Attnndgru : sets the is dimension visible
NCS :sets the dim visible
Code2seq : sets the is visible
HybridDrl: sets the <unk>
Astattdgru: sets the dim visible
CodeAstnn : sets the dimension visible
CAST  : sets the is dim visible
---------------------------------------
```

```
code
   public int getAxisOrdinal() {
    return axisOrdinal;
  }
Ground Truth: returns the axis ordinal
CodeNN : returns the ordinal axis
HDeepcom : returns the the of
Attnndgru : returns the ordinal ordinal
NCS :returns the ordinal of the axis
Code2seq : returns the ordinal of the axis
HybridDrl: returns the number of the
Astattdgru: returns the ordinal value of this code code
CodeAstnn : returns the number of the axis
CAST  : returns the axis ordinal
---------------------------------------
code
   public String getUniqueNameSqlColumnName() {
    return uniqueNameSqlColumnName;
  }
Ground Truth: returns the unique name sql column name
CodeNN : returns the unique name name name column
HDeepcom : returns the name name column column column column column column
Attnndgru : returns the unique name of the column that is unique in the
NCS :gets the unique name sql column name
Code2seq : returns the name of the column
HybridDrl: returns the value of the
Astattdgru: returns the name of the column that will be used to create
CodeAstnn : returns the name of the sql name of the sql statement
CAST  : returns the unique name sql column name
---------------------------------------
code
        public PolicyLevel getLevel() {
                return level;
        }
Ground Truth: get the level of the policy
CodeNN : returns the level level
HDeepcom : returns the level level
Attnndgru : returns the level of the event
NCS :get the policy level for this policy
Code2seq : returns the level of the level
HybridDrl: returns the level
Astattdgru: returns the level of this policy
CodeAstnn : returns the policy level
CAST  : returns the level of the policy
---------------------------------------
code
     public void update(Org obj) throws DataAccessLayerException {
        super.saveOrUpdate(obj);
    }
Ground Truth: updates the state of a detached org
CodeNN : sets the data data the the
HDeepcom : updates the object
Attnndgru : update the data
NCS :update the database with the given object
Code2seq : save or update the given object
HybridDrl: updates the object
Astattdgru: update the object in the database
CodeAstnn : updates the data access layer
CAST  : updates the state of a detached event
---------------------------------------
code
```

```
     public void update(Parameter obj) throws DataAccessLayerException {
        super.saveOrUpdate(obj);
    }
Ground Truth: updates the state of a detached parameter
CodeNN : sets the data data the the
HDeepcom : updates the object
Attnndgru : update the data access object
NCS :updates the state of the given detached instance
Code2seq : save or update a parameter
HybridDrl: updates the parameter
Astattdgru: update the object in the database
CodeAstnn : update the data access layer
CAST  : updates the state of a detached event
---------------------------------------
code
 public StringItem getStringItemCertCertAlg () {
if (stringItemCertCertAlg == null) {//GEN-END:|41-getter|0|41-preInit
 // write pre-init user code here
stringItemCertCertAlg = new StringItem ("Signature algorithm:",
null);//GEN-LINE:|41-getter|1|41-postInit
 // write post-init user code here
}//GEN-BEGIN:|41-getter|2|
return stringItemCertCertAlg;
}
Ground Truth: returns an initiliazed instance of string item cert cert alg component
CodeNN : sets an initiliazed instance of string algorithm signature algorithm
HDeepcom : returns an initiliazed instance of string item cert algorithm algorithm component
Attnndgru : returns an initiliazed instance of string item cert enc ias
NCS :returns an initiliazed instance of string item cert alg component
Code2seq : returns an initiliazed instance of cert
HybridDrl: returns an initiliazed instance of string item component
Astattdgru: returns an initiliazed instance of string item cert alg component
CodeAstnn : returns an initiliazed instance of string item cert id component
CAST  : returns an initiliazed instance of string item cert cert alg component
---------------------------------------
code
   public int indexIn(CharSequence sequence, int start) {
    int length = sequence.length();
    Preconditions.checkPositionIndex(start, length);
    for (int i = start; i < length; i++) {
      if (matches(sequence.charAt(i))) {
        return i;
      }
    }
    return -1;
  }
Ground Truth: returns the index of the first matching character in a character sequence
CodeNN : returns the sequence of the the the
HDeepcom : returns the index of the sequence sequence sequence sequence
Attnndgru : returns the index of the given character in the sequence
NCS :returns the index within the input string of the first occurrence of the
Code2seq : returns the index of the first
HybridDrl: returns the index of the given sequence
Astattdgru: returns the index of the given character sequence
CodeAstnn : returns the index of the specified sequence
CAST  : returns the index of the first matching character in a sequence
---------------------------------------
code
        public void addDirectory(File directory) {
                inputDirectories.add(directory);
        }
```

```
Ground Truth: adds a directory to the list of input directories
CodeNN : sets a new to to
HDeepcom : adds a directory directory the the
Attnndgru : adds a directory to the directory
NCS :adds a directory to the list of directories to process
Code2seq : adds a directory to the directory
HybridDrl: adds a directory to the directory
Astattdgru: add a directory to the list of files
CodeAstnn : adds a directory to the input directories
CAST  : add a directory to the list of input directories
---------------------------------------
code
     public void setDisplay_resolved(java.lang.Boolean display_resolved) {
        this.display_resolved = display_resolved;
    }
Ground Truth: sets the display resolved value for this custom field definition data
CodeNN : sets for resolved display value this this this
HDeepcom : sets the display resolved value for this column
Attnndgru : sets the display resolved value for this weather parameters type
NCS :sets the display resolved value for this email template
Code2seq : set the value related to the
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the display resolved value for this merchant item attributes
CodeAstnn : sets the display resolved value for this file
CAST  : sets the display resolved value for this custom field
---------------------------------------
code
     public TreeNode setCurrentParent(TreeNode newParent) {
        TreeNode oldParent = currentParent;
        currentParent = newParent;
        return oldParent;
    }
Ground Truth: set the current parent of the node
CodeNN : sets the parent of
HDeepcom : sets the current parent parent parent parent
Attnndgru : sets the parent of this node
NCS :sets the current parent of this node to code new parent code
Code2seq : insert the methods description here
HybridDrl: sets the tree node
Astattdgru: sets the parent node
CodeAstnn : sets the current parent of the tree
CAST  : sets the current parent of the node
---------------------------------------
code
        public void resetState() {

                m_initialState = m_fieldToMonitor.isSelected();
        }
Ground Truth: reset the state of the comparator
CodeNN : returns the state state state state
HDeepcom : resets the state state
Attnndgru : resets the state of the state
NCS :resets the state of this configuration
Code2seq : sets the initial state of the
HybridDrl: sets the selected
Astattdgru: resets the state of the state
CodeAstnn : resets the state of the field
CAST  : reset the state of the field
---------------------------------------
code
     public String getReplyAuthor() {
        return strReplyAuthor;
    }
Ground Truth: returns the author of the reply
CodeNN : returns the value of the
HDeepcom : gets the reply author
Attnndgru : gets the value of the reply author property
NCS :gets the value of the str reply author property
Code2seq : gets the value of the reply
HybridDrl: returns the value of the
Astattdgru: returns the reply author
CodeAstnn : returns the reply author
CAST  : get the author of the reply
---------------------------------------
code
     public void focusFilterField() {
        this.filterText.setFocus();
    }
Ground Truth: sets the focus to the filter text field
CodeNN : sets the focus filter
HDeepcom : focus the field field field field
Attnndgru : focus filter field
NCS :focus on the filter text field
Code2seq : passing the focus request to the
HybridDrl: sets the focus
Astattdgru: focus the text field
CodeAstnn : filter the filter field
CAST  : sets the focus to the filter text field
---------------------------------------
code
        public void setNumTel (java.lang.String numTel) {
                this.numTel = numTel;
        }
Ground Truth: set the value related to the column num tel
CodeNN : sets the num related to the column num
HDeepcom : sets the num tel value for column num telephone
Attnndgru : sets the num tel value for this admin promotion
NCS :sets the num tel value for this phone return
Code2seq : sets the num tel value for
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the num tel value for this
CodeAstnn : sets the num tel value for this get list
CAST  : set the value related to the column num tel
---------------------------------------
code
        public void setCbNum (java.lang.String cbNum) {
                this.cbNum = cbNum;
        }
Ground Truth: set the value related to the column cb num
CodeNN : sets the num value value value this cc
HDeepcom : sets the value num value for this cb num
Attnndgru : sets the cb num value for this
NCS :sets the cb num value for this phone return
Code2seq : sets the num num value for
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the cb num value for this docs is status obj
CodeAstnn : sets the cb num value for this get get response
CAST  : set the value related to the column cb num
---------------------------------------
code
        public void setCbCle (java.lang.String cbCle) {
                this.cbCle = cbCle;
        }
Ground Truth: set the value related to the column cb cle
CodeNN : returns the value value value the column
HDeepcom : sets the value value to the this cb cb cb
Attnndgru : sets the cb value for this
NCS :sets the cb { value for this merchant item attributes
Code2seq : set the value related to the
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: sets the cb value for this docs is status obj output
CodeAstnn : sets the cb <unk> value for this get get request
CAST  : set the value related to the column cb cle
---------------------------------------
code
        public void setCbValidAnnee (java.lang.Integer cbValidAnnee) {
                this.cbValidAnnee = cbValidAnnee;
        }
Ground Truth: set the value related to the column cb valid annee
CodeNN : sets the valid value value this column cc
HDeepcom : set the value related to the column cb valid
Attnndgru : sets the cb valid value for this
NCS :sets the cb valid annee value for this curriculum 1
Code2seq : sets the valid valid valid value
HybridDrl: sets the <unk> value for this <unk>
Astattdgru: set the value related to the column cb valid
CodeAstnn : sets the cb valid extract value for this get valid group
CAST  : set the value related to the column cb valid annee
---------------------------------------
code
        public void setCommandes (java.util.Set<commande.Commande> commandes) {
                this.commandes = commandes;
        }
Ground Truth: set the value related to the column comm andes
CodeNN : returns the comm comm
HDeepcom : set the value related to the column comm
Attnndgru : set the value related to the column comm
NCS :set the value related to the column comm )
Code2seq : setter for property
HybridDrl: set the value related to the column <unk>
Astattdgru: setter for property comm
CodeAstnn : set the value related to the column comm <unk>
CAST  : set the value related to the column comm andes
---------------------------------------
code
        public void setIngredientsPizza (java.util.Set<ingredientsPizza.IngredientsPizza>
ingredientsPizza) {
                this.ingredientsPizza = ingredientsPizza;
        }
Ground Truth: set the value related to the column ingredients pizza
CodeNN : removes the ingredients
HDeepcom : set the value related to the column ingredients
Attnndgru : sets the ingredients value for this
NCS :set the value related to the column ingredients only
Code2seq : sets the
HybridDrl: set the value related to the column <unk>
Astattdgru: setter for property ingredients
CodeAstnn : set the value related to the column cash botschaft
CAST  : set the value related to the column ingredients pizza
---------------------------------------
code
        public void setBoissons (java.util.Set<boisson.Boisson> boissons) {
                this.boissons = boissons;
        }
Ground Truth: set the value related to the column boissons
CodeNN : returns the related to the column
HDeepcom : set the value related to the column
Attnndgru : setter for property
NCS :set the value related to the column )
Code2seq : setter for the field
HybridDrl: set the value related to the column <unk>
Astattdgru: setter for property
CodeAstnn : set the value related to the column <unk>
CAST  : set the value related to the column boissons
---------------------------------------
code
        public void setPizzas (java.util.Set<pizza.Pizza> pizzas) {
                this.pizzas = pizzas;
        }
Ground Truth: set the value related to the column pizzas
CodeNN : returns the related to the column
HDeepcom : set the value related to the column
Attnndgru : setter for property
NCS :set the value related to the column )
Code2seq : set the value related to the
HybridDrl: set the value related to the column <unk>
Astattdgru: setter for property
CodeAstnn : set the value related to the column <unk>
CAST  : set the value related to the column pizzas
---------------------------------------
code
        public void setDateCommande (java.util.Date dateCommande) {
                this.dateCommande = dateCommande;
        }
Ground Truth: set the value related to the column date commande
CodeNN : returns the date related to the column date
HDeepcom : set the value related to the column date date
Attnndgru : sets the date value for this
NCS :set the value related to the column date {
Code2seq : sets the date value for this
HybridDrl: set the value related to the column date
Astattdgru: set the value related to the column date
CodeAstnn : set the value related to the column date <unk>
CAST  : set the value related to the column date commande
---------------------------------------
code
        public void setDateLivraison (java.util.Date dateLivraison) {
                this.dateLivraison = dateLivraison;
        }
Ground Truth: set the value related to the column date livraison
CodeNN : returns the date related to the column date
HDeepcom : set the value related to the column date date
Attnndgru : sets the date value for this
NCS :set the value related to the column date {
Code2seq : sets the date value value
HybridDrl: set the value related to the column date
Astattdgru: set the value related to the column date
CodeAstnn : set the value related to the column date <unk>
CAST  : set the value related to the column date livraison
---------------------------------------
code
        public void setBoissons (java.util.Set<commandeBoisson.CommandeBoisson> boissons) {
                this.boissons = boissons;
        }
```

```
Ground Truth: set the value related to the column boissons
CodeNN : returns the related to the column
HDeepcom : set the value related to the column
Attnndgru : setter for property
NCS :set the value related to the column )
Code2seq : set the value related to the
HybridDrl: set the value related to the column <unk>
Astattdgru: setter for property
CodeAstnn : set the value related to the column <unk>
CAST  : set the value related to the column boissons
---------------------------------------
code
        public void setPizzas (java.util.Set<commandePizza.CommandePizza> pizzas) {
                this.pizzas = pizzas;
        }
Ground Truth: set the value related to the column pizzas
CodeNN : returns the related to the column
HDeepcom : set the value related to the column
Attnndgru : setter for property
NCS :set the value related to the column )
Code2seq : set the value related to the
HybridDrl: set the value related to the column <unk>
Astattdgru: setter for property
CodeAstnn : set the value related to the column <unk>
CAST  : set the value related to the column pizzas
---------------------------------------
code
   public boolean remove(Object arg0) {
    Element<T> cursor = head;
    while (cursor != null && !(arg0 == null ? cursor.entry == null : cursor.entry.equals(arg0))) {
      cursor = cursor.next;
    }
    if (cursor == null) {
      return false;
    } else {
      removeInternal(cursor);
      return true;
    }
  }
Ground Truth: remove the given element from the list
CodeNN : returns if cursor from the the the
HDeepcom : remove the element element
Attnndgru : removes a single object from the list
NCS :remove the element at the front of this list
Code2seq : removes the specified object from the
HybridDrl: removes a element from the list
Astattdgru: removes the cursor with the given arg and value
CodeAstnn : removes the first occurrence of the specified element from this list
CAST  : removes the given element from the list
---------------------------------------
code
     public LiveSet firstPEIKillSet() {
      return firstPEIKillSet;
    }
Ground Truth: returns the first pei kill set i
CodeNN : sets the first of
HDeepcom : returns the first of of the
Attnndgru : sets the first alive set
NCS :first pei kill set
Code2seq : returns the set of stop
HybridDrl: returns the value of the
```

```
Astattdgru: returns the first of the
CodeAstnn : returns the first extended kill set
CAST  : returns the first pei kill set
---------------------------------------
code
   public void addGraphEdge(SpaceEffGraphEdge e) {
    e.fromNode().appendOutEdge(e);
    e.toNode().appendInEdge(e);
  }
Ground Truth: add an edge to the graph
CodeNN : sets a edge to to to to
HDeepcom : adds an edge edge the graph
Attnndgru : adds a graph edge to the graph
NCS :add a graph edge to this graph
Code2seq : appends the specified edge to the
HybridDrl: invoked when the event is
Astattdgru: add a new graph edge to the graph
CodeAstnn : adds a space edge to the graph
CAST  : add an edge to the graph
---------------------------------------
code
   public void addChild(TreeNode node) {
    if (leftChild == null) {
      leftChild = node;
    } else {
      // get to the last sibling
      TreeNode siblingNode = leftChild;
      while (siblingNode.rightSibling != null) {
        siblingNode = siblingNode.rightSibling;
      }
      siblingNode.rightSibling = node;
    }
    node.parent = this;
  }
Ground Truth: adds a child to this node
CodeNN : returns the node sibling node the node
HDeepcom : adds a node to the node
Attnndgru : adds a child node to this node
NCS :similar to the normal contract of
Code2seq : sets the parent of this node
HybridDrl: adds a child to the node
Astattdgru: adds a node to the tree
CodeAstnn : adds a child to the tree
CAST  : add a child to this node
---------------------------------------
code
   public boolean similar(Operand op) {
    if (op instanceof TypeOperand) {
      TypeOperand that = (TypeOperand) op;
      return type == that.type && typeRef == that.typeRef;
    } else {
      return false;
    }
  }
Ground Truth: are two operands semantically equivalent
CodeNN : auto this operation code instruction semantically
HDeepcom : check two operands operand this
Attnndgru : operands semantically ce
NCS :are two operands similar
Code2seq : checks if this operand is equal
HybridDrl: returns true if the given operand is
```

```
Astattdgru: returns true if the operand is similar to the given op
CodeAstnn : returns true if the given operand is a similar operation
CAST  : are two operands semantically equivalent
---------------------------------------
code
   public String toString() {
    String result = "ppc trap ";
    switch (value) {
      case EQUAL:
        return result + "==";
      case NOT_EQUAL:
        return result + "!=";
      case LESS:
        return result + "<";
      case LESS_EQUAL:
        return result + "<=";
      case GREATER:
        return result + ">";
      case GREATER_EQUAL:
        return result + ">=";
      case HIGHER:
        return result + ">U";
      case LOWER:
        return result + "<U";
      case HIGHER_EQUAL:
        return result + ">=U";
      case LOWER_EQUAL:
        return result + "<=U";
      case NOT_SAME:
        return result + "U!=";
      case ALWAYS:
        return result + "always";
    }
    return "UNKNOWN";
  }
Ground Truth: returns the string representation of this operand
CodeNN : returns case derby case the the equal than
HDeepcom : returns a string representation of this
Attnndgru : returns a string representation of this operator
NCS :return the string representation of this values value
Code2seq : returns a string representation of this
HybridDrl: returns a string representation of this object
Astattdgru: returns a string representation of this code tabular type code
CodeAstnn : returns a string representation of this object
CAST  : returns the string representation of this object
---------------------------------------
code
   public boolean similar(Operand op) {
    return (op instanceof TIBConstantOperand) && value == ((TIBConstantOperand) op).value;
  }
Ground Truth: are two operands semantically equivalent
CodeNN : returns whether this semantically unresolved semantically
HDeepcom : check two operands operand operands operands operands
Attnndgru : returns true if the operands of the operands
NCS :determines whether two operations are semantically equivalent
Code2seq : checks if the given operand is
HybridDrl: returns true if the given operand is
Astattdgru: compares this expression to another operand
CodeAstnn : returns true if the given operand is a similar constant
CAST  : are two operands semantically equivalent
---------------------------------------
```

