string Parser::getCondition

lokale/globale Variablen und Attribute:
begin,end: char*
text,singlelineComment,multilineComment: Boolean
braces: Integer

```
braces ← 1
text ← false
singlelineComment ← false
multilineComment ← false
skipWhitespaces(begin, end)
expect(begin, end, "(")
conditionBegin ← begin
braces ≠ 0
    *begin = '\\' and begin + 2 < end and not multilineComment and not singlelineComment
        wahr:
            begin ← begin + 2
        falsch:
            ∅
    *begin = '\"' || *begin == '\" && !multilineComment && !singlelineComment
        wahr:
            text ← not text
        falsch:
            ∅
    matchWithFollowing(begin, end, "/", '*') or matchWithFollowing(begin, end, "/", '*')
        wahr:
            begin ← begin + 1
            multilineComment ← not multilineComment
        falsch:
            ∅
    matchWithFollowing(begin, end, "/", '/')
        wahr:
            singlelineComment ← true
        falsch:
            ∅
    *begin = '\n'
        wahr:
            singlelineComment ← false
        falsch:
            ∅
    *begin = '(' and not text
        wahr:
            braces ← braces + 1
            ∅
        falsch:
            *begin = ')' and not text
                wahr:
                    braces ← braces - 1
                falsch:
                    ∅
            begin = end
                wahr:
                    throw std::runtime_error("invalid condition")
                falsch:
                    ∅
            braces ≠ 0
                wahr:
                    begin ← begin + 1
                falsch:
                    ∅
conditionEnd ← begin
begin ≠ end
    wahr:
        begin ← + 1
    falsch:
        ∅
skipWhitespaces(begin, end)
return cleanSyntax(conditionBegin,conditionEnd)
```

getName

lokale/globale Variablen und Attribute:
begin,end: char*

```
begin ≠ end and ((*begin ≥ 'A' and *begin ≤ 'Z') or (*begin ≥ 'a' and *begin ≤ 'z') or (*begin ≥ '0' and *begin ≤ '9') or *begin = ':')
    begin ← begin + 1
```