

5.1 Typregeln

Gegeben seien die Klassen aus der Vorlesung:

```
class C11 {
  char m1 () {
    int b;
    C12 x = new C12 ()
    return x.m2(x.v, b);
  }
}

class C12 {
  C13 v;
  char m2(c13 v, int w) { ...}
}

class C13 { ...}
```

Leiten Sie für den Block

```
{
  int b;
  C12 x = new C12 ()
  return x.m2(x.v, b);
}
```

mit den Regeln aus der Vorlesung den Typ char her.

5.2 Typchecker

Programmieren Sie die Funktionen

```
typecheckStmt :: Stmt -> [(String, Type)] -> [Class] -> Stmt
```

für das Ergebnis aus Aufgabe 4.2.

Beispiel für If:

```
typecheckStmt :: Stmt -> [(String, Type)] -> [Class] -> Stmt
typecheckStmt (If(be, ifs, Nothing)) symtab cls =
  let
    bexp = typecheckExpr be symtab cls
    ifstmt = typecheckStmt ifs symtab cls
  in
    if ((getTypeFromExpr bexp) == "boolean") then
      TypedStmt(If(bexp, ifstmt, Nothing), getTypeFromStmt ifstmt)
    else
      error "boolean expected"
```

```
getTypeFromExpr :: Expr -> Type
```

```
getTypeFromExpr (TypedExpr(_, typ)) = typ
```

```
getTypeFromStmt :: Stmt -> Type
```

```
getTypeFromStmt (TypedStmt(_, typ)) = typ
```

Hinweis: Für die Expressions müssen Sie eine Funktion

```
typecheckExpr :: Expr -> [(String, Type)] -> [Class] -> Expr
```

programmieren.