```
code
   public boolean similar(Operand op) {
    return (op instanceof LongConstantOperand) && (value == ((LongConstantOperand) op).value);
  }
Ground Truth: are two operands semantically equivalent
CodeNN : returns whether this semantically unresolved semantically
HDeepcom : check two operands opcode operands operands operands
Attnndgru : returns true if the operand operator is semantically similar
NCS :are two longs semantically equivalent
Code2seq : checks if the given operand is
HybridDrl: returns true if the given operand is
Astattdgru: compares the given operand with this operand
CodeAstnn : returns true if the given operand is a constant
CAST  : are two operands semantically equivalent
---------------------------------------
code
   public boolean similar(Operand op) {
    return (op instanceof ObjectConstantOperand) && value.equals(((ObjectConstantOperand) op).value);
  }
Ground Truth: are two operands semantically equivalent
CodeNN : returns whether constant semantically
HDeepcom : check two operands operand operands operands
Attnndgru : returns true if the operands of the operands
NCS :are two operands similar
Code2seq : checks if the given operand is
HybridDrl: returns true if the given operand is
Astattdgru: compares this expression to the given operand
CodeAstnn : returns true if the given operand is a constant
CAST  : are two operands semantically equivalent
---------------------------------------
code
     public Instruction next() {
      if (labelInstruction != null) {
        Instruction temp = labelInstruction;
        labelInstruction = null;
        return temp;
      } else if ((implicitInstructions != null) && implicitInstructions.hasNext()) {
        return implicitInstructions.next();
      } else {
        return explicitInstructions.next();
      }
    }
Ground Truth: get the next instruction in the enumeration
CodeNN : returns the instruction instruction instruction instruction instruction instruction
HDeepcom : returns the next instruction
Attnndgru : returns the next instruction
NCS :returns the next
Code2seq : returns the next instruction
HybridDrl: returns the next instruction
Astattdgru: returns the next instruction
CodeAstnn : returns the next instruction in the stack
CAST  : get the next instruction in the iteration
---------------------------------------
code
   public String toString() {
    String exmsg = " (catches ";
    for (int i = 0; i < exceptionTypes.length - 1; i++) {
      exmsg = exmsg + exceptionTypes[i].toString() + ", ";
    }
    exmsg = exmsg + exceptionTypes[exceptionTypes.length - 1].toString();
    exmsg = exmsg + " for";
    BasicBlockEnumeration in = getIn();
    while (in.hasMoreElements()) {
      exmsg = exmsg + " " + in.next().toString();
    }
    exmsg = exmsg + ")";

    return super.toString() + exmsg;
  }
Ground Truth: return a string representation of the basic block
CodeNN : returns the string representation of this
HDeepcom : returns a string representation of the
Attnndgru : returns a string representation of this object
NCS :returns a string with the code basic block code and the
Code2seq : returns a string representation of the
HybridDrl: returns a string representation of this
Astattdgru: returns a string representation of this type
CodeAstnn : returns a string representation of this block
CAST  : returns a string representation of the basic block
---------------------------------------
code
   public void addAllEquationsToWorkList() {
    for (Enumeration<DF_Equation> e = getEquations(); e.hasMoreElements();) {
      DF_Equation eq = e.nextElement();
      addToWorkList(eq);
    }
  }
Ground Truth: add all equations to the work list
CodeNN : this the unit of to equations equations
HDeepcom : add all equations equations the list work
Attnndgru : adds all equations to the work list
NCS :adds all equations to the work list
Code2seq : this method is called by the
HybridDrl: sets the <unk>
Astattdgru: add all the equations to work list
CodeAstnn : adds all maneuver to the work list
CAST  : add all equations to the work list
---------------------------------------
code
   public void assertIsType(Operand op, TypeReference type) {
    if (VM.VerifyAssertions) {
      if (op.isDefinitelyNull()) {
        VM._assert(type.isReferenceType());
      } else if (op.isIntLike()) {
        VM._assert(type.isIntLikeType());
      } else {
        TypeReference type1 = op.getType();
        if (ClassLoaderProxy.includesType(type, type1) == NO) {
          VM._assert(false, op + ": " + type + " is not assignable with " + type1);
        }
      }
    }
  }
Ground Truth: assert that the given operand is of the given type or of
CodeNN : returns the is type is is type
HDeepcom : checks if type type is is is is
Attnndgru : checks if the given type is assignable from the operand
NCS :is the given operand a type safe or a reference
Code2seq : check if the type is a
HybridDrl: compares the given operand
Astattdgru: assert that the type is a type of the operands
CodeAstnn : asserts that the given operand is a type
```

```
CAST  : assert that the given operand is of the given type
---------------------------------------
code
   protected CompiledMethod genCode() throws VerifyError {
    if (VM.writingBootImage) {
      return BootImageCompiler.compile(this);
    } else {
      return RuntimeCompiler.compile(this);
    }
  }
Ground Truth: generate the code for this method
CodeNN : returns the image compile the the
HDeepcom : generate the code
Attnndgru : generates a compiled syntax instruction
NCS :generate the code
Code2seq : compile the given compiler
HybridDrl: generates a sequence of a
Astattdgru: returns a code compiled code object
CodeAstnn : generates a code for the given code
CAST  : generates the code for this method
---------------------------------------
code
   public Object addressAsObject(Address address) {
    VM.sysWriteln("BootImageObjectAddressRemapper: called addressAsObject");
    VM._assert(VM.NOT_REACHED);
    return null;
  }
Ground Truth: map an address to an object
CodeNN : returns the address address a the a
HDeepcom : converts the object as
Attnndgru : returns the object associated with the given address
NCS :an object of a type t
Code2seq : get the address of the specified
HybridDrl: returns the object of the given address
Astattdgru: returns an address object with the given address
CodeAstnn : returns the object as an object
CAST  : convert an address to an object
---------------------------------------
code
   public boolean handleRemsetAddress(Address addr) {
    if(subspace.addressInRange(addr)) {
      // increment tile
      int index = subspace.getIndex(addr);
      remsetStream.increment(index, (short)1);
      // increment summary
      totalRemset++;
      return true;
    } else {
      return false;
    }
  }
Ground Truth: handle a rem set address
CodeNN : returns the address address the the the
HDeepcom : perform a address set address the
Attnndgru : remove the set address from the address
NCS :add a address to a rem set
Code2seq : increment the value of the
HybridDrl: increments the given address
Astattdgru: handles rem set address
CodeAstnn : handle the rem set address
CAST  : handle a rem set address
```

```
---------------------------------------
code
   public void store(Word val) {
    SimulatedMemory.setWord(this, val.value);
  }
Ground Truth: stores the word value in the memory location pointed to by the
CodeNN : setter the value of the word
HDeepcom : store the word
Attnndgru : store the value of the word
NCS :stores the contents of the word in the memory location pointed to by
Code2seq : sets the value of the memory
HybridDrl: store a word
Astattdgru: stores the word value in the buffer
CodeAstnn : stores the given word in the memory
CAST  : stores the value in the memory location pointed to by the word
---------------------------------------
code
        protected String resolve(String $resource) {

                if ($resource.startsWith("/")) $resource = $resource.substring(1);
                $resource = $resource.replace('/', '.');
                String resource = null;
                String caller = getCaller();
                ByteCode callerCode = (ByteCode)byteCode.get(caller + ".class");

                if (callerCode != null) {
                        // Jar-local first, then global.
                        String tmp = callerCode.codebase + "/" + $resource;
                        if (byteCode.get(tmp) != null) {
                                resource = tmp;
                        }
                }
                if (resource == null) {
                        // One last try.
                        if (byteCode.get($resource) == null) {
                                resource = null;
                        } else {
                                resource = $resource;
                        }
                }
                VERBOSE("resource " + $resource + " resolved to " + resource);
                return resource;
        }
Ground Truth: resolve a resource name
CodeNN : this the resource code
HDeepcom : resolve the resource resource the the
Attnndgru : resolve a resource
NCS :for a resource name this method tries to locate
Code2seq : returns the resource name of the
HybridDrl: resolves the given name
Astattdgru: resolve a resource
CodeAstnn : resolves the given resource
CAST  : resolve a resource name
---------------------------------------
code
     public String execute(Object data) {
        console = DbConsole.getInstance();
        in = console.getIn();
        out = console.getOut();

        // store the current db info, just in case the user changes to a
        // different db
        saveCurrentDbInfo();

        setupDAO();
        return null;
    }
Ground Truth: execute the command
CodeNN : executes the the db
HDeepcom : execute the the
Attnndgru : executes the given command
NCS :execute the specified data
Code2seq : this method is called when the
HybridDrl: executes the given data
Astattdgru: executes the given object
CodeAstnn : execute the database
CAST  : execute the command
---------------------------------------
code
        public String getProxyUserPassword() {
                return proxyUserPassword;
        }
Ground Truth: gets the proxy user password
CodeNN : returns the password password
HDeepcom : returns the proxy user password
Attnndgru : gets the value of the proxy user password property
NCS :returns the proxy user password
Code2seq : returns the user password
HybridDrl: returns the value of the
Astattdgru: returns the proxy user password
CodeAstnn : returns the proxy user password
CAST  : gets the proxy user password
---------------------------------------
code
     public void setEasytomcatUpdates(EasytomcatUpdates value) {
        this.easytomcatUpdates = value;
    }
Ground Truth: sets the value of the easytomcat updates property
CodeNN : returns the value of updates
HDeepcom : sets the value of the updates property
Attnndgru : sets the value of the updates property
NCS :sets the value of the ) updates property
Code2seq : sets the value of the updates
HybridDrl: sets the value of the <unk> property
Astattdgru: sets the value of the updates property
CodeAstnn : sets the value of the <unk> updates property
CAST  : sets the value of the easytomcat updates property
---------------------------------------
code
        public void setTomcatOverviewPanel(TomcatOverviewPanel tomcatOverviewPanel) {
                this.tomcatOverviewPanel = tomcatOverviewPanel;
        }
Ground Truth: sets the tomcat overview panel
CodeNN : sets the panel the the the
HDeepcom : sets the panel panel panel
Attnndgru : sets the gi web service
NCS :setter for property tomcat overview panel
Code2seq : sets the overview panel
HybridDrl: sets the <unk>
Astattdgru: setter for property code
CodeAstnn : sets the dialog overview panel
CAST  : sets the tomcat overview panel
```

```
---------------------------------------
code
        public String toString() {
                final String TAB = "    ";

                StringBuffer retValue = new StringBuffer();

                retValue.append("Server (
").append(super.toString()).append(TAB).append("newShutdownPort =
").append(this.newShutdownPort).append(
                        TAB).append("shutdownPort =
").append(this.shutdownPort).append(TAB).append("isNewShutdownPort = ").append(
                        this.isNewShutdownPort).append(TAB).append("services =
").append(this.services).append(TAB).append(" )");

                return retValue.toString();
        }
Ground Truth: constructs a code string code with all attributes in name value
CodeNN : shutdown a tab representation of this tab
HDeepcom : constructs a string string representing a string
Attnndgru : returns a string representation of this object
NCS :retrieves a code string code representation of this
Code2seq : returns a string representation of this
HybridDrl: constructs a string representation of this attribute
Astattdgru: constructs a string with the name of the component
CodeAstnn : returns a string representation of this object
CAST  : constructs a code string code with all attributes in this
---------------------------------------
code
        public boolean isNewPort() {
                return isNewPort;
        }
Ground Truth: returns the is new port
CodeNN : determines if the port is is
HDeepcom : returns if the port is new new
Attnndgru : returns true if the port is new
NCS :returns whether this is a new port
Code2seq : returns true if the port is
HybridDrl: returns true if the
Astattdgru: returns the new port
CodeAstnn : returns true if the port is a new port
CAST  : returns the is new port
---------------------------------------
code
        public boolean isNewProtocol() {
                return isNewProtocol;
        }
Ground Truth: returns the is new protocol
CodeNN : determines if the protocol is is
HDeepcom : returns if the protocol is is
Attnndgru : returns true if the protocol is a new protocol
NCS :returns whether or not this is a new protocol
Code2seq : returns true if the user is
HybridDrl: returns true if the
Astattdgru: returns the new protocol
CodeAstnn : returns true if the protocol is a new protocol
CAST  : returns the is new protocol
---------------------------------------
code
        public boolean isClientAuth() {
                return clientAuth;
        }
Ground Truth: returns the client auth
CodeNN : returns whether client client is
HDeepcom : returns the client auth client
Attnndgru : returns true if the client is a client auth entification
NCS :returns true if the client authentication is enabled
Code2seq : returns true if the client is
HybridDrl: returns true if the
Astattdgru: returns the client authentication flag
CodeAstnn : returns true if the client is currently being used
CAST  : returns the client auth
---------------------------------------
code
        public boolean isNewRedirectPort() {
                return isNewRedirectPort;
        }
Ground Truth: returns the is new redirect port
CodeNN : returns the the client is redirected redirected
HDeepcom : returns whether the redirect is redirect redirect redirect
Attnndgru : returns true if the connection manager is new
NCS :returns whether or not the redirect port is a new one
Code2seq : returns the new redirect port
HybridDrl: returns true if the
Astattdgru: returns the new redirect port
CodeAstnn : returns true if the server is a new redirect port
CAST  : returns the is new redirect port
---------------------------------------
code
        public boolean isNewSslEnabled() {
                return newSslEnabled;
        }
Ground Truth: returns the new ssl enabled
CodeNN : returns the enabled enabled enabled enabled enabled enabled
HDeepcom : returns if the ssl enabled enabled enabled
Attnndgru : returns true if ssl is enabled for ssl
NCS :returns true if the socket should be used for new ssl
Code2seq : returns true if ssl is enabled
HybridDrl: returns true if the
Astattdgru: returns the new ssl enabled status
CodeAstnn : returns true if ssl is enabled
CAST  : returns the new ssl enabled
---------------------------------------
code
        public boolean isNewScheme() {
                return isNewScheme;
        }
Ground Truth: returns the is new scheme
CodeNN : determines the this this is is
HDeepcom : returns if the scheme is scheme new
Attnndgru : returns true if the current scheme is new
NCS :returns whether this is a new scheme
Code2seq : returns true if the new scheme
HybridDrl: returns true if the
Astattdgru: returns the new scheme
CodeAstnn : returns true if the user is a new scheme
CAST  : returns the is new scheme
---------------------------------------
code
        public boolean isNewSslProtocol() {
                return isNewSslProtocol;
        }
```

```
Ground Truth: returns the is new ssl protocol
CodeNN : determines if the protocol is is
HDeepcom : returns if this protocol protocol is ssl
Attnndgru : returns true if the protocol is a new ssl protocol
NCS :returns true if this is a new ssl protocol
Code2seq : returns true if the connection is
HybridDrl: returns true if the
Astattdgru: returns the new ssl protocol
CodeAstnn : returns true if ssl protocol is set
CAST  : returns the is new ssl protocol
---------------------------------------
```

100 Examples in TL-CodeSum

```
---------------------------------------
code
 private int currentDepth(){
  try {
    Integer oneBased=((Integer)DEPTH_FIELD.get(this));
    return oneBased - 1;
  }
 catch (  IllegalAccessException e) {
    throw new AssertionError(e);
  }
}
Ground Truth: returns a 0 - based depth within the object graph of the current object being serialized
.
CodeNN : returns a , object based depth in an in to the the the object being . .
HDeepcom : returns a - of depth the the current current current current .
Attgru : returns a 0 - based depth as intended
NCS :returns a 0 - based depth within the object graph graph of the object being serialized .
Code2seq : returns the next value of the
HybridDrl: returns the number of milliseconds for the current thread
Astattgru: returns the ( 5 ) time , or more alternative base object
CodeAstnn: returns the 0 - based depth within the object graph depth .
CAST  : returns a 0 - based depth within the object graph of the current object being serialized .
---------------------------------------
code
 protected void requestPassCodeConfirmation(){
  clearBoxes();
  mPassCodeHdr.setText(R.string.pass_code_reenter_your_pass_code);
  mPassCodeHdrExplanation.setVisibility(View.INVISIBLE);
  mConfirmingPassCode=true;
}
Ground Truth: ask to the user for retyping the pass code just entered before saving it as the current
pass code .
CodeNN : ask to the user for the pass code code entered before saving it as the current pass code
HDeepcom : ask to the user for the the . the pass pass pass pass the pass the the the the code . code
. . code the the the code the the the . . . . . . . .
Attgru : ask to the user for the pass code just entered before saving it as the current pass code .
NCS :ask to the user for retyping the current pass code as the user entered for saving code .
Code2seq : resets the default view the
HybridDrl: called when the user view view view is view
Astattgru: ask to the user about to the user to allow for more control to stop
CodeAstnn: ask to the pass code just entered before saving code just for retyping the pass code as the
user code just entered by the user code
CAST  : ask to the user for retyping the pass code just entered before saving it as the current pass
code .
---------------------------------------
code
 public void addRestrictedDomain(String domainName){
  if (StringUtils.isEmpty(domainName)) {
    return;
  }
  if (restrictedDomains == null) {
    restrictedDomains=new ArrayList<>();
  }
  restrictedDomains.add(domainName);
}
Ground Truth: adds specified domain name to the list of restricted domains
CodeNN : adds specified qname expression scope the tail those global domains is is is is child if if -
.
HDeepcom : adds full to domain to the generated . . . this this generated generated generated
generated
Attgru : adds an exit language .
```

```
NCS :add a restricted name to the list of restricted domains
Code2seq : adds a name name to the
HybridDrl: adds the specified domain to the list
Astattgru: adds domain name which maps domain to
CodeAstnn: add specified domain name to the list of permitted domains
CAST  : adds specified domain name to the list of restricted domains
---------------------------------------
code
 public void writeNoScale(Image c,Component i){
  if (page == null) {
    newpage();
  }
  int x=x0 + width - (c.getWidth(null) + charwidth);
  int y=y0 + (linenum * lineheight) + lineascent;
  if (page != null && pagenum >= prFirst) {
    page.drawImage(c,x,y,c.getWidth(null),c.getHeight(null),null);
  }
}
Ground Truth: write a graphic to the printout . < p > this was not in the original class , but was
added afterwards by kevin dickerson . it is a copy of the write , but without the scaling . < p > the
image is positioned on the right side of the paper , at the current height .
CodeNN : draws a graphic so to printout color
HDeepcom : writes the rgb of of rgb value image image rgb image
Attgru : this method is called by the native object directly , as well as scheme , appends at or equal
to .
NCS :in the requirements of a line , indefinitely the image out to the printout . < p > this was not
in this method , but was added afterwards by dennis miller miller . < p > should exactly the image is
positioned on the right side of the paper
Code2seq : draws a new image
HybridDrl: calculate the background of the given component
Astattgru: draw a from
CodeAstnn: add a graphic to the workspace . < p > simultaneously was not in the original class , but
was added afterwards by bob jacobsen . modified by d miller . < p > the image is positioned on the
right side of the paper , at the current height
CAST  : write a graphic to the printout . < p > this was not in the original class , but was added
afterwards by bob jacobsen . modified by d miller . < p > the image is positioned on the right side of
the paper , at the current height
---------------------------------------
code
 protected void create_y_axis(int i){
  Log.e("index vlaue",i + "");
  float ver_height=((graphheight / label_size) * i) + border;
  if (i == largest_value_set.size() - 1)   canvas.drawLine(horstart,ver_height,width -
(border),ver_height,paint);
 else   canvas.drawLine(horstart,ver_height,border,ver_height,paint);
  paint.setColor(Color.BLACK);
  int Y_labels=(int)size - 1 - i;
  String y_labels=String.format("%.1f",Y_labels * y_axis_ratio);
  paint.setTextAlign(Paint.Align.RIGHT);
  canvas.drawText(y_labels,horstart - 15,ver_height - 10,paint);
  paint.setTextAlign(Paint.Align.LEFT);
}
Ground Truth: this function is used to plot the y axis and its breakdown with value . uses drawline (
) method to create the y axis and lines to show breakdown of y axis and drawtext ( ) method to plot
labels related with breakdowns
CodeNN : draws the plots scaled ' plot label y axis . . labels given axis axis axis - -
HDeepcom : fills method the the cell the vertical vertical for size for . . . . to to to to to the the
the the and and
Attgru : process a small color actions on the left side of the disable of the disable labels
NCS :this function is used to create labels and lines with value
Code2seq : draws the text for the given
```

HybridDrl: this function is used to create a component
Astattgru: create the plot axis and .
CodeAstnn: this function is used to plot the y axis and drawtext and drawtext ( ) method to create breakdown with y axis and drawtext ( )
CAST : this function is used to plot the y axis and its breakdown with value . uses drawline ( ) method to create the y axis and lines to show breakdown of y axis and drawtext ( ) method to plot labels related with breakdowns

----

code

