«enumeration» ReductionStrategy

NORMAL\_ORDER
APPLICATIVE\_ORDER
CALL\_BY\_NAME
CALL\_BY\_VALUE

wbind»
T -> LambdaTerm
LambdaTermVisitor<T>

## BetaReductionVisitor

- -strategy: ReductionStrategy
- -result: LambdaTerm = null
- -hasReduced: boolean = false
- +BetaReductionVisitor(strategy: ReductionStrategy)
- +visit(node: LambdaApplication)
- +visit(node: LambdaAbstraction)
- +visit(node: LambdaVariable) (
- +getResult(): LambdaTerm

```
if (hasReduced) {
  result = node;
 else {
  if (node.getFirst().isAbstraction()) {
     LambdaAbstraction abstraction = ((LambdaAbstraction) node.getFirst());
     switch (strategy) {
       case NORMAL_ORDER: {
         result = abstraction.getInside().accept(new ApplicationVisitor(abstraction.getColor(), node.getSecond()));
         break;
    hasReduced = true;
   else {
    node.setFirst(node.getFirst().accept(this));
    node.setSecond(node.getSecond().accept(this));
```

```
if (!hasReduced && strategy != CALL_BY_NAME && strategy != CALL_BY_VALUE) {
    node.setInside(node.getInside().accept(this));
}
result = node;
```

result = node;