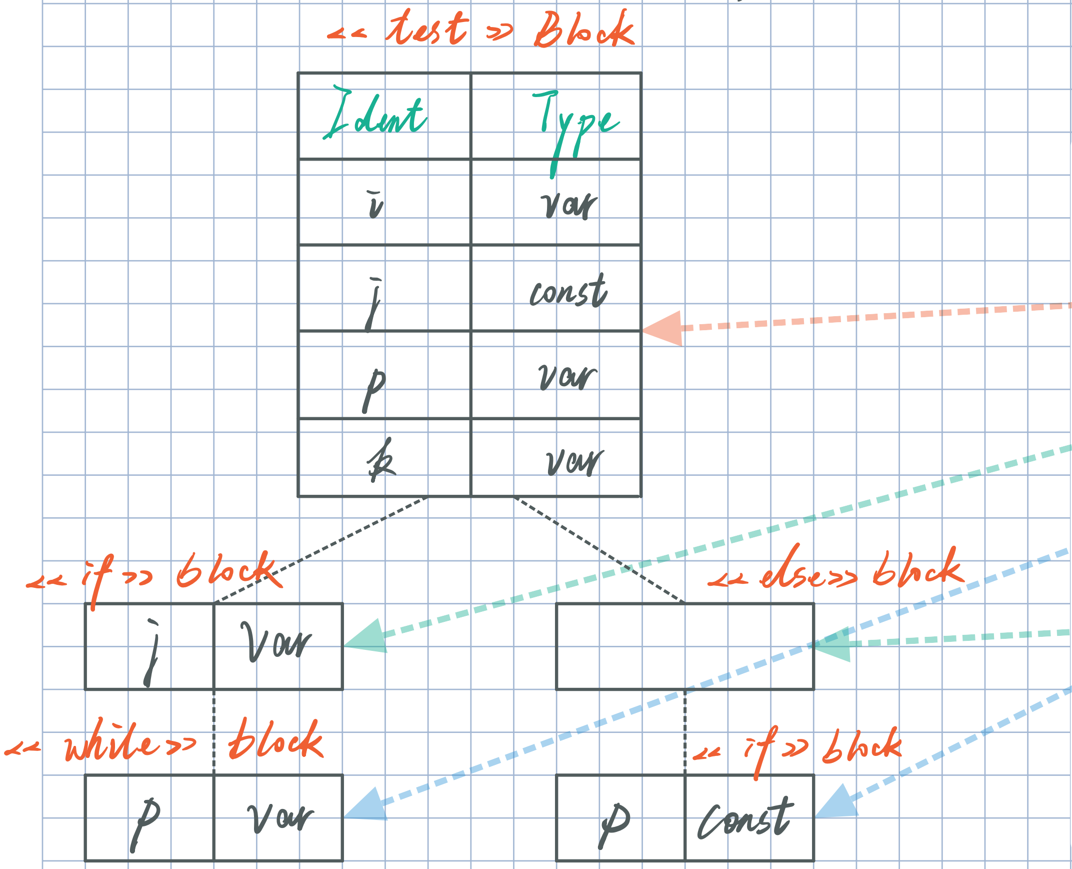
« Sémantique et Traduction de Langage. »