```
 private FileChannel updateCache(String filePath){
  FileChannel fileChannel=fileNameAndStreamCache.get(filePath);
  try {
    if (null == fileChannel) {
      FileInputStream stream=new FileInputStream(filePath);
      fileChannel=stream.getChannel();
      fileNameAndStreamCache.put(filePath,fileChannel);
    }
  }
 catch (  IOException e) {
    LOGGER.error(e,e.getMessage());
  }
  return fileChannel;
}
```

Ground Truth: this method will be used to check whether stream is already present in cache or not for filepath if not present then create it and then add to cache , other wise get from cache
CodeNN : creates method cache the a to entity is corresponding to to the the which in we working filepath cache cache . . . if will will will will cache cache wise
HDeepcom : this method will ' used it it it not not not not present present not not not not cache is is is is is is
Attgru : this function will read the cache into a file that reads the head of the quorum .
NCS :this method will be used to check whether cache is already present in cache or not for filepath if not present then create it and then add to cache
Code2seq : saves the contents of the file
HybridDrl: this method will be used to read a stream of file
Astattgru: this method will be used to check for file channel with cache
CodeAstnn: this method will be used to check whether stream is already present in cache or not for filepath if not present then create new one
CAST : this method will be used to check whether stream is already present in cache or not for filepath if not present then create it and then add to cache , other wise get from cache

----

code

```
 private float calculateSelectorRadius(){
  if (mfSelectorRadius == 0.0f) {
    int nHeight=mSelectedView.getHeight();
    int nWidth=mSelectedView.getWidth();
    int nLargerSide=Math.max(nWidth,nHeight);
    mfSelectorRadius=((nLargerSide ^ 2) / 2);
  }
  return mfSelectorRadius;
}
```

Ground Truth: calculate selector radius the selector radius can be calculated by a simple math calculation of the larger side .
CodeNN : calculate selector radius of selector radius on be calculated . . . math calculation the the larger side side
HDeepcom : calculate selector radius on selector radius can can calculated calculated on calculation on
Attgru : calculate selector radius the selector radius can be calculated by a simple screen .
NCS :calculates a simple math math radius radius can be calculated by a math math .
Code2seq : calculate the normalized scale of the
HybridDrl: calculate the current state of the current state

```
Astattgru: calculate selector radius the selector radius can be calculated by a simple math
calculation of the larger than the .
CodeAstnn: calculate selector radius the simple selector radius can be calculated by a simple math of
the larger side of the larger selector .
CAST  : calculate selector radius the selector radius can be calculated by a simple math calculation
of the larger side .
---------------------------------------
code
 public static void slideInFromTopAnimator(@NonNull List<Animator> animators,@NonNull View
view,RecyclerView recyclerView){
  alphaAnimator(animators,view,0f);
  animators.add(ObjectAnimator.ofFloat(view,"translationY",-recyclerView.getMeasuredHeight() >> 1,0));
  if (FlexibleAdapter.DEBUG)   Log.v(TAG,"Added TOP Animator");
}
Ground Truth: item will slide from top of the screen to its natural position .
CodeNN : report will slide representation bottom of the screen to . natural . . the the the . the
HDeepcom : this the in the triangle the the the to to . . . . keep .
Attgru : this method is called from ( ) when ( int ) , this will only called at debug ( ) , even if
the view is not null !
NCS :slide the item from the top of the screen to the screen . < p / > this works around the item
position .
Code2seq : add a view to the end
HybridDrl: item moves a view to starts the animations that will be drawn
Astattgru: keeps the specified view with the given ordering . < p > < p > < p > < p > < p > < p > < p
> < p > < p > < p > < p > < p > < p > <
CodeAstnn: item will slide to the natural position of the screen .
CAST  : item will slide from bottom of the screen to its natural position .
---------------------------------------
code
 @RequestMapping(value="/SAML2/IDPSSO/{tenant:.*}") public void ssoBindingError(Locale
locale,@PathVariable(value="tenant") String tenant,HttpServletResponse response) throws IOException {
  logger.info("IDP SSO binding error! The client locale is " + locale.toString() + ", tenant is "+
tenant);
  ssoDefaultTenantError(locale,response);
}
Ground Truth: handle request sent with a wrong binding
CodeNN : handle a a for . wrong binding element . specified to the is request request .
HDeepcom : handle request counter seen binding binding binding binding invalid invalid invalid
Attgru : handle the definition function details
NCS :handle request sent with the default binding
Code2seq : this method calls the request to
HybridDrl: handle a new entry to the request
Astattgru: grab display about a new collection and its causes when needed .
CodeAstnn: handle request sent with default tenant
CAST  : handle request sent with a wrong binding
---------------------------------------
code
 private void paintForegroundEnabled(Graphics2D g,int width,int height){
  Shape s=decodeArrowPath(width,height);
  g.setPaint(enabledColor);
  g.fill(s);
}
Ground Truth: paint the arrow in enabled state .
CodeNN : paint the for image a s a an color of image a a color color color color centered color color
color .
HDeepcom : paint the arrow component enabled . black black background black a . .
Attgru : paint button in a paint with a frequency button .
NCS :paint the arrow in the list of arrow state .
Code2seq : draws the given shape
HybridDrl: paint the state of the view
Astattgru: paint the foreground or disabled labels .
```

```
CodeAstnn: paint the arrow in pressed state .
CAST  : paint the arrow in enabled state .
---------------------------------------
code
 public void add(int position,@NonNull T item){
synchronized (lock) {
    if (items == null) {
      items=new ArrayList<>();
    }
    items.add(position,item);
  }
  if (notifyOnChange) {
    notifyItemInserted(position);
  }
}
Ground Truth: adds the specified item to the list with the specified position
CodeNN : add the item with the the end inserts the the list list
HDeepcom : adds a given item to the adapter of the the .
Attgru : insert the item at the position if currently set .
NCS :inserts the specified item at the specified position in the list .
Code2seq : adds a new item to the
HybridDrl: adds a new item to the list
Astattgru: adds an item to the history . if the list is empty , it will be sent to the market .
CodeAstnn: inserts the specified item at the specified position of the list .
CAST  : adds the specified item to the array ' s list .
---------------------------------------
code
 public boolean verifyServerEvidenceMessage(BigInteger serverM2) throws CryptoException {
  if ((this.A == null) || (this.M1 == null) || (this.S == null)) {
    throw new CryptoException("Impossible to compute and verify M2: " + "some data are missing from
the previous operations (A,M1,S)");
  }
  BigInteger computedM2=SRP6Util.calculateM2(digest,N,A,M1,S);
  if (computedM2.equals(serverM2)) {
    this.M2=serverM2;
    return true;
  }
  return false;
}
Ground Truth: authenticates the server evidence message m2 received and saves it only if correct .
CodeNN : creates the new test define
HDeepcom : authenticates the new message as m1 the the : : : : : : messages : : : before before .
Attgru : verify the client evidence server against an ip .
NCS :authenticates the received client message m1 and saves it .
Code2seq : check if the server is a
HybridDrl: returns true if the key is a new
Astattgru: authenticates to server in a .
CodeAstnn: here we preserve the 2063 message and server doclet it .
CAST  : checks the received server evidence message m1 and saves it only if correct .
---------------------------------------
code
 public Metadata.Property.Builder clear(){
  Metadata_Property_Builder _defaults=new Metadata.Property.Builder();
  type=_defaults.type;
  boxedType=_defaults.boxedType;
  name=_defaults.name;
  capitalizedName=_defaults.capitalizedName;
  allCapsName=_defaults.allCapsName;
  getterName=_defaults.getterName;
  codeGenerator=_defaults.codeGenerator;
  fullyCheckedCast=_defaults.fullyCheckedCast;
  accessorAnnotations.clear();
  _unsetProperties.clear();
  _unsetProperties.addAll(_defaults._unsetProperties);
  return (Metadata.Property.Builder)this;
}
Ground Truth: resets the state of this builder .
CodeNN : checks the space properties a the the apk hierarchy . together validate name address the the
the of of of of of of of media media media media replacement replacement . . . . . cause cause cause
cause cause cause cause cause cause of of of of of
HDeepcom : resets the node from from removes . .
Attgru : remove the objects for the given task types .
NCS :clears the state of this builder , allowing a mysql state .
Code2seq : removes all the elements from the
HybridDrl: removes all the values from the list
Astattgru: allow the to provide the specified observe state .
CodeAstnn: resets the state of the given formatting .
CAST  : resets the state of this builder .
---------------------------------------
code
 private void loadWorkflow(Properties ctx,int AD_Workflow_ID,HttpSession sess){
  MWorkflow wf=new MWorkflow(ctx,AD_Workflow_ID,null);
  MWFNode[] nodes=wf.getNodes(true,Env.getContextAsInt(ctx,"#AD_Client_ID"));
  MWFNode wfn=null;
  ArrayList nodes_ID=new ArrayList();
  for (int i=0; i < nodes.length; i++) {
    wfn=nodes[i];
    nodes_ID.add(new Integer(wfn.getAD_WF_Node_ID()));
  }
  int imageMap[][]=generateImageMap(nodes_ID);
  sess.setAttribute(WORKFLOW,wf);
  sess.setAttribute(NODES,nodes);
  sess.setAttribute(NODES_ID,nodes_ID);
  sess.setAttribute(IMAGE_MAP,imageMap);
  sess.setAttribute(ACTIVE_NODE,new Integer(-999));
}
Ground Truth: load workflw and initialize the session attributes .
CodeNN : load / and . the session . .
HDeepcom : load the initialize the session attributes attributes
Attgru : load from a workflow after a background method
NCS :load workflw and initialize workflw and add workflw
Code2seq : creates a new instance of the
HybridDrl: compute the <unk>
Astattgru: load and initialize the session attributes .
CodeAstnn: load workflw workflw session attributes and initialize the workflw remote session
attributes .
CAST  : load workflw and initialize the session attributes .
---------------------------------------
code
 public void hspan(double start,double end,Paint color,String legend){
  LegendText legendText=new LegendText(color,legend);
  comments.add(legendText);
  plotElements.add(new HSpan(start,end,color,legendText));
}
Ground Truth: draws a horizontal span into the graph and optionally adds a legend .
CodeNN : add the statistics rule of namespace the the optionally horizontal . starting starting
starting the
HDeepcom : draws a single column the given . partition partition lines lines lines
Attgru : generates text for a symbol with a color
NCS :draws a vertical rule into the legend .
Code2seq : add a new text to the
HybridDrl: start a new <unk>
```

Astattgru: draws a line of range .

CodeAstnn: draws a horizontal color .

CAST : draws a horizontal rule into the graph and optionally adds a legend .

----

code

```
 @Override public void addRelations(Task task,Iterable<ObjectId> projectIds,String fieldName){
  List<Project> newProjectList=new LinkedList<>();
  Iterable<Project> projectsToAdd=projectRepository.findAll(projectIds,null);
  for (  Project project : projectsToAdd) {
    newProjectList.add(project);
  }
  try {
    if (PropertyUtils.getProperty(task,fieldName) != null) {
      Iterable<Project> projects=(Iterable<Project>)PropertyUtils.getProperty(task,fieldName);
      for (      Project project : projects) {
        newProjectList.add(project);
      }
    }
  }
 catch (  Exception e) {
    throw new RuntimeException(e);
  }
  try {
    PropertyUtils.setProperty(task,fieldName,newProjectList);
  }
 catch (  Exception e) {
    throw new RuntimeException(e);
  }
  taskRepository.save(task);
}
```

Ground Truth: a simple implementation of the addrelations method which presents the general concept of the method . it should not be used in production because of possible race condition - production ready code should perform an atomic operation .

CodeNN : pull simple project into project project project project project project

HDeepcom : add a default of to a service . . . . the a a a . . . . to to were were were were were were were were . . . . .

Attgru : adds the specified bean to the choices , populating unused information known for the associated tag .

NCS :a simple implementation of the addrelations method should perform an atomic projects . it should not be used in production because the general - production production code should perform any of the general method .

Code2seq : adds a list of the given

HybridDrl: check if the task is a new

Astattgru: implementation of the method ' s class implementation .

CodeAstnn: a utility method which allows the general project of the addrelations - concept method where addrelations general general support for the general project .

CAST : a simple implementation of the addrelations method which presents the general concept of the method . it should not be used in production because of possible race condition - production ready code should perform an atomic operation .

----

code

```
 private void stretchViewVertically(View view,int crossSize){
  LayoutParams lp=(LayoutParams)view.getLayoutParams();
  int newHeight=crossSize - lp.topMargin - lp.bottomMargin;
  newHeight=Math.max(newHeight,0);

view.measure(MeasureSpec.makeMeasureSpec(view.getMeasuredWidth(),MeasureSpec.EXACTLY),MeasureSpec.makeMeasureSpec(newHeight,MeasureSpec.EXACTLY));
}
```

Ground Truth: expand the view vertically to the size of the crosssize (considering the view margins)

```
CodeNN : stores the progress well to year height in on held fit ' ui view margins the
HDeepcom : called the the given the the size the the the the the
Attgru : expand a horizontally and a view to increase the initial width and calculating the bounding view should have higher
NCS :expand the view horizontally to the size of the crosssize that will be incremented
Code2seq : calculates the view for the view
HybridDrl: expand a view to the text area
Astattgru: the method where the original background size is generated at all on scrolling .
CodeAstnn: expand the view horizontally
CAST  : expand the view horizontally to the size of the crosssize ( considering the view margins )
---------------------------------------
code
 private Collection<Var> migrateColumns(SQLTable currentTable){
  List<Var> vars=new ArrayList<>();
  String tableType=currentTable.getEntityType();
  Map<String,ResourceType.DataType> columns=currentTable.getColumns();
  Map<String,String> foreignColumns=currentTable.getForeignKeyColumns();
  for (  String column : columns.keySet()) {
    ResourceType.DataType columnType=columns.get(column);
    if (foreignColumns.containsKey(column)) {
      vars.addAll(migrateAsRelation(tableType,column,foreignColumns.get(column)));
    }
 else {
      vars.addAll(migrateAsResource(tableType,columnType,column));
    }
  }
  return vars;
}
Ground Truth: loop through each of the columns in the given table , migrating each as a resource or relation .
CodeNN : returns a foreign - of columns the
HDeepcom : loop total columns of of column the the , , , column having . column .
Attgru : executes the function , including the java attributes from all columns rules found ( emoji - i - 1 ) where this query should be used to pick all edges to sparse data
NCS :migrate the columns of a table in the given resource query .
Code2seq : returns the list of the given
HybridDrl: finds a set of rows in a table
Astattgru: maps a list to the table
CodeAstnn: determines if the columns have tender_no , or resource as the columns in each resource , columns are schedulerstatemanageradaptor .
CAST  : loop through each of the columns in the table , migrating each as a resource or relation .
---------------------------------------
code
 protected Workflow.Method createRemoveVolumesFromCGMethod(URI vplexURI,URI cgURI,List<URI> vplexVolumeURIs){
  return new Workflow.Method(REMOVE_VOLUMES_FROM_CG_STEP,vplexURI,cgURI,vplexVolumeURIs);
}
Ground Truth: a method that creates the workflow method for removing vplex volumes from a consistency group .
CodeNN : a method object the the the the remain vplex creates the
HDeepcom : this method provided creates the method method consistency consistency consistency consistency consistency
Attgru : create project for workflow method
NCS :a method the creates the workflow method for adding a vplex vplex volumes to a vplex consistency group .
Code2seq : remove a workflow from the given
HybridDrl: a method to add the given method
Astattgru: returns the new vplex storage by the passed path the vplex connection .
CodeAstnn: a method the creates the workflow method for adding vplex volumes to a vplex consistency group natively volumes .
CAST  : a method that creates the workflow method for adding vplex volumes to a consistency group .
```

```
---------------------------------------
code
 public synchronized String generateCallIdentifier(String address){
  String date=Long.toString(System.currentTimeMillis() + callIDCounter++ + rand.nextLong());
  byte cid[]=digester.digest(date.getBytes());
  String cidString=Utils.toHexString(cid);
  return cidString + "@" + address;
}
Ground Truth: generate a call identifier . this is useful when we want to generate a call identifier
in advance of generating a message .
CodeNN : generates a string made of a a . url expect have the call a a call if in advance a a a a a
sorts sorts sorts sorts sorts a a
HDeepcom : push an returns with server the the the the the . . .
Attgru : try to generate parse string identifier .
NCS :generate a unique identifier .
Code2seq : returns a string representation of the
HybridDrl: generates a string representation of the address
Astattgru: generate identifier with random certificate
CodeAstnn: generate a call in this to generate a summary string in tail .
CAST  : generate a call identifier . this is useful when we want to generate a call identifier in
advance of generating a message .
---------------------------------------
code
 public void move(MouseEvent e){
  Point2D pnt=getProjectionPoint(e);
  int x=(int)pnt.getX();
  int y=(int)pnt.getY();
  if (poly.getRenderType() == OMGraphic.RENDERTYPE_OFFSET) {
    gpm=new OffsetGrabPoint(x,y);
    gpm.clear();
  }
 else {
    gpm=gpo;
    gpm.clear();
    gpm.set(x,y);
  }
  addPolyGrabPointsToOGP(gpm);
  movingPoint=gpm;
}
Ground Truth: called to set the offsetgrabpoint to the current mouse location , and update the
offsetgrabpoint with all the other grabpoint locations , so everything can shift smoothly . should
also set the offsetgrabpoint to the movingpoint . should be called only once at the beginning of the
general movement , in order to set the movingpoint . after that , redraw ( e ) should just be called ,
and the movingpoint will make the adjustments to the graphic that are needed .
CodeNN : moves to offsetgrabpoint forward given forward on . they they the the the the the the times
times times place place place times times times times times times times vertices vertices vertices
vertices vertices vertices vertices . .
HDeepcom : called to set on offsetgrabpoint , , mouse mouse mouse mouse , . . chart chart chart .
Attgru : invoked when the mouse button is dropped .
NCS :called from the offsetgrabpoint to the offsetgrabpoint , and update the offsetgrabpoint location
, so that the offsetgrabpoint menu item should be the first mouse , and update the offsetgrabpoint
with the offsetgrabpoint . should also set the offsetgrabpoint to the movingpoint thread . should be
called only when the
Code2seq : this method called to called when
HybridDrl: moves the mouse event to the location
Astattgru: move mouse mouse left to right . this moves the graphic to the left of the thumb relative
to the origin of the 3d area by the x and y set of the top left and ending point . < p / > this is not
guaranteed to go
CodeAstnn: called to set the offsetgrabpoint to the offsetgrabpoint , so it can shift the
offsetgrabpoint of the offsetgrabpoint . should be called at the current location . once the
offsetgrabpoint is first . if the movingpoint is going to set the offsetgrabpoint is locations , the
```

current mouse is removed
CAST : called to set the offsetgrabpoint to the current mouse location , and update the offsetgrabpoint with all the other grabpoint locations , so everything can shift smoothly . should also set the offsetgrabpoint to the movingpoint . should be called only once at the beginning of the general movement ,

----

code

