

## parseOperation()

lokale/globale Variablen und Attribute:

name,type,argName,argType: Text

arguments: vector<Argument>

begin,end,retBegin,argBegin: char\*

skip: Boolean&

abstract: Boolean

visibility,stereotype: GZ

match(begin, end, "virtual")		falsch	
skipWhitespaces(begin, end)		∅	
match(begin, end, "explicit")		falsch	
skipWhitespaces(begin, end)		∅	
retBegin ← begin			
begin ≠ end and *begin ≠ '('			
begin ← begin + 1			
*begin ≠ '('		falsch	
throw std::runtime_error("Invalid operation")		∅	
parseTypeAndName(retBegin, begin, name, type)			
begin ← begin + 1			
begin = end		falsch	
throw std::runtime_error("Invalid operation")		∅	
skipWhitespaces(begin, end)			
*begin ≠ ')'			
const char* argBegin ← begin			
begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '='			
begin ← begin + 1			
begin = end		falsch	
throw std::runtime_error("Invalid operation")		∅	
parseTypeAndName(argBegin, begin, argName, argType, true)			
arguments.emplace_back(argName, Type::createFromCppName(argType))			
*begin = '='		falsch	
begin ≠ end and *begin ≠ ',' and *begin ≠ ')'		∅	
begin ← begin + 1			
*begin = ','		falsch	
begin ← begin + 1		∅	
skipWhitespaces(begin, end)			
begin ← begin + 1			
abstract ← false			
begin ≠ end			
skipWhitespaces(begin, end)			
match(begin, end, "override")		falsch	
continue		∅	
match(begin, end, "const")		falsch	
continue		∅	
match(begin, end, "=")		falsch	
skipWhitespaces(begin, end)		∅	
match(begin, end, "0")			
wahr			falsch
abstract ← true			∅
continue			
match(begin, end, "default")			
wahr			falsch
continue			∅
match(begin, end, "delete")			
wahr			falsch
skip ← true			∅
continue			
throw std::runtime_error("Invalid operation")			
stereotype ← Operation::normal			
name = className			falsch
stereotype ← Operation::constructor		∅	
name = "~" + className		falsch	
stereotype ← Operation::destructor		∅	
return Operation(name, Type::createFromCppName(type), arguments, visibility, abstract, stereotype)			