TD4. Table des Symboles

Table des Symboles → Memory

根据第1页的"test"程序写出 Symbol table，如下所示：



« test » Block

Ident	Type
i	var
j	const
p	var
k	var

« if » block

j	Var

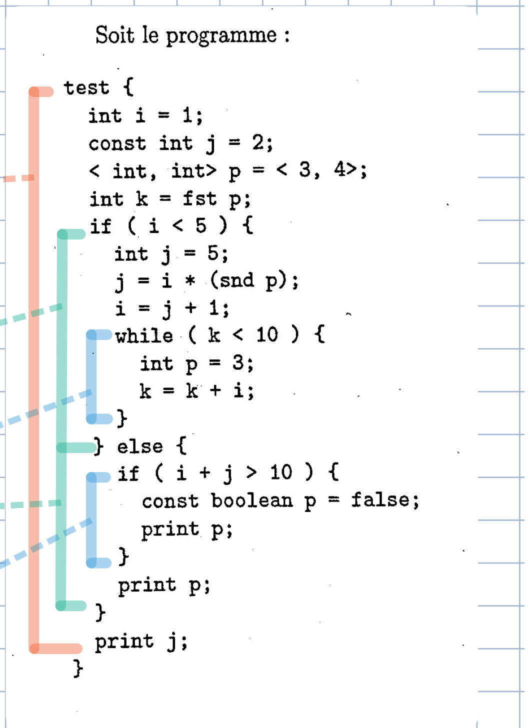
« else » block

« while » block

p	var

« if » block

p	Const



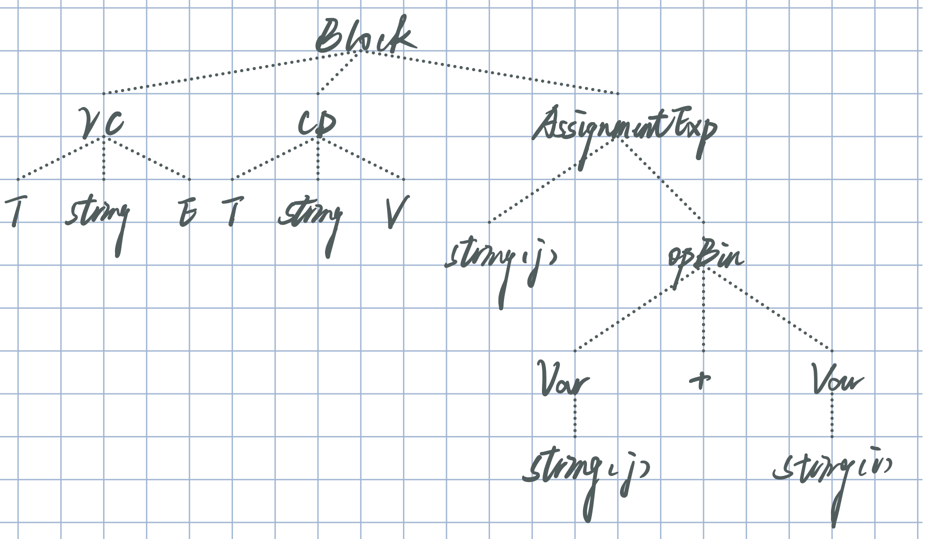
Soit le programme :

```
test {
  int i = 1;
  const int j = 2;
  < int, int> p = < 3, 4>;
  int k = fst p;
  if ( i < 5 ) {
    int j = 5;
    j = i * (snd p);
    i = j + 1;
    while ( k < 10 ) {
      int p = 3;
      k = k + i;
    }
  } else {
    if ( i + j > 10 ) {
      const boolean p = false;
      print p;
    }
    print p;
  }
  print j;
}
```

```
text {
    int j = 0;
    const int i = 1;
    j = j + 1;
}
```

« text » block

j	var
i	var



```
const  Declaration
       tds. add (this. setName( ), const);

Var  Declaration
       tds. add (this. setName ( ), Var);

Fullresolve ( ) { return true }
Collectandpasted Resolve ( ) { // TODO }

public boolean resolve (TDS tds) {
       boolean ok = true;
       TDS tdsblock = new TDS (tds)
       for (Instruction i : this.instruction) {
              ok = ok && i. resolve (tdsblock);
       }
       return OK;
}

        (1)  (2)    (3)
const   int  i  =   10 ;
public boolean resolve (TDS tds) {
     boolean OK1 = this. getType(). resolve (tds);
     boolean OK2 = this. getType(). resolve (tds);
     boolean OK3 = tds. contain (id.txt);
     if (OK3)
     {
        ...
        Error. mean (" All clear ");
     }
```

```
        else
            this . setExpression ( );
        tds . add ( this . setName ( ) , const );
        return  ok1 && ok2 && ok3
    }

   (1)    (2)
    i  =  j + 2 ;

    public boolean resolve (TDS tds) {
          boolean  ok1  =
          - . . -
          if (i < 10)
           {                                  if block
                 "marche"
           }
            else
            {
              "ne marche pas"                 else block
            }
```