```
 @Override public SSLEngineResult wrap(ByteBuffer[] srcs,int offset,int len,ByteBuffer dst) throws
SSLException {
  if (engine_was_shutteddown) {
    return new
SSLEngineResult(SSLEngineResult.Status.CLOSED,SSLEngineResult.HandshakeStatus.NOT_HANDSHAKING,0,0);
  }
  if ((srcs == null) || (dst == null)) {
    throw new IllegalStateException("Some of the input parameters are null");
  }
  if (dst.isReadOnly()) {
    throw new ReadOnlyBufferException();
  }
  if (!handshake_started) {
    beginHandshake();
  }
  SSLEngineResult.HandshakeStatus handshakeStatus=getHandshakeStatus();
  if ((session == null || engine_was_closed) &&
(handshakeStatus.equals(SSLEngineResult.HandshakeStatus.NEED_UNWRAP) ||
handshakeStatus.equals(SSLEngineResult.HandshakeStatus.NEED_TASK))) {
    return new SSLEngineResult(getEngineStatus(),handshakeStatus,0,0);
  }
  int capacity=dst.remaining();
  int produced=0;
  if (alertProtocol.hasAlert()) {
    if (capacity < recordProtocol.getRecordSize(2)) {
      return new SSLEngineResult(SSLEngineResult.Status.BUFFER_OVERFLOW,handshakeStatus,0,0);
    }
    byte[] alert_data=alertProtocol.wrap();
    dst.put(alert_data);
    if (alertProtocol.isFatalAlert()) {
      alertProtocol.setProcessed();
      if (session != null) {
        session.invalidate();
      }
      shutdown();
      return new
SSLEngineResult(SSLEngineResult.Status.CLOSED,SSLEngineResult.HandshakeStatus.NOT_HANDSHAKING,0,alert_
data.length);
    }
 else {
      alertProtocol.setProcessed();
      if (close_notify_was_sent && close_notify_was_received) {
        shutdown();
        return new
SSLEngineResult(SSLEngineResult.Status.CLOSED,SSLEngineResult.HandshakeStatus.NOT_HANDSHAKING,0,alert_
data.length);
      }
      return new SSLEngineResult(getEngineStatus(),getHandshakeStatus(),0,alert_data.length);
    }
  }
  if (capacity < recordProtocol.getMinRecordSize()) {
    if (logger != null) {
      logger.println("Capacity of the destination(" + capacity + ") < MIN_PACKET_SIZE("+
recordProtocol.getMinRecordSize()+ ")");
```

```
  }
  return new SSLEngineResult(SSLEngineResult.Status.BUFFER_OVERFLOW,handshakeStatus,0,0);
 }
 try {
  if (!handshakeStatus.equals(SSLEngineResult.HandshakeStatus.NEED_WRAP)) {
   dataStream.setSourceBuffers(srcs,offset,len);
   if ((capacity < SSLRecordProtocol.MAX_SSL_PACKET_SIZE) && (capacity <
recordProtocol.getRecordSize(dataStream.available()))) {
    if (logger != null) {
     logger.println("The destination buffer(" + capacity + ") can not take the resulting
packet("+ recordProtocol.getRecordSize(dataStream.available())+ ")");
    }
    return new SSLEngineResult(SSLEngineResult.Status.BUFFER_OVERFLOW,handshakeStatus,0,0);
   }
   if (remaining_wrapped_data == null) {
    remaining_wrapped_data=recordProtocol.wrap(ContentType.APPLICATION_DATA,dataStream);
   }
   if (capacity < remaining_wrapped_data.length) {
    return new
SSLEngineResult(SSLEngineResult.Status.BUFFER_OVERFLOW,handshakeStatus,dataStream.consumed(),0);
   }
 else {
    dst.put(remaining_wrapped_data);
    produced=remaining_wrapped_data.length;
    remaining_wrapped_data=null;
    return new SSLEngineResult(getEngineStatus(),handshakeStatus,dataStream.consumed(),produced);
   }
  }
 else {
   if (remaining_hsh_data == null) {
    remaining_hsh_data=handshakeProtocol.wrap();
   }
   if (capacity < remaining_hsh_data.length) {
    return new SSLEngineResult(SSLEngineResult.Status.BUFFER_OVERFLOW,handshakeStatus,0,0);
   }
 else {
    dst.put(remaining_hsh_data);
    produced=remaining_hsh_data.length;
    remaining_hsh_data=null;
    handshakeStatus=handshakeProtocol.getStatus();
    if (handshakeStatus.equals(SSLEngineResult.HandshakeStatus.FINISHED)) {
     session=recordProtocol.getSession();
    }
   }
   return new SSLEngineResult(getEngineStatus(),getHandshakeStatus(),0,produced);
  }
 }
 catch (  AlertException e) {
  alertProtocol.alert(AlertProtocol.FATAL,e.getDescriptionCode());
  engine_was_closed=true;
  if (session != null) {
   session.invalidate();
  }
  throw e.getReason();
 }
}
```

Ground Truth: encodes the application data into ssl / tls record . if handshake status of the engine differs from not_handshaking the operation can work without consuming of the source data . for more information about tls record fragmentation see tls v 1 specification (http : / / www . ietf . org / rfc / rfc2246 . txt) p 6 . 2 .

CodeNN : encodes the application data engine ssl the tls language view data handshake upload stream .

. differs differs the the ' ' work ' work work data data data data . tls tls tls tls tls tls tls tls / / / / / / / / / www

HDeepcom : load handshake offset data table ssl data tls the the the saves saves the the the the the the the the the the the the the the the the the the the the the the the the record record record record record record record record record record record

Attgru : encodes the application data - server

NCS :encodes the application data into ssl / tls record . tls tls the operation for ssl ssl data . tls the operation of ssl can work be ssl .

Code2seq : returns the next net engine and

HybridDrl: reads the ssl socket to an address

Astattgru: encodes the application data into ssl / tls record for completion .

CodeAstnn: encodes the application data into ssl / tls record . if handshake status of the engine differs from not_handshaking the application of the source / tls record . for more information about tls record (http : / / www . w3 . com / tls / tls / tls

CAST : encodes the application data into ssl / tls record . if handshake status of the engine differs from not_handshaking the operation can work without consuming of the source data . for more information about tls record fragmentation see tls v 1 specification (http : / / www . ietf

----

code

```
 public static ParsedSql parseSqlStatement(String sql){
  Set<String> namedParameters=new HashSet<String>();
  ParsedSql parsedSql=new ParsedSql(sql);
  char[] statement=sql.toCharArray();
  int namedParameterCount=0;
  int unnamedParameterCount=0;
  int totalParameterCount=0;
  int i=0;
  while (i < statement.length) {
    int skipToPosition=skipCommentsAndQuotes(statement,i);
    if (i != skipToPosition) {
      if (skipToPosition >= statement.length) {
        break;
      }
      i=skipToPosition;
    }
    char c=statement[i];
    if (c == ':' || c == '&') {
      int j=i + 1;
      if (j < statement.length && statement[j] == ':' && c == ':') {
        i=i + 2;
        continue;
      }
      while (j < statement.length && !isParameterSeparator(statement[j])) {
        j++;
      }
      if (j - i > 1) {
        String parameter=sql.substring(i + 1,j);
        if (!namedParameters.contains(parameter)) {
          namedParameters.add(parameter);
          namedParameterCount++;
        }
        parsedSql.addNamedParameter(parameter,i,j);
        totalParameterCount++;
      }
      i=j - 1;
    }
 else {
      if (c == '?') {
        unnamedParameterCount++;
        totalParameterCount++;
      }
    }
    i++;
  }
  parsedSql.setNamedParameterCount(namedParameterCount);
  parsedSql.setUnnamedParameterCount(unnamedParameterCount);
  parsedSql.setTotalParameterCount(totalParameterCount);
  return parsedSql;
}
Ground Truth: parse the sql statement and locate any placeholders or named parameters . named
parameters are substituted for a jdbc placeholder .
CodeNN : gets the sql statement and locate single placeholders , named parameters parameters
HDeepcom : parse placeholder sql statement for the given parameters a
Attgru : parse the request against the sql query string .
NCS :parse the sql statement and locate any placeholders or named parameters . named parameters are
substituted for a jdbc placeholder and an accompanying .
Code2seq : returns a new string for the
HybridDrl: returns a string representation of the sql statement
Astattgru: parse the sql statement and locate random information string .
CodeAstnn: locate any placeholders or jdbc parameters named placeholders and locate names of jdbc
placeholders or named parameters . named and locate any placeholders are jdbc .
CAST  : parse the sql statement and locate any placeholders or named parameters . named parameters are
substituted for a jdbc placeholder .
---------------------------------------
code
 public static boolean isAutoNew(Properties ctx,int WindowNo){
  if (ctx == null)   throw new IllegalArgumentException("Require Context");
  String s=getContext(ctx,WindowNo,"AutoNew",false);
  if (s != null) {
    if (s.equals("Y"))     return true;
 else     return false;
  }
  return isAutoNew(ctx);
}
Ground Truth: is window auto new record ( if not set use default )
CodeNN : locates function autocommit ( is is
HDeepcom : determine . . . . . .
Attgru : determines whether the stack is flag for the current window .
NCS :is window new window
Code2seq : checks if the given resource is
HybridDrl: check if the given profile is installed
Astattgru: checks whether the given configuration is available
CodeAstnn: is new program
CAST  : is window autocommit ( if not set use default )
---------------------------------------
code
 public void addPutTimeNanos(long duration){
  putTimeNanos.addAndGet(duration);
  if (delegate != null)   delegate.addPutTimeNanos(duration);
}
Ground Truth: increments the put time accumulator .
CodeNN : creates a data time accumulator
HDeepcom : add the . .
Attgru : adds a delay of the to this instance which is not changed .
NCS :increments the clock time accumulator .
Code2seq : adds a new time to the
HybridDrl: increments the value of the given duration
Astattgru: add time in seconds to be automatically run later instances
CodeAstnn: increments remove time accumulator .
CAST  : increments the put time accumulator .
---------------------------------------
code
```

```
 private static String padRight(String s,int minLength){
  return String.format("%1$-" + minLength + "s",s);
}
Ground Truth: adds padding to the end of a string with a repeating character if the string ' s length
is less than the minimum length
CodeNN : converts padding leading edge value of string length of length repeating length length length
string the the
HDeepcom : adds the to of the the a string a repeating the at at at at the the the the the the the the
the the the the the the the the the the the the the the the is is is is is is the is the is is
Attgru : pad a type with a given epsilon ( case str )
NCS :adds padding to the beginning of the string with spaces if the string ' s length is less than the
given length
Code2seq : converts a string to a string
HybridDrl: adds a string to the string of the string
Astattgru: adds a length to the end of the string .
CodeAstnn: adds padding to the beginning of the string with spaces
CAST  : adds padding to the right of a string with a repeating character if the string ' s length is
less than the minimum length
---------------------------------------
code
 public void addListener(final IFilterPanelListener<T> listener){
  m_listeners.addListener(listener);
}
Ground Truth: adds a listener that is notified about changes in the filter .
CodeNN : adds a new to . the notified button listeners the is will is .
HDeepcom : adds a listener that notified notified notified about in in
Attgru : adds a listener object that is notified about changes .
NCS :add a listener that is notified about changes in the panel .
Code2seq : add a listener to be notified
HybridDrl: adds a listener to the list of listeners
Astattgru: adds a listener that will be notified of close drawer of an existing or update of the data
.
CodeAstnn: adds a listener to the list of listeners
CAST  : adds a listener that is notified about changes in the panel .
---------------------------------------
code
 public static boolean isFileURL(URL url){
  String protocol=url.getProtocol();
  return (URL_PROTOCOL_FILE.equals(protocol) || URL_PROTOCOL_VFSFILE.equals(protocol) ||
URL_PROTOCOL_VFS.equals(protocol));
}
Ground Truth: determine whether the given url points to a resource in the file system , that is , has
protocol " file " , " vfsfile " or " vfs " .
CodeNN : returns if the given url and file file resource the the and the the the , , , , " " " " " " "
. . . . . . . . compared compared compared compared compared compared compared compared compared
compared compared compared compared compared that
HDeepcom : returns true the url is path to url , file file , , , , , file file file " " " " " " " " "
" "
Attgru : determine whether the given url points to a resource in the current url string .
NCS :determine whether the given url points to a resource in the file system , that " vfs " , " file "
, " vfsfile " .
Code2seq : checks if the url is the
HybridDrl: check if the given url is a protocol
Astattgru: determine whether the given url points to a resource in the file system , that is , has
protocol the specified file .
CodeAstnn: determine whether the given url points to a resource in the file system , that is , has
protocol " file " vfs " .
CAST  : determine whether the given url points to a resource in the file system , that is , has
protocol " file " , " vfsfile " or " vfs " .
---------------------------------------
code
```

```
 private static InputStream nullInputStream() throws NullPointerException {
  if (currentTimeMillis() > 0) {
    return null;
  }
  throw new NullPointerException();
}
Ground Truth: the following two methods exist because in , out , and err must be initialized to null .
the compiler , however , cannot be permitted to inline access to them , since they are later set to
more sensible values by initializesystemclass ( ) .
CodeNN : returns a default at . buffer for a
HDeepcom : convert stream take return returning , , , , , err err err err err stream . stream stream
stream stream stream stream stream stream stream stream stream stream stream stream stream stream
stream stream stream stream stream stream stream . . .
Attgru : the following two methods exist because we have to use that ( ) , ( ) , ( ) ) and then system
. will be invoked in the input stream , if still system running creation time .
NCS :the following methods checks , however , and err , or err , clear , err ,
Code2seq : read the next stream from the
HybridDrl: returns the input stream of the input stream
Astattgru: the following two methods members through files .
CodeAstnn: the err function to two methods in the same time , since later access to them , and
permitted to err err .
CAST  : the following two methods exist because in , out , and err must be initialized to null . the
compiler , however , cannot be permitted to inline access to them , since they are later set to more
sensible values by initializesystemclass ( ) .
---------------------------------------
code
 public static void printLine(Object message){
  if (!isDisabled()) {
    printLine(String.valueOf(message));
  }
}
Ground Truth: prints the message passed as a non - string object and a new line .
CodeNN : enable the position message representation of message either either that . . . line message
ident line
HDeepcom : print line line message . debugging t t print it it . . . .
Attgru : prints the value of the given object as it as a dot
NCS :prints a line to the output stream .
Code2seq : prints the message to the output
HybridDrl: prints the message to the default
Astattgru: prints the given line string .
CodeAstnn: prints the message and a new line object .
CAST  : prints the message passed as a non - string object .
---------------------------------------
code
 public void accumulate(TaggedLogAPIEntity entity) throws Exception {
  AggregateAPIEntity current=root;
  for (  String groupby : groupbys) {
    String tagv=locateGroupbyField(groupby,entity);
    if (tagv == null || tagv.isEmpty()) {
      tagv=UNASSIGNED_GROUPBY_ROOT_FIELD_NAME;
    }
    Map<String,AggregateAPIEntity> children=current.getEntityList();
    if (children.get(tagv) == null) {
      children.put(tagv,factory.create());
      current.setNumDirectDescendants(current.getNumDirectDescendants() + 1);
    }
    AggregateAPIEntity child=children.get(tagv);
    if (counting)     count(child);
    for (    String sumFunctionField : sumFunctionFields) {
      sum(child,entity,sumFunctionField);
    }
    current=child;
  }
}
Ground Truth: currently only group by tags groupbys ' first item always is site , which is a reserved
field
CodeNN : currently the group entity tags . ' of exists always attributes site which which is aggregate
reserved reserved reserved a
HDeepcom : tell only group nodes ) , view the the the the ' is reserved reserved reserved reserved
reserved . . . reserved reserved reserved is . . . . . . . . . . .
Attgru : currently only group by tags url get again .
NCS :currently only group by tags groupbys ' first item always is site groupbys
Code2seq : checks if the given entity entity
HybridDrl: test the entity
Astattgru: currently managed an entity instance from e . g . " " information . $ tuple in the contexts
for which one field is requested .
CodeAstnn: currently only group by groupbys groupbys tags
CAST  : currently only group by tags groupbys ' first item always is site , which is a reserved field
---------------------------------------
code
 @Override public void refreshUsersOrGroupsListFromServer(){
  showLoadingDialog(getString(R.string.common_loading));
  GetShareWithUsersAsyncTask getTask=new GetShareWithUsersAsyncTask(this);
  Object[] params={getFile(),getAccount(),getStorageManager()};
  getTask.execute(params);
}
Ground Truth: get users and groups from the server to fill in the " share with " list
CodeNN : call users new in from the share and a information and share share " of the
HDeepcom : returns all groups the the fill fill in share share share share share
Attgru : get users and groups from the server to fill in the dictionary that were * de changed changes
from the exported definition
NCS :get users from the server with users in the " share " list and " share with users .
Code2seq : this method is called when the
HybridDrl: this method is used to execute a file
Astattgru: no - op if there are details selected in the experiment
CodeAstnn: fill " share " and groups to groups in the server with " share " list
CAST  : get users and groups from the server to fill in the " share with " list
---------------------------------------
code
 @Override public void parse(InputStream in,String baseURI) throws IOException, RDFParseException,
RDFHandlerException {
  if (in == null) {
    throw new IllegalArgumentException("Input stream cannot be 'null'");
  }
  if (baseURI == null) {
    throw new IllegalArgumentException("Base URI cannot be 'null'");
  }
  InputSource inputSource=new InputSource(new BOMInputStream(in,false));
  inputSource.setSystemId(baseURI);
  parse(inputSource);
}
Ground Truth: parses the data from the supplied inputstream , using the supplied baseuri to resolve
any relative uri references .
CodeNN : parses the of the inputstream using an for the creating baseuri url the generic generic be
generic equivalent uri uri uri uri uri now .
HDeepcom : parses a main the xml xml xml . for for for for for a ( ( calls calls calls calls calls
calls calls calls ( ( ( ( ( ( ) ) ) ) ) ) . .
Attgru : compile a value and appends it to xml stream and store any value children which are for non -
and the exception of failure process .
NCS :parses the data from the supplied inputstream , using the supplied baseuri references as uri .
Code2seq : parses the http response from the
HybridDrl: parses the input stream from the given input stream
```

```
Astattgru: parses the specified information from the supplied inputstream . uri it is parsed to handle
the data from this string is representing the parser .
CodeAstnn: parses the data from the supplied inputstream , using the supplied baseuri to resolve any
relative uri references before
CAST  : parses the data from the supplied inputstream , using the supplied baseuri to resolve any
relative uri references .
---------------------------------------
code
 public ResourcesPoet addTypedArray(String name,@NotNull List<String> values){
  Element element=document.createElement("array");
  element.setAttribute("name",name);
  for (  String value : values) {
    Element valueElement=document.createElement("item");
    valueElement.appendChild(document.createTextNode(value));
    element.appendChild(valueElement);
  }
  resourceElement.appendChild(element);
  return this;
}
Ground Truth: add a typed array to the config
CodeNN : creates an of the config config
HDeepcom : adds the elements of the config
Attgru : add a . the name of the options compared to the provided name .
NCS :add an integer array to the config
Code2seq : creates a new element for the
HybridDrl: add a list of values to the config
Astattgru: add an array of values to the config
CodeAstnn: add an array of config
CAST  : add a string array to the config
---------------------------------------
code
 @Override public String toString(){
  StringBuffer BfString=new StringBuffer();
  BfString.append("\tAttribute ranking.\n");
  if (m_starting != null) {
    BfString.append("\tIgnored attributes: ");
    BfString.append(startSetToString());
    BfString.append("\n");
  }
  if (m_threshold != -Double.MAX_VALUE) {
    BfString.append("\tThreshold for discarding attributes: " + Utils.doubleToString(m_threshold,8,4)
+ "\n");
  }
  return BfString.toString();
}
Ground Truth: returns a description of the search as a string
CodeNN : returns a program of of payload
HDeepcom : returns the print representation of the , this this .
Attgru : returns a string representation of this format string . yes , if this call is .
NCS :returns a description of the threshold as a string
Code2seq : returns a string representation of this
HybridDrl: returns a string representation of this predictor
Astattgru: returns a string representation of this estimator
CodeAstnn: returns a description of the classifier as a string
CAST  : returns a description of the search as a string
---------------------------------------
code
 private boolean startsWithSpace(String str){
  return str.length() != 0 && str.charAt(0) == ' ';
}
Ground Truth: checks if the string starts with a space character , false if the string is empty or
```

starts with a non - space character .

CodeNN : returns if the the starts is the the the the slash slashes slashes

HDeepcom : check if a builder starts a a ' ' ' ' isn isn isn isn isn isn isn isn isn isn isn isn isn isn . isn

Attgru : gets the if value has a nested keyword .

NCS :check if the string starts with a space character , false if the string is empty or is not a space

Code2seq : checks if the string is a

HybridDrl: returns true if the string is a valid string

Astattgru: checks if the string ends with an space character .

CodeAstnn: checks if the string ends with a real number

CAST : checks if the string starts with a space character , false if the string is empty or starts with a non - space character

----

code

```
 public boolean mouseDragged(MouseEvent e){
  if (grabbed_plot_graphics_) {
    int x=e.getX();
    int y=e.getY();
    int dx=x - prevX;
    int dy=y - prevY;
    plotX+=dx;
    plotY+=dy;
    prevX=x;
    prevY=y;
    graph.resize(plotX,plotY,plotWidth,plotHeight);
    OMGraphicList plotGraphics=graph.getPlotGraphics();
    plotGraphics.generate(getProjection(),true);
    repaint();
  }
  return false;
}
```

Ground Truth: called whenever the mouse is dragged on this layer and one of the requested mouse modes is active .

CodeNN : compute the mark , . position image point the the a point point the the the . . . .

HDeepcom : set the mouse mouse whether rectangle mouse mouse mouse mouse pressed pressed

Attgru :

NCS :if the user is one box of the box , the box will be placed to the map , is unexpected the user of the box .

Code2seq : this method is called to the

HybridDrl: returns true if the mouse is a mouse event

Astattgru: called when a mouse button is being made with the horizontal location .

CodeAstnn: 41e the mouse is pressed for the requested mouse event .

CAST : called whenever the mouse is pressed by the user and one of the requested mouse modes is active .

----

code

```
 public synchronized int exportObject(String name,Object obj) throws CannotCompileException {
  Class clazz=obj.getClass();
  ExportedObject eo=new ExportedObject();
  eo.object=obj;
  eo.methods=clazz.getMethods();
  exportedObjects.addElement(eo);
  eo.identifier=exportedObjects.size() - 1;
  if (name != null)   exportedNames.put(name,eo);
  try {
    stubGen.makeProxyClass(clazz);
  }
 catch (  NotFoundException e) {
    throw new CannotCompileException(e);
  }
  return eo.identifier;
}
Ground Truth: exports an object . this method produces the bytecode of the proxy class used to access
the exported object . a remote applet can load the proxy class and call a method on the exported
object .
CodeNN : exports an object class field the produces the bytecode the the proxy class . . method the
exported class . call proxy proxy proxy proxy find find proxy proxy proxy call call the the the the
the the the the packages the packages packages packages packages the the the
HDeepcom : exports an object . this method proxy the the proxy the proxy proxy to to ' be the be be be
the the the the the the the the the the the the the proxy proxy the exported . . . . . . . . . . .
Attgru : exports an object . this method produces the bytecode of the proxy class with all the given
object exported in the array .
NCS :exports an object to a remote object . this method produces the proxy class and call its own
bytecode of the proxy class . a remote proxy class can load non - proxy object . a remote applet on
the exported object .
Code2seq : adds a new object to the
HybridDrl: create a method from the given object
Astattgru: exports an object . this method produces a that was accessed when using json object and may
then invoke the underlying table .
CodeAstnn: exports an object . produces a remote class used to produces the proxy object . the proxy
class cannot be used .
CAST  : exports an object . this method produces the bytecode of the proxy class used to access the
exported object . a remote applet can load the proxy class and call a method on the exported object .
---------------------------------------
code
 @Override public boolean connectionAllowed(EventSetDescriptor esd){
  return connectionAllowed(esd.getName());
}
Ground Truth: returns true if , at this time , the object will accept a connection via the named event

CodeNN : returns true if , at this time , the object will accept a connection according to the
supplied eventsetdescriptor
HDeepcom : returns true if , at this time , the object will accept a connection according the the
supplied
Attgru : returns true if , at this time , the object will accept a connection according to the
supplied eventsetdescriptor
NCS :returns true if , at this time , the object will accept a connection according to the supplied
eventsetdescriptor
Code2seq : returns true if at the given
HybridDrl: returns true if at this time the object will accept a connection
Astattgru: returns true if , at this time , the object will accept a connection according to the
supplied eventsetdescriptor
CodeAstnn: returns true if , at this time , the object will accept a connection according to the
supplied eventsetdescriptor
CAST  : returns true if , at this time , the object will accept a connection via the named event
---------------------------------------
code
 public static Request newPostOpenGraphActionRequest(Session session,OpenGraphAction
openGraphAction,Callback callback){
  if (openGraphAction == null) {
    throw new FacebookException("openGraphAction cannot be null");
  }
  if (Utility.isNullOrEmpty(openGraphAction.getType())) {
    throw new FacebookException("openGraphAction must have non-null 'type' property");
  }
  String path=String.format(MY_ACTION_FORMAT,openGraphAction.getType());
  return newPostRequest(session,path,openGraphAction,callback);
}
Ground Truth: creates a new request configured to publish an open graph action .
CodeNN : creates a new request configured . publish be .
HDeepcom : creates a new request configured to publish . open open the the . .
```

```
Attgru : creates a new request configured to publish an open request . this callback must be re for
the request progress .
NCS :creates an action request configured to publish an open graph action .
Code2seq : creates a new request request
HybridDrl: returns a new http request
Astattgru: creates a new request configured to publish an access limit with the new and adds them to
session session with the specified sign - bound properties . this allows you to change the initial
request the information relating to the new session id
CodeAstnn: creates an open graph configured to publish a new request configured .
CAST  : creates a new request configured to publish an open graph action .
---------------------------------------
code
 public static void generateExtensionManager(File rootFolder) throws IOException {
  File outFile=new File(rootFolder,REGISTRY_PATH);
  StringBuilder builder=new StringBuilder();
  builder.append("package org.eclipse.che.ide.client;\n\n");
  generateImports(builder);
  generateClass(builder);
  FileUtils.writeStringToFile(outFile,builder.toString());
}
Ground Truth: generate to source of the class .
CodeNN : generate a source of the class file
HDeepcom : generate the source of the class .
Attgru : generate into an xml version field .
NCS :generate source of the class
Code2seq : creates a new file file to
HybridDrl: writes a file to a file
Astattgru: generate to source of the class
CodeAstnn: generate to source of the class
CAST  : generate to source of the class .
---------------------------------------
code
 public void send(Set recipients,boolean multicast) throws InterruptedException, ReplyException {
  final boolean isDebugEnabled=logger.isDebugEnabled();
  if (Thread.interrupted())   throw new InterruptedException();
  recipients=new HashSet(recipients);
  DistributedMember me=originDm.getDistributionManagerId();
  if (recipients.contains(me)) {
    recipients.remove(me);
  }
  if (isDebugEnabled) {
    logger.debug("Recipients for SerialAckedMessage are {}",recipients);
  }
  rp=new ReplyProcessor21(originDm,recipients);
  processorId=rp.getProcessorId();
  setRecipients(recipients);
  setMulticast(multicast);
  Set failures=originDm.putOutgoing(this);
  if (failures != null && failures.size() > 0) {
    for (Iterator i=failures.iterator(); i.hasNext(); ) {
      InternalDistributedMember mbr=(InternalDistributedMember)i.next();
      if (isDebugEnabled) {
        logger.debug("Unable to send serial acked message to {}",mbr);
      }
    }
  }
  rp.waitForReplies();
}
Ground Truth: send the message and wait for replies
CodeNN : send the message and an timeout notifications
HDeepcom : send a for message sends for .
```

Attgru : close the datagram connection log (either by an assignment or stop associated to the tcp and client client to finish) . it notifies the on the associated process tracking " server : success . and

NCS :send the message to a listening , and set the local listening to the server . replies replies with replies ones

Code2seq : this method is called when the

HybridDrl: sends a profile to the request

Astattgru: send a notification object and updates the message and it ' s profile .

CodeAstnn: send replies message and wait for replies

CAST : send the message and wait for replies

----

code

```
 public static boolean hasTokenInformation(Bundle bundle){
  if (bundle == null) {
    return false;
  }
  String token=bundle.getString(TOKEN_KEY);
  if ((token == null) || (token.length() == 0)) {
    return false;
  }
  long expiresMilliseconds=bundle.getLong(EXPIRATION_DATE_KEY,0L);
  if (expiresMilliseconds == 0L) {
    return false;
  }
  return true;
}
```

Ground Truth: returns a boolean indicating whether a bundle contains properties that could be a valid saved token .

CodeNN : returns a boolean indicating whether the json contains specified or could be a valid saved token

HDeepcom : returns whether a whether indicating a bundle the the could could could could a a

Attgru : indicates whether a bundle throw an exception

NCS :returns a boolean indicating whether a bundle contains properties that could be a valid bundle .

Code2seq : checks if the given key is

HybridDrl: returns true if the given container has been established

Astattgru: returns whether the given bundle contains a bundle or a bundle with the same false . optionally caller if both the given are valid and known for the request .

CodeAstnn: returns a boolean indicating whether a bundle saved saved properties could be a valid saved saved bundle .

CAST : returns a boolean indicating whether a bundle contains properties that could be a valid saved token .

----

code

```
 public static void toString(Iterator<?> iter,String separator,StringBuilder sb){
  while (iter.hasNext()) {
    sb.append(iter.next());
    if (iter.hasNext()) {
      sb.append(separator);
    }
  }
}
```

Ground Truth: converts an iterator to a string by concatenating all of the string representations of objects in the iterator , divided by a separator .

CodeNN : similar a debug a a given a iteration with given string the the the the separator separator separator separator separator item given

HDeepcom : helper multiple converts separated of concatenating concatenating of a a a a a a number number . redundant redundant . be be be be be be be be be be be be be be be be be be be be be . . . .

Attgru : converts a string into a charset

NCS :creates a string containing all elements of the iterator equivalents .

Code2seq : appends the given string to the

HybridDrl: returns a string representation of the string

Astattgru: converts the given iterator to a string , optionally joining them .

CodeAstnn: converts the iterator into a string by concatenating all part of the specified separator in the iterator , iterator , and arrays .

CAST : converts an iterator to a string by concatenating all of the string representations of objects in the iterator , divided by a given separator .

----

code

```
 public boolean isToRead(){
  Iterator<VariableValue> i=variables.iterator();
  while (i.hasNext()) {
    VariableValue v=i.next();
    if (v.isToRead()) {
      return true;
    }
  }
  return false;
}
```

Ground Truth: this variable needs to be read if any of it ' s subsidiary variables needs to be read .

CodeNN : checks variable iff uses has contains contains variable the of variable variable this far method method method method throwing far far method method far thrown thrown thrown thrown thrown . needs needs needs needs needs needs needs needs needs needs needs needs needs needs needs needs method method method

HDeepcom : a if there a is " real "

Attgru : returns whether the rhs is transient as a letter ; value do not convert record to the cursor localization values converted .

NCS :variable variable needs to be written to the variable .

Code2seq : returns true if the value is

HybridDrl: returns true if the node is a variable value

Astattgru: check if it is a (ready)

CodeAstnn: this returns true if any of the variables needs to be written .

CAST : this variable needs to be written if any of it ' s subsidiary variables needs to be written .

----

code

```
 public int write(byte[] destMac,byte[] packet,int offset,int byteCount){
  if (destMac == null) {
    throw new NullPointerException("destMac == null");
  }
  if (packet == null) {
    throw new NullPointerException("packet == null");
  }
  Arrays.checkOffsetAndCount(packet.length,offset,byteCount);
  if (destMac.length != 6) {
    throw new IllegalArgumentException("MAC length must be 6: " + destMac.length);
  }
  return sendPacket(fd,mInterfaceName,mProtocolType,destMac,packet,offset,byteCount);
}
```

Ground Truth: writes a raw packet to the desired interface . a l2 header will be added which includes the specified destination address , our source mac , and the specified protocol type . the caller is responsible for computing correct ip - header and payload checksums .

CodeNN : use a raw packet to the desired . . % l2 header and be be includes includes includes includes on ip ip the the the the the the the . . . .

HDeepcom : writes a used packet to from endpoint will . l2 l2 the the the the the the the the the the the the the , , the the the the the mac , header header header header header header header header header header header header header the . .

Attgru : writes a raw packet to the desired interface . a l2 header will be added which includes the specified destination address , our source mac , and the specified protocol type . this method blocks until all records will be updated , the initial specified protocol .

NCS :writes a raw packet to the desired interface . a raw header will be added which includes the specified destination address , our source mac , and the specified protocol type . the caller includes

```
computing the specified destination address , and payload the specified header will be added to
Code2seq : writes a byte array from the
HybridDrl: writes a packet to the given byte
Astattgru: writes a single , encoded as bytes , padding down by the specified
CodeAstnn: writes a raw packet to the desired interface . the payload includes a l2 payload , our
protocol will be added which includes the desired interface .
CAST  : writes a raw packet to the desired interface . a l2 header will be added which includes the
specified destination address , our source mac , and the specified protocol type . the caller is
responsible for computing correct ip - header and payload checksums .
---------------------------------------
code
 protected void stopWraparoundTest(){
  if (testRunning && wrapTest) {
    wrapTimer.stop();
    statusText1.setText("Wraparound Test Stopped, " + Integer.toString(numErrors) + " Errors Found");
    statusText1.setVisible(true);
    statusText2.setText(Integer.toString(numIterations) + " Cycles Completed");
    statusText2.setVisible(true);
  }
}
Ground Truth: local method to stop a wraparound test
CodeNN : this the for . . errors .
HDeepcom : stops method test based the
Attgru : called by the test method that stops on .
NCS :stop test of the method , to stop the test aug
Code2seq : called when the main button is
HybridDrl: stops the <unk>
Astattgru: records the current usage text ( generally any of the or cpu sets in the text area ) .
CodeAstnn: local method to stop an ctrl message
CAST  : local method to stop an output test
---------------------------------------
code
 private void nukeSymbols(){
  _count=0;
  _longestCollisionList=0;
  Arrays.fill(_mainHash,0);
  Arrays.fill(_mainNames,null);
  Arrays.fill(_collList,null);
  _collCount=0;
  _collEnd=0;
}
Ground Truth: helper method called to empty all shared symbols , but to leave arrays allocated
CodeNN : diff distributed on a object
HDeepcom : helper that be edit an is order symbols to to to to to to . . . . . forms
Attgru : this runs the program has been allocated and its original value , to list the list over
variables
NCS :helper method to leave all shared symbols
Code2seq : resets the current list of the
HybridDrl: returns a new test
Astattgru: sets the range " count " if not < p / > < br > on , replacement , issue is not called , but
we don ' t need to be explicitly
CodeAstnn: helper method to assign all shared symbols , to leave arrays allocated
CAST  : helper method called to empty all shared symbols , but to leave arrays allocated
---------------------------------------
code
 private static float determineEnemyBlitzStrength(final Territory blitzHere,final List<Route>
blitzTerrRoutes,final List<Territory> blockTerr,final GameData data,final PlayerID ePlayer){
  final HashSet<Integer> ignore=new HashSet<>();
  ignore.add(1);
  final CompositeMatch<Unit> blitzUnit=new
CompositeMatchAnd<>(Matches.unitIsOwnedBy(ePlayer),Matches.UnitCanBlitz,Matches.UnitCanMove);
```

```
  final CompositeMatch<Territory> validBlitzRoute=new
CompositeMatchAnd<>(Matches.territoryHasNoEnemyUnits(ePlayer,data),Matches.TerritoryIsNotImpassableToL
andUnits(ePlayer,data));
  final List<Route> routes=new ArrayList<>();
  final List<Unit>
blitzUnits=findAttackers(blitzHere,2,ignore,ePlayer,data,blitzUnit,validBlitzRoute,blockTerr,routes,fa
lse);
  for (  final Route r : routes) {
    if (r.numberOfSteps() == 2) {
      blitzTerrRoutes.add(r);
    }
  }
  return strength(blitzUnits,true,false,true);
}
Ground Truth: determine the enemy potential for blitzing a territory - all enemies are combined
CodeNN : determine the enemy potential for of territory - are are are combined
HDeepcom : determine the enemy potential for territory territory territory territory combined are
Attgru : determine the enemy potential for a territory - all are combined
NCS :determine the enemy potential potential for blitzing
Code2seq : returns the next set of the
HybridDrl: returns the first of the given objects in the given list
Astattgru: determine the enemy and compute the relevant , , .
CodeAstnn: determine the enemy territory - all enemies for blitzing a territory - all enemies are
ignored
CAST  : determine the enemy potential for blitzing a territory - all enemies are combined
---------------------------------------
code
 public Name join(String identifier){
  validateLowerUnderscore(identifier);
  List<NamePiece> newPieceList=new ArrayList<>();
  newPieceList.addAll(namePieces);
  newPieceList.add(new NamePiece(identifier,CaseFormat.LOWER_UNDERSCORE));
  return new Name(newPieceList);
}
Ground Truth: returns a new name containing the pieces from this name plus the given identifier added
on the end .
CodeNN : follows a new from containing based
HDeepcom : generate a css with from as name form hash form form form form form form form form form
form form form
Attgru : encodes a name and name into a new name within the context of the specified imports the
supplied sequence , and all the sequence attributes with this identifier .
NCS :returns a new name containing the pieces from another range of the requested name .
Code2seq : creates a new list of the
HybridDrl: returns a new name with the given name
Astattgru: creates a new empty list with the given name and type ( name , list ) pairs of the given
string name and the list of dependencies .
CodeAstnn: creates a name from a given identifier name and lower case - plus the methods of the given
name .
CAST  : returns a new name containing the pieces from this name plus the pieces of the given
identifier added on the end .
---------------------------------------
code
 protected boolean readReceiveDelay(){
  try {
    receiveDelay=Integer.parseInt(receiveDelayField.getText());
  }
 catch (  Exception e) {
    statusText1.setText(rb.getString("Error7"));
    statusText1.setVisible(true);
    receiveDelay=0;
    errorInStatus1=true;
    resetNotes2();
    return (false);
  }
  if (receiveDelay < 0) {
    statusText1.setText(rb.getString("Error8"));
    statusText1.setVisible(true);
    receiveDelay=0;
    errorInStatus1=true;
    resetNotes2();
    return (false);
  }
  if (receiveDelay > 65535) {
    statusText1.setText(rb.getString("Error9"));
    statusText1.setVisible(true);
    receiveDelay=0;
    errorInStatus1=true;
    resetNotes2();
    return (false);
  }
  return true;
}
Ground Truth: read receive delay from window returns ' true ' if successful , ' false ' if an error
was detected . if an error is detected , a suitable error message is placed in the notes area
CodeNN : read the status and unable notes ' ' ' is is is is is is is . . detected detected
HDeepcom : read - delay reply read password password password ' ' ' ' ' ' ' ' comment notes notes
notes notes notes notes notes notes notes notes notes notes
Attgru : read number from user 2 s filling invoked as invoked when is set to available .
NCS :read ( ) from window returns ' true ' if successful , a suitable error message is placed in the
notes area
Code2seq : check if the current message is
HybridDrl: read text to see if the text is successful
Astattgru: read number of new messages from the
CodeAstnn: read receive delay time , ' true ' if successful , ' false ' if an error was detected . if
an error was detected , a suitable error message is placed in the notes area
CAST  : read receive receive receive receive events from window returns ' true ' if successful , '
false ' if an error was detected . if an error is detected , a suitable error message is placed in the
notes area
---------------------------------------
code
 @Override public void paintComponent(Graphics g){
  super.paintComponent(g);
  Graphics2D graphics=(Graphics2D)g;
  graphics.setBackground(mColorSpectrumBackground);
  RenderingHints renderHints=new
RenderingHints(RenderingHints.KEY_ANTIALIASING,RenderingHints.VALUE_ANTIALIAS_ON);
  renderHints.put(RenderingHints.KEY_RENDERING,RenderingHints.VALUE_RENDER_QUALITY);
  graphics.setRenderingHints(renderHints);
  drawFrequencies(graphics);
  drawCursor(graphics);
}
Ground Truth: renders the channel configs , lines , labels , and cursor
CodeNN : multiline a background array a graphics , a the given image
HDeepcom : renders the comment paint a top top paint paint paint . . . . . being being being being
being being being being . . being being being being being being being being being being being being
being being being . . . . . on on on on
Attgru : paints the of the .
NCS :renders the channel using the nominal labels
Code2seq : draws the background and show the
HybridDrl: paints the caret with the given graphics
Astattgru: renders the component using the selected .
CodeAstnn: renders the background for the channel .
```

```
CAST  : renders the channel configs , lines , labels , and cursor
---------------------------------------
code
 public boolean deleteAttachmentPoint(DatapathId sw,OFPort port){
  AttachmentPoint ap=new AttachmentPoint(sw,port,new Date(0));
  if (this.oldAPs != null) {
    ArrayList<AttachmentPoint> apList=new ArrayList<AttachmentPoint>();
    apList.addAll(this.oldAPs);
    int index=apList.indexOf(ap);
    if (index > 0) {
      apList.remove(index);
      this.oldAPs=apList;
    }
  }
  if (this.attachmentPoints != null) {
    ArrayList<AttachmentPoint> apList=new ArrayList<AttachmentPoint>();
    apList.addAll(this.attachmentPoints);
    int index=apList.indexOf(ap);
    if (index > 0) {
      apList.remove(index);
      this.attachmentPoints=apList;
      return true;
    }
  }
  return false;
}
Ground Truth: delete ( sw , port ) from the list of list of attachment points and oldaps .
CodeNN : delete the attachment attachment point older the the of list of attachment points points points
HDeepcom : delete a new and from a and and and a a a a a a a a ) ) attachment attachment ) attachment attachment attachment attachment attachment attachment attachment attachment attachment
Attgru : the method that the train will not be modified if the work returns deleted platform the network .
NCS :delete a list of attachment points from the list of attachment points .
Code2seq : returns true if the given device
HybridDrl: remove a given port from the given port and the given port
Astattgru: delete ( limit , ) from the given attachment device .
CodeAstnn: remove ( sw , port ) from the list of attachment points and oldaps .
CAST  : delete ( sw , port ) from the list of list of attachment points and oldaps .
---------------------------------------
code
 public synchronized int totalRecoverFiles(){
  int total=0;
  for (  File file : fileDetails.values()) {
    if (file.reused() == false) {
      total++;
    }
  }
  return total;
}
Ground Truth: total number of files to be recovered ( potentially not yet done )
CodeNN : tell a of of from the recovered
HDeepcom : implements disk files files the try yet yet if files if files files files files disk disk disk disk disk disk disk disk disk disk disk disk
Attgru : to be called to persistent of all concurrent updates .
NCS :total files within the file system
Code2seq : returns the index of the file
HybridDrl: returns the total of the file
Astattgru: fetches the files and return the index of all files that have been decoded . in this case the filter ' s have been loaded information it is yet removed .
CodeAstnn: total number of files recovered ( potentially open ) files to be recovered
```

```
CAST  : total files of files that were recovered ( potentially not yet done )
---------------------------------------
code
 public static synchronized void cleanup(){
  w.lock();
  try {
    secp256k1_destroy_context(Secp256k1Context.getContext());
  }
  finally {
    w.unlock();
  }
}
Ground Truth: libsecp256k1 cleanup - this destroys the secp256k1 context object this should be called
at the end of the program for proper cleanup of the context .
CodeNN : libsecp256k1 cleanup - this destroys the secp256k1 context with it lock called done . the .
thread thread . .
HDeepcom : libsecp256k1 cleanup by destroys the secp256k1 context the be be be be the the the the the
the the the the the the cleanup .
Attgru : libsecp256k1 cleanup - this destroys the secp256k1 context object this should be called at
the end of the program and then initialize itself as an element .
NCS :libsecp256k1 cleanup - destroys the secp256k1 for the secp256k1 object . this should be called at
the end of the secp256k1 for future usage .
Code2seq : this method is called when the
HybridDrl: clears the state of the reader
Astattgru: libsecp256k1 cleanup - this destroys the secp256k1 context object this should be called at
the context main thread finishes when the last created context environment is disconnected .
CodeAstnn: libsecp256k1 cleanup - this destroys the secp256k1 context at the end of the program this
context .
CAST  : libsecp256k1 cleanup - this destroys the secp256k1 context object this should be called at the
end of the program for proper cleanup of the context .
---------------------------------------
code
 public InputStreamReader reader(final String charset) throws HttpRequestException {
  try {
    return new InputStreamReader(stream(),getValidCharset(charset));
  }
 catch (  UnsupportedEncodingException e) {
    throw new HttpRequestException(e);
  }
}
Ground Truth: get reader to response body using given character set . < p > this will fall back to
using the utf - 8 character set if the given charset is null
CodeNN : get the request response . stream . the to . . < charset < fall fall fall back back the . . .
. . . . . . . . . . .
HDeepcom : get a writes content body given given character given < < p p utf ( ( ( specified ( ( (
charset ( charset charset charset charset charset charset charset charset specified specified
specified specified specified specified the specified specified the charset the the the the the the
the
Attgru : get reader to response body using given character set . < p > this will fall back as a
snapshot of a string using the specified character .
NCS :get reader to response body using given character set
Code2seq : reads the contents of the given
HybridDrl: get reader from response stream
Astattgru: get reader to an input stream
CodeAstnn: creates a charset object with charset as json string
CAST  : get reader to response body using given character set . < p > this will fall back to using the
utf - 8 character set if the given charset is null
---------------------------------------
code
 private static int indexOf(int fromIndex,CharSequence csq){
  if (csq == null)   return 0;
  int length=csq.length();
  int j=fromIndex;
  int i=(j < length) ? csq.charAt(j++) : 0;
  i<<=16;
  i|=(j < length) ? csq.charAt(j++) : 0;
  return i;
}
Ground Truth: returns the index starting at the specified index ( two characters at a time ) .
CodeNN : returns the prepared index at the specified index part of specified specified
HDeepcom : returns the index starting at the specified specified in characters characters characters
specified specified specified specified specified . .
Attgru : returns the position within the specified character sequence .
NCS :returns the index starting at a time from this cursor points at a time . returns - 1 if the end
points of the first character at this point .
Code2seq : returns the index of the given
HybridDrl: returns the index of the specified character in the specified character
Astattgru: returns the index starting at the specified index ( inclusive ) .
CodeAstnn: returns the index starting at the specified index ( two starting at the specified index
CAST  : returns the index starting at the specified index ( two characters at a time ) .
---------------------------------------
code
 private ArrayList<String> filterYears(ArrayList<String> items){
  Iterator<String> iterator;
  Calendar now=Calendar.getInstance();
  int year=now.get(Calendar.YEAR);
  String yearInString=String.valueOf(year);
  Log.d(TAG,"Year: " + yearInString);
  int prevYear=year - 1;
  String prevYearInString=String.valueOf(prevYear);
  Log.d(TAG,"Previous year: " + prevYearInString);
  for (iterator=items.iterator(); iterator.hasNext(); ) {
    String s=iterator.next();
    if (s.matches(".*(19|20)\\d{2}.*") && !s.contains(yearInString) && !s.contains(prevYearInString))
{
      Log.d(TAG,"Filtering out year " + s);
      iterator.remove();
    }
  }
  Log.d(TAG,"Items: " + items.toString());
  return items;
}
Ground Truth: remove categories that contain a year in them ( starting with 19__ or 20__ ) , except
for this year and previous year rationale : https : / / github . com / commons - app / apps - android
- commons / issues / 47
CodeNN : checks categories from of year .
HDeepcom : remove categories that contain a year any ( ( ( ( ( . rationale rationale 1 1 commons
commons commons commons commons commons commons commons commons commons commons commons commons
commons commons commons commons commons commons commons commons commons commons
Attgru : remove categories that don ' t configure , use
NCS :remove categories that contain them in a year that contain the " year " or " if the last year and
previous year in them , then remove them that that contain the status of the year in the year
Code2seq : returns the string representation of the
HybridDrl: returns a list of strings from the specified list of strings
Astattgru: remove categories that contain a year in them ( starting with a single line of the common
possible number ) , and convert them to select one
CodeAstnn: remove categories that contain a year in them ( starting with 19__ or 20__ )
CAST  : remove categories that contain a year in them ( starting with 19__ or 20__ ) , except for this
year and previous year rationale : https : / / github . com / commons - app / apps - android - commons
/ issues / 47
---------------------------------------
code
```

```
 public static Configuration load(Reader reader) throws IOException {
  try {
    Properties properties=new Properties();
    properties.load(reader);
    return from(properties);
  }
  finally {
    reader.close();
  }
}
Ground Truth: obtain a configuration instance by loading the properties from the supplied reader .
CodeNN : create the token a the and stream properties the parsing stream
HDeepcom : loads the from using properties loading the the the the the the the
Attgru : returns a property object as a reader .
NCS :loads the configuration from the supplied reader .
Code2seq : loads the properties from the given
HybridDrl: create a configuration from the given reader
Astattgru: loads the configuration from the specified configuration , must be used for getting one
configuration in properties .
CodeAstnn: construct a configuration from the supplied properties instance .
CAST  : load a configuration instance by loading the properties from the supplied reader .
---------------------------------------
code
 public void poll(final TcpSocketStatisticsHandler handler){
  this.statisticsHandler=handler;
  try {
    fileLoader.load();
    final ByteBuffer buffer=fileLoader.getBuffer();
    lineParser.reset();
    lineParser.handleToken(buffer,buffer.position(),buffer.limit());
  }
  finally {
    this.statisticsHandler=null;
  }
  monitoredSockets.purgeEntriesOlderThan(updateCount);
  updateCount++;
}
Ground Truth: read from monitored file , report any changed values for monitored socket statistics .
not thread - safe , only call from a single thread .
CodeNN : resets google new . scanner the managed been available successful
HDeepcom : sends for handler . storage udp . .
Attgru : signals a dummy task , which is currently active ;
NCS :read the combined classloader from the provided file , and puts the line for the file , and puts
the statistics suppressed count .
Code2seq : this method will be called by
HybridDrl: reads a message from the queue
Astattgru: blocks for uploading to or video by current point in the file watcher .
CodeAstnn: read not thread from monitored file , report any statistics used for udp statistics .
CAST  : read from monitored file , report any changed values to the supplied handler . not thread -
safe , only call from a single thread .
---------------------------------------
code
 public int compareTo(CharBuffer otherBuffer){
  int compareRemaining=(remaining() < otherBuffer.remaining()) ? remaining() :
otherBuffer.remaining();
  int thisPos=position;
  int otherPos=otherBuffer.position;
  char thisByte, otherByte;
  while (compareRemaining > 0) {
    thisByte=get(thisPos);
    otherByte=otherBuffer.get(otherPos);
    if (thisByte != otherByte) {
      return thisByte < otherByte ? -1 : 1;
    }
    thisPos++;
    otherPos++;
    compareRemaining--;
  }
  return remaining() - otherBuffer.remaining();
}
Ground Truth: compare the remaining chars of this buffer to another char buffer ' s remaining chars .
CodeNN : compare the remaining shorts of into buffer to another . two ' remaining remaining shorts .
HDeepcom : compare the remaining parser int to . . . . remaining remaining remaining remaining
character character character character character character character character character character
character character character character character character character character character character
character character character character character character character character character character
character character character character character
Attgru : compares construction of unicode character in the remaining first stream .
NCS :compare the remaining shorts of this buffer to another remaining shorts of this buffer ' s
remaining shorts .
Code2seq : returns the number of consecutive characters
HybridDrl: compares the remaining bytes of the remaining chars of this buffer to
Astattgru: compare the remaining shorts of this buffer to another character buffer ' s remaining longs
.
CodeAstnn: compares the remaining bytes of this buffer to another byte buffer ' s remaining longs .
CAST  : compares the remaining bytes of this buffer to another char buffer ' s remaining bytes .
---------------------------------------
code
 public static String removePattern(final String source,final String regex){
  return replacePattern(source,regex,StringUtils.EMPTY);
}
Ground Truth: removes each substring of the source string that matches the given regular expression
using the dotall option .
CodeNN : calculates the substring from the filter the of of of a regex expression the
HDeepcom : removes tab . . the substring the the the the expression the expression expression . .
Attgru : removes each substring of the source string that matches the given regular expression using
the backslash .
NCS :removes each substring of the substring that matches the dotall string .
Code2seq : returns a string of the given
HybridDrl: replaces all occurrences of the given regular expression with the given regular
Astattgru: removes each substring of the source string that matches the given regular expression with
the given replacement string .
CodeAstnn: removes each substring of the source string that matches the given source string using the
dotall dotall .
CAST  : removes each substring of the source string that matches the given regular expression using
the dotall option .
---------------------------------------
code
 private void retrieveWaypoints(SearchQuery query,ArrayList<Waypoint> waypoints){
  String queryLikeSelection2="%" + query.textQuery + "%";
  String[] waypointSelectionArgs=new
String[]{queryLikeSelection2,queryLikeSelection2,queryLikeSelection2};
  Cursor cursor=null;
  try {

cursor=providerUtils.getWaypointCursor(WAYPOINT_SELECTION_QUERY,waypointSelectionArgs,WAYPOINT_SELECTI
ON_ORDER,MAX_SCORED_WAYPOINTS);
    if (cursor != null) {
      waypoints.ensureCapacity(cursor.getCount());
      while (cursor.moveToNext()) {
        Waypoint waypoint=providerUtils.createWaypoint(cursor);
        if (LocationUtils.isValidLocation(waypoint.getLocation())) {
          waypoints.add(waypoint);
        }
      }
    }
  }
  finally {
    if (cursor != null) {
      cursor.close();
    }
  }
}
Ground Truth: retrieves waypoints matching the given query from the database .
CodeNN : move the query the
HDeepcom : constructs and parameters from query
Attgru : performs a binary search for the query .
NCS :retrieves a selection query from the database .
Code2seq : this method will called when the
HybridDrl: returns predecessors list of the query
Astattgru: remove any that occur on the filter array lists returns . if no will be found to , search
for the live database .
CodeAstnn: retrieves matching database specific query .
CAST  : retrieves waypoint matching the given query from the database .
---------------------------------------
code
 public static String plnoun(final int quantity,final String noun){
  final String enoun=fullForm(noun);
  if (quantity == 1) {
    return singular(enoun);
  }
 else {
    return plural(noun);
  }
}
Ground Truth: returns either the plural or singular form of the given noun , depending on the quantity
.
CodeNN : creates either pkcs history a singular form the the the
HDeepcom : returns the of plural or singular form of the given depending depending depending depending
the available available available quantity quantity .
Attgru : find where which ends " " , " " , " { } } " structures , " an " or " accept " . if " is not
found , no rewrites .
NCS :returns the plural form of the given noun , depending on the full of the full noun .
Code2seq : returns a string representing the given
HybridDrl: returns an plural form of the given noun depending on the plural
Astattgru: helper method to use a human readable string of format gaps .
CodeAstnn: returns either the plural or singular form of the given noun , depending on the quantity ;
also prefixes the quantity .
CAST  : returns either the plural or singular form of the given noun , depending on the quantity ) .
---------------------------------------
code
 private void unshareMain(){
  int[] old=_mainHash;
  int len=_mainHash.length;
  _mainHash=new int[len];
  System.arraycopy(old,0,_mainHash,0,len);
  _mainHashShared=false;
}
Ground Truth: method that needs to be called , if the main hash structure is ( may be ) shared . this
happens every time something is added , even if addition is to the collision list ( since collision
list index comes from lowest 8 bits of the primary hash entry )
CodeNN : and the needs hash box set the the set . the the the . the the the the . . . . . hash hash
HDeepcom : called that needs , be be is be be is is up to to to to to . . . . collision collision
```

collision collision

Attgru : updates the current score if necessary . does nothing and existing term . it may be called if the current state is full and zero , its main sets , and processing each time zone pos .

NCS :method that needs to be called , if collision is (may be) shared hash since collision .

Code2seq : this method is called to the

HybridDrl: add the capacity to the array

Astattgru: run the main (in all of its parent and the children of initialization to the inside csv views and . this method must be called when the main hash code of the in the label array being added to the outer

CodeAstnn: if the shared hash is shared , the collision list of collision (primary) comes (primary) , the collision list will be added to the collision list . if primary is shared , the collision list is added to the collision list contains primary entry is shared

CAST : method that needs to be called , if the main hash structure is (may be) shared . this happens every time something is added , even if addition is to the collision list (since collision list index comes from lowest 8 bits of the primary hash entry

----

code

```
 public ComponentConfigBuilder addItemInMultifield(String multifieldLabel,int index,String itemType,String itemLabel,String value){
  String type=String.format("multifield#%s#%s",index,itemType);
  String label=String.format("%s#%s",multifieldLabel,itemLabel);
  config.add(new ConfigurationEntry(currentTab,type,label,value));
  return this;
}
```

Ground Truth: adds multifield entry , equivalent of : | tab_name | multifield # index # itemtype | multifieldlabel # itemlabel | value |

CodeNN : creates a label text of a specified | | | | | | |

HDeepcom : adds an item in the -

Attgru : adds an item to the item to be , as well as the labels of a trace variable , at the item id and the base name is at the current position .

NCS :adds a new entry .

Code2seq : creates a new row with the

HybridDrl: adds an item to the list

Astattgru: adds a number of element names and a set of labels , labels , labels , and document , registers the item , and sets the position of the label at the specified position in the list . if the item then its mapping is not a union add

CodeAstnn: adds fieldset entry , equivalent , | tab_name | value |

CAST : adds item entry , equivalent of : | tab_name | label | label | multifieldlabel # itemlabel | value |

----

code

```
 private void blackmanHarris4sMin(int size){
  int start=(windowFunction.length - size) / 2;
  int stop=(windowFunction.length + size) / 2;
  double scale=1.0 / (double)size / 0.36;
  for (int i=0; start < stop; start++, i++)   windowFunction[i]=scale * (0.35875 - 0.48829 * Math.cos(twoPI * i / size) + 0.14128 * Math.cos(2 * twoPI * i / size) - 0.01168 * Math.cos(3 * twoPI * i / size));
}
```

Ground Truth: fill an array with the values of a minimum 4 - sample blackman - harris window function

CodeNN : fill the array with the values - sample 74 - sample sample sample - - - and

HDeepcom : (the with the of of of . . ,

Attgru : fill an array with the values of 2 : 1 . 0 the number of components - - forwards - negative values * clipping intervals . supported bits are not equal 2 . intervals fills this test because the power - of - phase is longer , measured in

NCS :fill an array with the values of a minimum blackman - sample blackman - harris blackman - sample - sample blackman - sample blackman - sample - sample blackman - sample blackman - sample in the sample - sample blackman - - e . g .

Code2seq : calculate the distance of the problem

HybridDrl: updates the number of values in the given number

Astattgru: fill the specified range of the range rings into account .

CodeAstnn: fill an array with the values of a 74 3 - sample blackman - harris window function

CAST : fill an array with the values of a minimum 4 - sample blackman - harris window function

----

code

```
 public static BigDecimal parseBigDecimalForEntity(String bigDecimalString) throws
NumberFormatException {
  if (bigDecimalString == null) {
    return null;
  }
  bigDecimalString=bigDecimalString.trim();
  bigDecimalString=bigDecimalString.replaceAll(",","");
  if (bigDecimalString.length() < 1) {
    return null;
  }
  return new BigDecimal(bigDecimalString);
}
```

Ground Truth: return nulls for empty strings , as the entity engine can deal with nulls . this will provide blanks in fields where bigdecimal display . blank meaning null , vs . 0 which means 0

CodeNN : nulls entity empty biginteger specific

HDeepcom : return nulls for empty . . , . . nulls nulls nulls nulls nulls nulls nulls nulls display display be nulls nulls display . . . vs vs 1

Attgru : return nulls for empty strings , as the entity engine can deal with nulls . this will provide blanks in order to parse the operation ' s 8 bit value . this is used for queries such as keys as a der - encoding .

NCS :return nulls for empty entity engine , as normal engine can deal with nulls . this will provide blanks in fields where bigdecimal display . blank meaning 0 as normal entity , vs { vs }

Code2seq : formats the given string to a

HybridDrl: convert a big decimal to a big decimal

Astattgru: return nulls for empty strings , as if it exceeds or null value , otherwise null < p > can ' t be null , it returns null .

CodeAstnn: return nulls for fields where nulls can deal with nulls or null . bigdecimal , vs . 0 . 0

CAST : return nulls for empty strings , as the entity engine can deal with nulls . this will provide blanks in fields where bigdecimal display . blank meaning null , vs . 0 which means 0

----

code

```
 private void doPermissionWorkBeforeAndroidM(@NonNull Activity activity,@NonNull String[]
permissions,@Nullable PermissionsResultAction action){
  for (  String perm : permissions) {
    if (action != null) {
      if (!mPermissions.contains(perm)) {
        action.onResult(perm,Permissions.NOT_FOUND);
      }
 else       if (ActivityCompat.checkSelfPermission(activity,perm) !=
PackageManager.PERMISSION_GRANTED) {
        action.onResult(perm,Permissions.DENIED);
      }
 else {
        action.onResult(perm,Permissions.GRANTED);
      }
    }
  }
}
```

Ground Truth: when request permissions on devices before android m (android 6 . 0 , api level 23) do the granted or denied work directly according to the permission status

CodeNN : check an permissions on devices before android handle handle android 6 6 . , related the 23 / the granted granted granted the permission permission permission permission permission permission permission permission permission permission permission permission . . . . . classes classes classes classes classes classes . .

HDeepcom : see all of for the status higher background . . . . exchange change change

change change change change to

Attgru : handles action to stop for specified trees

NCS :when request permissions on devices before android granted work status

Code2seq : this method is called when the

HybridDrl: method to customize the user

Astattgru: call display from android . android . auth . . cancel ( ) even when request ( ) , or even (directly) * openam * (not more) the " " (e . g . " auth ") , " "

CodeAstnn: when request permissions do denied android before android m , do directly according to android 6 23 . 0

CAST : when request permissions on devices before android m (android 6 . 0 , api level 23) do the granted or denied work directly according to the permission status

----

code

```
 static boolean pairSum(int ar[],int x){
  Arrays.sort(ar);
  int len=ar.length;
  for (int i=0, j=len - 1; i < j; ) {
    if (ar[i] + ar[j] == x) {
      return true;
    }
 else     if (ar[i] + ar[j] < x) {
      i++;
    }
 else {
      j--;
    }
  }
  return false;
}
```

Ground Truth: using sorting . if we use merge sort or heap sort then (-) (nlogn) in worst case . if we use quick sort then o (n ^ 2) in worst case .

CodeNN : sorting time ( ] arrays and , intervals heap heap , , range range time two , , worst , worst , , , , , , then then then worst 2 . . lowercase lowercase lowercase lowercase lowercase lowercase lowercase cancel cancel cancel cancel cancel cancel two

HDeepcom : private to to sort arrays array . .) or or or)))))))))

Attgru : uses an array of coordinates in a given array . since cos matrix can be on the same parser that a both cg .

NCS :if we use an array or sort then sort its heap in worst . if we use 5 , return sort or heap

Code2seq : returns true if the given array

HybridDrl: returns true if the given range is a valid array

Astattgru: the bit hash values at the beginning of the input vector

CodeAstnn: using sorting . if we use merge sort or heap sort then o (nlogn) in worst case . if we use quick sort or equal to lowercase then o (n ^ 2) if we use quick sort then o (right now ^ 2) in

CAST : using sorting . if we use merge sort or heap sort then (-) (nlogn) in worst case . if we use quick sort then o (n ^ 2) in worst case .

----

code

```
 public double localScore(int i,int... parents){
  for (  int p : parents)   if (forbidden.contains(p))   return Double.NaN;
  double residualVariance=covariances.get(i,i);
  int n=getSampleSize();
  int p=parents.length;
  Matrix covxx=getSelection1(covariances,parents);
  try {
    Matrix covxxInv=covxx.inverse();
    Matrix covxy=getSelection2(covariances,parents,i);
    Matrix b=covxxInv.times(covxy);
    double dot=0.0;
    for (int j=0; j < covxy.getRowDimension(); j++) {
      for (int k=0; k < covxy.getColumnDimension(); k++) {
        dot+=covxy.get(j,k) * b.get(j,k);
      }
    }
    residualVariance-=dot;
    if (residualVariance <= 0) {
      if (isVerbose()) {
        out.println("Nonpositive residual varianceY: resVar / varianceY = " + (residualVariance / covariances.get(i,i)));
      }
      return Double.NaN;
    }
    double c=getPenaltyDiscount();
    return score(residualVariance,n,logn,p,c);
  }
 catch (  Exception e) {
    boolean removedOne=true;
    while (removedOne) {
      List<Integer> _parents=new ArrayList<>();
      for (int y=0; y < parents.length; y++)       _parents.add(parents[y]);
      _parents.removeAll(forbidden);
      parents=new int[_parents.size()];
      for (int y=0; y < _parents.size(); y++)       parents[y]=_parents.get(y);
      removedOne=printMinimalLinearlyDependentSet(parents,covariances);
    }
    return Double.NaN;
  }
}
```

Ground Truth: calculates the sample likelihood and bic score for i given its parents in a simple sem model

CodeNN : this the . (the current size of the . for one based that that that the labels

HDeepcom : calculates one first metric is is point for i for i i)) method metric metric metric metric metric) metric from)) . .

Attgru : returns the sample distance from the given distance , this new composite and given distance . it will be in order

NCS :calculates the predicted linear given the given parents times the i of the predicted model and returns the real value of the predicted model .

Code2seq : returns the distance of the given

HybridDrl: performs a complete matrix

Astattgru: get the sample likelihood with closest focus

CodeAstnn: calculates the sample likelihood and bic score for the given cid in the slot in the given position

CAST : calculates the sample likelihood and bic score for i given its parents in a simple sem model

----

code

```
 public boolean equals(Object obj){
  if (this == obj)   return true;
  if (obj instanceof ECFieldFp) {
    return (p.equals(((ECFieldFp)obj).p));
  }
  return false;
}
```

Ground Truth: compares this prime finite field for equality with the specified object .

CodeNN : returns whether the specified equality with specified specified the object

HDeepcom : compares whether to to object with with .

Attgru : compares this given object / from this object with another object .

NCS :compares this object with another object for equality . returns true if the specified object is identical to the same .

Code2seq : checks if the specified object is

HybridDrl: returns true if the specified object is equal to this object

Astattgru: compares the two objects .

CodeAstnn: compares this finite field equality with the specified object .

```
CAST  : compares this elliptic curve for equality with the specified object .
---------------------------------------
code
 @Override public boolean equals(Object o){
  if (this == o) {
    return true;
  }
  if (!(o instanceof PrototypeSize)) {
    return false;
  }
  PrototypeSize size=(PrototypeSize)o;
  return prototype.equals(size.prototype);
}
Ground Truth: indicates whether some other constantsize is " equal to " this one .
CodeNN : returns the many radians radians listens
HDeepcom : indicates whether if for object another exactly the the
Attgru : checks whether this map is equal or equal to another .
NCS :returns true if the priority map of this priority can be in some other domain .
Code2seq : compares two objects to equality
HybridDrl: returns true if the specified object is equal to this object
Astattgru: compares two instances , returning false if an exception is thrown .
CodeAstnn: compares this preview to another .
CAST  : indicates whether some other object is " equal to " this one .
---------------------------------------
code
 public void writeOperationsCarFile(){
  makeBackupFile(defaultOperationsFilename());
  try {
    if (!checkFile(defaultOperationsFilename())) {
      java.io.File file=new java.io.File(defaultOperationsFilename());
      java.io.File parentDir=file.getParentFile();
      if (!parentDir.exists()) {
        if (!parentDir.mkdir()) {
          log.error("Directory wasn't created");
        }
      }
      if (file.createNewFile()) {
        log.debug("File created");
      }
    }
    writeFile(defaultOperationsFilename());
  }
 catch (  Exception e) {
    log.error("Exception while writing the new CSV operations file, may not be complete: " + e);
  }
}
Ground Truth: store the all of the operation car objects in the default place , including making a backup if needed
CodeNN : store a file to drive player , def the the , place directory a write backup backup backup and new new new new new directory directory directory directory directory directory backup backup backup directory backup directory directory directory directory directory directory directory directory directory directory directory directory directory
HDeepcom : writes backup the used the the the - save and and and and to small small small small small . . .
Attgru : this method wraps a backup of the operation shouldn ' t use ' ' and stored in each directory and all subdirectories
NCS :store all the operation - copied in the backup while written out a backup
Code2seq : writes the file to the file
HybridDrl: saves the file to the file
Astattgru: writes the statistics file ( copy ) of the existing change into the existing file .
CodeAstnn: store the all of the operation in the default place if needed then build a backup with the
```

```
backup .
CAST  : store the all of the operation engine objects in the default place , including making a backup
if needed
---------------------------------------
code
 public static void sortFields(FieldBinding[] sortedFields,int left,int right){
  Arrays.sort(sortedFields,left,right,FIELD_COMPARATOR);
}
Ground Truth: sort the field array using a quicksort
CodeNN : set new a together the fields of a given . ( vertices .
HDeepcom : sorts numerical to array array array array
Attgru : helper method to map the range with val inclusive to one position in relationship .
NCS :sort the field array
Code2seq : sorts the given range in the
HybridDrl: sorts a range of a range of a range
Astattgru: sorts this set in to the specified row and sort the natural sort order .
CodeAstnn: sort the field array with the components
CAST  : sort the field array using the specified quicksort
---------------------------------------
code
 public static boolean isChildGroup(String group){
  return group.indexOf(GROUP_DELIMITER) != -1;
}
Ground Truth: indicates if a group is a child group , a non - top - level data group in a set of
nested data groups ( e . g . , the node or edge table of a graph or tree ) .
CodeNN : creates if the group is group child group , a non non ? .
HDeepcom : check if a group is a a a a a - group - group group group group group - - , - - - - - - - -
- - - - - - - - - - - - - . . . .
Attgru : indicates if group is a child group name / value pairs
NCS :indicates if a child group is a child of a child group , a child data set in a set of nested data
groups , or a child node is a child group - free non - top - top - top - top - top - top - level
Code2seq : returns true if the group group
HybridDrl: returns true if the given group is a group
Astattgru: indicates whether a given group is a child group , false otherwise .
CodeAstnn: indicates if a group node is a child data group of a data group , e . g . a top - level
group in the node or edge
CAST  : indicates if a group is a child group , a non - top - level data group in a set of nested data
groups ( e . g . , the node or edge table of a graph or tree ) .
---------------------------------------
code
 public int count(){
  return n;
}
Ground Truth: returns the number of data values .
CodeNN : returns the counter of is . reports syntax . . characters . is mappings is this this to to .
. . recent digit recent .
HDeepcom : counts component filter number counts 2 counts
Attgru : returns the number of values in this array .
NCS :counts the number of counter in the log .
Code2seq : returns the number of the elements
HybridDrl: returns the number of elements in this vector
Astattgru: counts the number of times in this rule .
CodeAstnn: counts the number of values in this fluent .
CAST  : returns the number of upper tokens .
---------------------------------------
code
 @Deprecated public boolean isValid(String regex){
  if (regex == null) {
    return false;
  }
  Pattern pattern=Pattern.compile(regex);
  Matcher matcher=pattern.matcher(getText());
  return matcher.matches();
}
Ground Truth: if the main text matches the regex
CodeNN : check the a pattern matches the regex the the matches leaves matches matches the .
HDeepcom : if if this the is the the certain . . . not . . . . . . . . . . . . . . . . . . . . . . . .
Attgru : determine if the assertion with the specified pattern is equal to a percent - value as a valid list identifier name at the moment and port .
NCS :validate the main text matches the regex
Code2seq : returns true if the given string
HybridDrl: check if the text is valid
Astattgru: if the main text matches the regular expression .
CodeAstnn: if the main regex matches the regex
CAST  : if the main text matches the regex
---------------------------------------
code
 public void addRole(String role){
  if (role != null) {
    roles.add(role);
  }
}
Ground Truth: add a role to this user .
CodeNN : adds a role role to this
HDeepcom : adds a alias to this for . . roles this this roles
Attgru : add this message .
NCS :adds the given role to the list of role .
Code2seq : adds a new argument to the
HybridDrl: adds a new role to the list
Astattgru: adds a role of . to indicate that the user already exists .
CodeAstnn: adds a role to this user if the list of those added .
CAST  : adds a role to this user
---------------------------------------
code
 public ForumPostConfig createReply(ForumPostConfig config){
  config.addCredentials(this);
  String xml=POST(this.url + "/create-reply",config.toXML());
  Element root=parse(xml);
  if (root == null) {
    return null;
  }
  try {
    ForumPostConfig reply=new ForumPostConfig();
    reply.parseXML(root);
    return reply;
  }
 catch (  Exception exception) {
    this.exception=SDKException.parseFailure(exception);
    throw this.exception;
  }
}
Ground Truth: create a reply to a forum post . you must set the parent id for the post replying to .
CodeNN : update a reply to a forum post . must the set the parent for post the post post to
HDeepcom : create a new to a to forum . to to to the the the the the the the the the the the the the the the the the the . . . . . . . . . . . . . . . . . . . .
Attgru : create a reply to a forum post . you must first get the " scheduler " button for event . this is used in special patch config .
NCS :create a reply to a reply . you must set the parent id to the post .
Code2seq : parses a new request from the
HybridDrl: create a reply to the forum post
Astattgru: create a reply to a forum post . you ' t include the ' - ' mode , which only will be overwritten by the owner id . you must set the id for a specific pick .
```

```
CodeAstnn: create a reply to a forum post . you must set the post id for the post replying to .
CAST  : create a reply to a forum post . you must set the parent id for the post replying to .
---------------------------------------
code
 protected void updateRadioLinks(){
  m_CVBut.setEnabled(true);
  m_CVText.setEnabled(m_CVBut.isSelected());
  m_CVLab.setEnabled(m_CVBut.isSelected());
  m_SeedText.setEnabled(m_CVBut.isSelected());
  m_SeedLab.setEnabled(m_CVBut.isSelected());
  if (m_AttributeEvaluatorEditor.getValue() instanceof AttributeTransformer) {
    m_CVBut.setSelected(false);
    m_CVBut.setEnabled(false);
    m_CVText.setEnabled(false);
    m_CVLab.setEnabled(false);
    m_SeedText.setEnabled(false);
    m_SeedLab.setEnabled(false);
    m_TrainBut.setSelected(true);
  }
}
Ground Truth: updates the enabled status of the input fields and labels .
CodeNN : displays a enabled of
HDeepcom : updates the the in the the input input selected button editor editor
Attgru : updates the pre - clicked actions of the current event being displayed as a " pressed .
NCS :updates the enabled status of the selected enabled .
Code2seq : updates the state of the current
HybridDrl: called when the buttons is selected
Astattgru: brings by .
CodeAstnn: updates the enabled status of the button panel and labels .
CAST  : updates the enabled state of the input fields and labels .
---------------------------------------
code
 public static String right(String s,int width,char fillChar){
  if (s.length() >= width) {
    return s;
  }
  StringBuffer sb=new StringBuffer(width);
  for (int i=width - s.length(); --i >= 0; ) {
    sb.append(fillChar);
  }
  sb.append(s);
  return sb.toString();
}
Ground Truth: right justify a string .
CodeNN : left justify string string to
HDeepcom : returns a string with a single
Attgru : returns the specified width and char array , and returns the character . if the provided
string is at point not a width or does not specify words .
NCS :fills a string with the given width .
Code2seq : returns a string representation of the
HybridDrl: returns a string representation of the string
Astattgru: forward a string with a whitespace ratio .
CodeAstnn: left justify a string .
CAST  : right justify a string .
---------------------------------------
code
 public static DoubleVector rnorm(int n,double mean,double sd,Random random){
  if (sd < 0.0)   throw new IllegalArgumentException("standard deviation < 0.0");
  if (sd == 0.0)   return new DoubleVector(n,mean);
  DoubleVector v=new DoubleVector(n);
  for (int i=0; i < n; i++)   v.set(i,(random.nextGaussian() + mean) / sd);
  return v;
}
Ground Truth: generates a sample of a normal distribution .
CodeNN : this a normal random of random real normal random
HDeepcom : calculates a mean arc the based
Attgru : gets a object that can be under to ( ) and return .
NCS :generates a random real number from a random vector of n ' s random number .
Code2seq : returns the distance of the problem
HybridDrl: generates a double of the distribution
Astattgru: returns a random number from the distribution .
CodeAstnn: generates a random m of the distribution
CAST  : generates a sample of a chi - square distribution .
---------------------------------------
code
 protected void warningOccurred(int code){
  cbLock.lock();
  try {
    if ((code < 0) || (code > MAX_WARNING)) {
      throw new InternalError("Invalid warning index");
    }

processWarningOccurred("com.sun.imageio.plugins.jpeg.JPEGImageReaderResources",Integer.toString(code))
;
  }
  finally {
    cbLock.unlock();
  }
}
Ground Truth: called by the native code or other classes to signal a warning . the code is used to
lookup a localized message to be used when sending warnings to listeners .
CodeNN : copy by the machine for debugging . . caller signal warning
HDeepcom : called by single object releases only code releases releases by by by message message
message message message message message message message message message message message message
message message . .
Attgru : check if the requested time has finished loading with its aliases . this form a callback that
isn ' t done using split .
NCS :called by the native code or other classes to signal a warning message to be used to lookup a
warning or other classes .
Code2seq : called when the given thread and
HybridDrl: adds a code to the object
Astattgru: this method gets called when a token has been override
CodeAstnn: called by the native code or other classes to signal a warning . the code is used to lookup
a localized message to listeners .
CAST  : called by the native code or other classes to signal a warning . the code is used to lookup a
localized message to be used when sending warnings to listeners .
---------------------------------------
code
 private void calculateD(double[] b){
  int length=mExtremalIndices.size() - 1;
  mD=new double[length];
  for (int k=0; k < length; k++) {
    mD[k]=b[k] * (mGrid.getCosineFrequencyGrid()[mExtremalIndices.get(k)] -
mGrid.getCosineFrequencyGrid()[mExtremalIndices.get(length)]);
  }
}
Ground Truth: calculates the set of d values for the current extremal index set . implements oppenheim
/ schafer discrete time signal processing , 3e , 2016 , equation 116c
CodeNN : the gamma of two
HDeepcom : calculates the indices step number the the . .
Attgru : calculate ( between * n ) step for each point in the split , and the scores for to the same .
NCS :calculates the value of the set of extremal indices to be relatively using the standard error
```

signal processing . this signal uses the standard error signal counts , how often / schafer , which might be set to get the value of a corresponding frequency signal processing / schafer to

Code2seq : returns the minimum from the given

HybridDrl: computes the maximum of the extremal index of the extremal index

Astattgru: calculate the i - dependent time - in frequency grid for integer sorting .

CodeAstnn: calculates the value of b processing by extremal the set of extremal indices in the frequency set of extremal indices .

CAST : calculates the value of b across the set of extremal indices . implements oppenheim / schafer discrete time signal processing , 3e , 2016 , equation 115

----

code

```
 public double adjustedPow10(double val){
  boolean negFlag=(val < 0.0);
  if (negFlag) {
    val=-val;
  }
  double res;
  if (val < 1.0) {
    res=(Math.pow(10,val + 1.0) - 10.0) / 9.0;
  }
 else {
    res=Math.pow(10,val);
  }
  return negFlag ? (-res) : res;
}
```

Ground Truth: returns an adjusted power of 10 value for graphing purposes . the first adjustment is that negative values are changed to positive during the calculations , and then the answer is negated at the end . the second is that , for values less than 1 , a progressive logarithmic offset is subtracted such that at 0 the returned result is also 0 .

CodeNN : returns the adjusted value value for adjusted based direction the positive adjustment is negative into positive positive attempt . . . . . if answer answer " " calculations is is is is value value value value value value value value is is is is below otherwise is below

HDeepcom : returns the double into respect double range coordinate . . . . adjustment range range range range range . . . . negated negated negated negated negated negated negated negated negated negated negated negated negated negated negated negated negated

Attgru : convert a ipv4 first , digits to a character . < p > this method creates a number of times to see if it has value (with " numerical . i . e ") = - value " (" - > ") - linear

NCS :returns an adjusted log10 value for graphing purposes . the first adjustment is that negative values are stored in the positive numbers , and then the answer is not accurate to the calculations .

Code2seq : returns the distance of the given

HybridDrl: returns the value of the given value

Astattgru: calculates an adjusted log10 by a normal value .

CodeAstnn: returns a calculations value , positive the first adjustment between 0 and the calculations is changed to positive (0 , 0) . the answer is always an increasingly large function , and the first is that , and then the answer is the upper positive at the first

CAST : returns an adjusted log10 value for graphing purposes . the first adjustment is that negative values are changed to positive during the calculations , and then the answer is negated at the end . the second is that , for values less than 10 , an increasingly large (0

----

code

```
 private void fetchMyServiceCardsFromServer(){
  RetroCallback retroCallback;
  retroCallback=new RetroCallback(this);
  retroCallback.setRequestId(HttpConstants.ApiResponseCodes.GET_MY_SERVICES);
  retroCallbackList.add(retroCallback);
  mYeloApi.getMyServiceCards(retroCallback);
}
```

Ground Truth: helper http function to get list of my service cards .

CodeNN : add http call callback get filter - my service cards cards .

HDeepcom : api this operation during configuration waiting waiting waiting - ' ' received received received . . . . . received received received received received received received received . . . . . .

```
. . .
Attgru : this method removes a server state .
NCS :helper http function to fetch list of my service cards
Code2seq : initialize the server manager
HybridDrl: this method calls the method
Astattgru: helper http function to clone the service
CodeAstnn: helper http function call fetches service cards
CAST  : helper http function to get list of my service cards .
--------------------------------------
code
 public List<ColourChange> sampleConditionalMigrationEvents2(int parentColour,double parentHeight,int 
childColour,double childHeight,double[] m){
  List<ColourChange> colourChanges=new ArrayList<ColourChange>();
  if (parentHeight < childHeight) {
    throw new IllegalArgumentException("sampleConditionalMigrationEvents: parentHeight=" + 
parentHeight + " childHeight="+ childHeight+ ", not good.");
  }
  try {
    int currentColour=parentColour;
    double currentHeight=parentHeight;
    while (true) {
      ColourChange 
nextEvent=randomConditionalMigrationEvent(currentColour,currentHeight,childColour,childHeight,m);
      currentHeight=nextEvent.getTime();
      currentColour=nextEvent.getColourAbove();
      colourChanges.add(nextEvent);
    }
  }
 catch (  NoEventException nee) {
  }
  reverseColourChangeList(colourChanges,parentColour);
  return colourChanges;
}
Ground Truth: samples migration events on a two - coloured branch , conditional on colours at both 
ends migration process is forwards in ( natural ) time , so we are going down the tree . returns a 
list of events , ordered forward in time ( i . e . colour refers to branch * below * the event ) *
CodeNN : calculate the of that a and bit the % is event and and unbalanced and on recorded two . . 
test . . . . . . . .
HDeepcom : this a by on on an , here here here , , , , , , the the events events events events events 
events events events events events events events events events events events events events events 
events events events events events events events events events events events events
Attgru : convenience function to signal that the given task is on cell and that they build a rectangle 
. the expected position and edges are taken on single tree using . tries to collect it information in 
a pass of the events .
NCS :samples migration events on a two - ) time out of colours branch is forwards , so time is 
forwards in forwards migration process migration in time ( natural ) time , conditional returns a 
migration time . returns a conditional on both migration time , conditional on whether there
Code2seq : this method will be the layout
HybridDrl: returns a list of y translation of the y translation of the
Astattgru: based on a tree , starting at least one pixel .
CodeAstnn: expands events in a list of events , so ordered , handling an ordered list of events at 
both ends in time .
CAST  : samples migration events on a two - coloured branch , conditional on colours at both ends 
migration process is forwards in ( natural ) time , so we are going down the tree . returns a list of 
events , ordered forward in time ( i . e . colour
--------------------------------------
code
 public final int first_common_layer(BrdItem p_other){
  int max_first_layer=Math.max(first_layer(),p_other.first_layer());
  int min_last_layer=Math.min(last_layer(),p_other.last_layer());
  if (max_first_layer > min_last_layer) {
    return -1;
  }
  return max_first_layer;
}
```

Ground Truth: returns the first layer , where both this item and p_other have a shape . returns - 1 , if such a layer does not exisr .

CodeNN : returns the set layer the the on . vector and p_other . . shape . . . . . .

HDeepcom : returns the layer operation is the the inside it it it it the is is is , , , is is is indicate indicate indicate indicate indicate indicate indicate indicate the . . . .

Attgru : merges this item with the current selection . if the item at the previous most end of other most have the end of the input vector , this operation is the total most - legal call to the caller .

NCS :returns the first layer , where both , or has a layer . returns - 1 if such a layer can be excluded .

Code2seq : returns the height of the view

HybridDrl: returns the last item in this item in this item

Astattgru: returns the smallest (not b) on the right - intersection .

CodeAstnn: returns the last layer , where both this item and p_other have a shape , and p_other . returns - 1 , if both .

CAST : returns the first layer , where both this item and p_other have a shape . returns - 1 , where such a layer does not exisr . returns - 1 , if such a layer does not exisr .

----

code

```
 @Override public boolean lock(){
  try {
    if (!FileFactory.isFileExist(location,FileFactory.getFileType(tmpPath))) {
      FileFactory.mkdirs(location,FileFactory.getFileType(tmpPath));
    }
    String lockFilePath=location + CarbonCommonConstants.FILE_SEPARATOR + lockFile;
    if (!FileFactory.isFileExist(lockFilePath,FileFactory.getFileType(location))) {
      FileFactory.createNewLockFile(lockFilePath,FileFactory.getFileType(location));
    }
    fileOutputStream=new FileOutputStream(lockFilePath);
    channel=fileOutputStream.getChannel();
    try {
      fileLock=channel.tryLock();
    }
 catch (    OverlappingFileLockException e) {
      return false;
    }
    if (null != fileLock) {
      return true;
    }
 else {
      return false;
    }
  }
 catch (  IOException e) {
    return false;
  }
}
```

Ground Truth: lock api for locking of the file channel of the lock file .

CodeNN : lock method new locking object the a channel path destination to it it the database

HDeepcom : lock the and locking the the acquire

Attgru : lock api for locking from .

NCS :lock api for locking of file channel

Code2seq : creates a new entry of the

HybridDrl: load the lock data from the file

Astattgru: file object for high bits

CodeAstnn: lock api for locking of the file of the lock api .

CAST : lock api for locking of the file channel of the api api .

```
---------------------------------------
code
 public void removeEventListener(ZWaveEventListener eventListener){
synchronized (zwaveEventListeners) {
    zwaveEventListeners.remove(eventListener);
  }
}
Ground Truth: remove a listener for zwave events to this controller .
CodeNN : removes a messages events of the . . . .
HDeepcom : removes a listener to the listener . .
Attgru : used to remove a listener to this listener with your overridden to receive an event listener
.
NCS :remove a zwave event listener .
Code2seq : removes the event listener from
HybridDrl: removes a listener from the list
Astattgru: remove a event listener from this event
CodeAstnn: removes a listener from this controller .
CAST  : remove a listener for zwave events .
---------------------------------------
code
 public void resetReferences(){
  if (_refs != null)   _refs.clear();
}
Ground Truth: resets the references for streaming .
CodeNN : removes the list to the variables variables collection collection references variables
variables waiting
HDeepcom : trim all streaming
Attgru : clear the list of players names for .
NCS :resets references for streaming .
Code2seq : removes all the elements from the
HybridDrl: removes the list list
Astattgru: resets the references to its initial value .
CodeAstnn: resets the references for references .
CAST  : resets the references for streaming .
---------------------------------------
code
 public static Request newUploadVideoRequest(Session session,File file,Callback callback) throws
FileNotFoundException {
  ParcelFileDescriptor descriptor=ParcelFileDescriptor.open(file,ParcelFileDescriptor.MODE_READ_ONLY);
  Bundle parameters=new Bundle(1);
  parameters.putParcelable(file.getName(),descriptor);
  return new Request(session,MY_VIDEOS,parameters,HttpMethod.POST,callback);
}
Ground Truth: creates a new request configured to upload a photo to the user ' s default photo album .
the photo will be read from the specified file descriptor .
CodeNN : launch a new request configured to upload video video the user user device device times .
album video video video the the the the file file . . . . . . " " " " " " " " configuration
configuration configuration configuration configuration configuration configuration configuration
HDeepcom : add a new request . the to
Attgru : creates a new request for a download request .
NCS :creates a new request configured to upload a video to upload .
Code2seq : creates a new request to the
HybridDrl: creates a new instance of the given file
Astattgru: creates a speech object from a request .
CodeAstnn: checks a video request configured to upload the video specified by the user ' s video . the
video id will be read from the video and the new video .
CAST  : creates a new request configured to upload a video to the user ' s default video album . the
video will be read from the specified file .
---------------------------------------
code
 protected void encodeLineSuffix(OutputStream aStream) throws IOException {
  pStream.println();
}
```

Ground Truth: encode the suffix that ends every output line . by default this method just prints a < newline > into the output stream .

CodeNN : 109 a encode whether suffix the file stream

HDeepcom : 109 * encode the the followed ends output output * 110 * 110 110 method method method method newline . . . .

Attgru : 109 * encode the suffix that ends every output line . by default 110 this method just prints a new legal split (which the algorithm on source string can ' t be specified in brackets , but as groovy on top after this method pattern after a reference

NCS :109 * prints the suffix that ends with a prefix of the output stream . by default this method just prints a < newline >

Code2seq : writes the contents of the given

HybridDrl: writes a stream to the output stream

Astattgru: 109 * encode the suffix that ends every output line . by default 110 this method just prints a < newline > into an output stream that is not a multiple - state following the following char : just appends the original line suffix to an application that is

CodeAstnn: 109 * encode a newline into the 109 output stream .

CAST : 109 * encode the suffix that ends every output line . by default 110 this method just prints a < newline > into the output stream . 111

----

code

```
 public void testWildcardsDoesNotNeedTwoDots() throws Exception {
  String cert="-----BEGIN CERTIFICATE-----\n" +
"MIIBjDCCATagAwIBAgIJAOVulXCSu6HuMA0GCSqGSIb3DQEBBQUAMBAxDjAMBgNV\n" +
"BAMUBSouY29tMCAXDTEwMTIyMDE2NDkzOFoYDzIxMTAxMTI2MTY0OTM4WjAQMQ4w\n"+
"DAYDVQQDFAUqLmNvbTBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDJd8xqni+h7Iaz\n"+
"ypItivs9kPuiJUqVz+SuJ1C05SFc3PmlRCvwSIfhyD67fHcbMdl+A/LrIjhhKZJe\n"+
"1joO0+pFAgMBAAGjcTBvMB0GA1UdDgQWBBS4Iuzf5w8JdCp+EtBfdFNudf6+YzBA\n"+
"BgNVHSMEOTA3gBS4Iuzf5w8JdCp+EtBfdFNudf6+Y6EUpBIwEDEOMAwGA1UEAxQF\n"+
"Ki5jb22CCQDlbpVwkruh7jAMBgNVHRMEBTADAQH/MA0GCSqGSIb3DQEBBQUAA0EA\n"+
"U6LFxmZr31lFyis2/T68PpjAppc0DpNQuA2m/Y7oTHBDi55Fw6HVHCw3lucuWZ5d\n"+ "qUYo4ES548JdpQtcLrW2sA==\n"+
"-----END CERTIFICATE-----";
  CertificateFactory cf=CertificateFactory.getInstance("X.509");
  InputStream in=new ByteArrayInputStream(cert.getBytes("UTF-8"));
  X509Certificate x509=(X509Certificate)cf.generateCertificate(in);
  mySSLSession session=new mySSLSession(new X509Certificate[]{x509});
  HostnameVerifier verifier=HttpsURLConnection.getDefaultHostnameVerifier();
  assertTrue(verifier.verify("google.com",session));
}
```

Ground Truth: earlier implementations of android ' s hostname verifier required that wildcard names wouldn ' t match " * . com " or similar . this was a nonstandard check that we ' ve since dropped . it is the ca ' s responsibility to not hand out certificates that match so broadly .

CodeNN : earlier implementations certificate ' s verifier default wildcard names wouldn ' ' that * com com com . similar similar is was a a a ' ' ' since since since ' ' is is ' ' s s s s s s s certificates

HDeepcom : earlier implementations of if s ' hostname hostname was supports hostname wouldn wouldn . . . . ' ve ve ve ' ' ' ' ' ' ' ' ' ' ' ' ve the the the the '

Attgru : earlier implementations of android ' s hostname verifier required that wildcard names wouldn ' t match " * . com " or similar . this was a check that we can check on original pool when source is fully - use from the .

NCS :earlier implementations of android ' s hostname verifier required that wildcard names wouldn ' t match " * " * " * " or similar . com " * " , com " * " * 32 " . com " if the nonstandard earlier earlier out , or similar

Code2seq : test the ssl ssl socket

HybridDrl: test the password of the url

Astattgru: earlier implementations of android ' s hostname verifier required that wildcard names wouldn ' t match " .

CodeAstnn: earlier implementations of hostname ' s hostname verifier required that match the ca ' s hostname . since nonstandard names match this hostname is dropped . it ' s responsibility to match the ca and implementations that ' s responsibility must match the ca is required since that ' s

CAST : earlier implementations of android ' s hostname verifier required that wildcard names wouldn ' t match " * . com " or similar . this was a nonstandard check that we ' ve since dropped . it is the ca ' s responsibility to not hand out certificates that match

----

code

```
 private void generateUniformIntegerExamples(Instances format,int
numInstances,SubspaceClusterDefinition cl,String cName){
  Instance example=null;
  int numAtts=m_NumAttributes;
  if (getClassFlag()) {
    numAtts++;
  }
  example=new DenseInstance(numAtts);
  example.setDataset(format);
  boolean[] attributes=cl.getAttributes();
  double[] minValue=cl.getMinValue();
  double[] maxValue=cl.getMaxValue();
  int[] minInt=new int[minValue.length];
  int[] maxInt=new int[maxValue.length];
  int[] intValue=new int[maxValue.length];
  int[] numInt=new int[minValue.length];
  int num=1;
  for (int i=0; i < minValue.length; i++) {
    minInt[i]=(int)Math.ceil(minValue[i]);
    maxInt[i]=(int)Math.floor(maxValue[i]);
    numInt[i]=(maxInt[i] - minInt[i] + 1);
    num=num * numInt[i];
  }
  int numEach=numInstances / num;
  int rest=numInstances - numEach * num;
  for (int i=0; i < m_NumAttributes; i++) {
    if (attributes[i]) {
      example.setValue(i,minInt[i]);
      intValue[i]=minInt[i];
    }
 else {
      example.setMissing(i);
    }
  }
  if (getClassFlag()) {
    example.setClassValue(cName);
  }
  int added=0;
  int attr=0;
  do {
    for (int k=0; k < numEach; k++) {
      format.add(example);
      example=(Instance)example.copy();
      added++;
    }
    if (rest > 0) {
      format.add(example);
      example=(Instance)example.copy();
      added++;
      rest--;
    }
    if (added >= numInstances) {
      break;
    }
    boolean done=false;
    do {
      if (attributes[attr] && (intValue[attr] + 1 <= maxInt[attr])) {
        intValue[attr]++;
        done=true;
      }
 else {
        attr++;
      }
    }
 while (!done);
    example.setValue(attr,intValue[attr]);
  }
 while (added < numInstances);
}
```

Ground Truth: generate examples for a uniform cluster dataset .
CodeNN : generate a to of uniform cluster dataset .
HDeepcom : generates is - to an for
Attgru : generates a new table (for example) .
NCS :generate a simple dataset for the dataset .
Code2seq : this method used to check the
HybridDrl: generate example for the dataset
Astattgru: generate an integer value from the passed in manner .
CodeAstnn: generate all uniform cluster values for the dataset .
CAST : generate examples for uniform cluster dataset .

----

code

```
 public synchronized boolean removeAll(Collection c){
  Object[] elements=getArray();
  int len=elements.length;
  if (len != 0) {
    int newlen=0;
    Object[] temp=new Object[len];
    for (int i=0; i < len; ++i) {
      Object element=elements[i];
      if (!c.contains(element))       temp[newlen++]=element;
    }
    if (newlen != len) {
      setArray(copyOfRange(temp,0,newlen,Object[].class));
      return true;
    }
  }
  return false;
}
```

Ground Truth: removes from this list all of its elements that are contained in the specified collection . this is a particularly expensive operation in this class because of the need for an internal temporary array .
CodeNN : retains union of state in this list that are contained in in specified collection . in other words removes removes removes list list list of of of that contained contained the the the the collection collection
HDeepcom : removes all elements elements elements this this this this this this this this this this this this this this this .
Attgru : removes all instances of the given collection , starting from the specified value .
NCS :removes all of the elements in this list that are contained in the specified collection . this is a particularly expensive operation in this class because of its children that are contained in the specified collection ' s iterator .
Code2seq : returns true if the given object
HybridDrl: removes all of the elements from this list
Astattgru: removes all of the elements from this list . the source is in that .
CodeAstnn: removes from this list all of its elements that are contained in the specified collection .
CAST : removes from this list all of its elements that are contained in the specified collection . this is a particularly expensive operation in this class because of the need for an internal temporary array .

```
---------------------------------------
code
 public Phone createPhone(String value){
  PhoneField phoneImpl=new PhoneField();
  try {
    phoneImpl.setValue(value);
  }
 catch (  SdpException s) {
    s.printStackTrace();
  }
  return phoneImpl;
}
Ground Truth: returns phone object with the specified value .
CodeNN : this a to phonenumber with the phone phone report to the
HDeepcom : create contact with the list . . operator operator operator operator operator operator
operator operator operator operator
Attgru : it is called by the & resources in the configuration
NCS :returns a new string with the specified value of this object .
Code2seq : returns a new value of the
HybridDrl: creates a new value
Astattgru: returns a string with the .
CodeAstnn: creates a phone phone with the specified value .
CAST  : returns bandwidth object with the specified value .
---------------------------------------
code
 private void reloadModelWithNewValue(final Long newValue){
  final long newValueAsPrimitive=newValue == null ? getModelUpdatePeriod() : newValue;
  for (  final PropertyChangeListener listener : modelUpdatePeriodListeners) {
    final PropertyChangeEvent event=new
PropertyChangeEvent(this,"model",getModelUpdatePeriod(),newValueAsPrimitive);
    listener.propertyChange(event);
  }
}
Ground Truth: notify all listeners about cacheperiod property changed . if passed newvalue is null ,
the oldvalue is taken as new value . this is the case when the reloadmodel is invoked .
CodeNN : traverses train property change . . changed . features the be event listeners listeners the
the the
HDeepcom : update all entries to value to
Attgru : the result to be loaded property .
NCS :notify all listeners about cacheperiod with the given value changed . if the piecewise is null ,
the case per is taken . < p > the oldvalue is the case of the property with those updated , the
listeners is taken . this is invoked when the oldvalue is
Code2seq : this method is called when a
HybridDrl: notify the property change of a value
Astattgru: update the new set property names if the value has changed ( or the latest action with the
new values are to force the associated property definition in the object property changed since the
delete operations are ignored if the event has been changed and then re -
CodeAstnn: notify all the listeners which newvalue newvalue in the oldvalue . if the passed newvalue
is null , this is the new value is added .
CAST  : notify all listeners about cacheperiod property changed . if passed newvalue is null , the
oldvalue is taken as new value . this is the case when the reloadcache is invoked .
---------------------------------------
```