parseTypeAndName	
lokale/globale Variablen und Attribute:	
begin,end,nameBegin: char* name,type: Text&	
argumentMode: Boolean	
skipWhitespaces(begin, end)	
skipWhitespacesBackwards(end, begin)	
L	n = end
wahr beging throw runtime_error("Invalid type and name")	n = end fal
nameBegin ← end	v v
nameBegin-1 ≠ begin and *(nameBegin-1) ≠ ''	
nameBegin ← nameBegin - 1	
wahr nameBeg	in-1 = begin fal
wahr argument Mode falsch	const char* typeEnd ← nameBegin
type ← string(begin, end) name ← string(begin, end)	skipWhitespacesBackwards(typeEnd, begin)
Ø	name ← string(nameBegin, end)
Ø	type ← string(begin, typeEnd)
parseOperation	
lokale/globale Variablen und Attribute:	
name,type: Text	
arguments: vector <argument></argument>	
begin,end,retBegin,argBegin: char* skip: Boolean&	
abstract: Boolean	
visibility: GZ	
argName,argType: Text	
wahr match(begin	, end, "virtual") fal
skipWhitespaces(begin, end)	Ø
match/hegin	and "avaligit")
Walli	Idl
skipWhitespaces(begin, end)	Ø
retBegin ← begin	
begin ≠ end and *begin ≠ '('	
begin ← begin + 1	
wahr *be	gin ≠ '(' fal
throw std::runtime_error("Invalid operation")	Ø
	ž
parseTypeAndName(retBegin, begin, name, type)	
begin ← begin + 1	
wahr begi	n = end fal
throw std::runtime_error("Invalid operation")	Ø
throw std::runtime_error("Invalid operation") skipWhitespaces(begin, end)	Ø
	Ø
skipWhitespaces(begin, end)	Ø
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin	Ø
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '='	Ø
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1	
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr	gin = end fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation")	
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true)	gin = end fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation")	gin = end fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType))	gin = end fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType))	gin = end fal. Ø egin = '=' fal.
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')'	gin = end fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin ← 1	gin = end fal Ø egin = '=' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr	gin = end fal Ø egin = '=' fal Ø pegin = ',' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1	gin = end fal Ø egin = '=' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end)	gin = end fal Ø egin = '=' fal Ø pegin = ',' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 *I wahr *I begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1	gin = end fal Ø egin = '=' fal Ø pegin = ',' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false	gin = end fal Ø egin = '=' fal Ø pegin = ',' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 *I wahr *I begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1	gin = end fal Ø egin = '=' fal Ø pegin = ',' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false	gin = end fal Ø egin = '=' fal Ø pegin = ',' fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end)	gin = end fal egin = '=' fal o begin = ',' fal o fal o fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error('Invalid operation') parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr match(begin ← begin + 1) wahr skipWhitespaces(begin, end)	gin = end
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin ≠ end skipWhitespaces(begin, end) wahr continue match(begin + 1) match(begin + 1) match(begin + 1) wahr match(begin + 2) match(begin	gin = end fal egin = '=' fal pegin = ';' fal o n, end, "override") fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr match(begin + 1) wahr match(begin + 2) match(begin + 2) match(begin + 3)	gin = end fat @ egin = '=' fat @ begin = ';' fat @ n, end, "override") fat @ gin, end, "const") fat
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin ≠ end skipWhitespaces(begin, end) wahr continue wahr match(begin ← continue)	gin = end fat @ egin = '=' fat @ segin = ',' fat @ n, end, "override") fat @ gin, end, "const") fat
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin ≠ end skipWhitespaces(begin, end) wahr continue wahr match(begin ← continue)	gin = end fat @ egin = '=' fat @ begin = ';' fat @ n, end, "override") fat @ gin, end, "const") fat
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr match(begin ← begin + 1 match(begin ← begin + 1 match(begin ← begin + 1 match(begin ← begin	gin = end fal egin = '=' fal pegin = ',' fal pegin = ',' fal gin, end, "override") fal gin, end, "const") fal pegin end "-")
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr match(begin + 1) wahr match(begin + 2) match(begin + 2) match(begin + 3) match(begin, end, "0")	gin = end fal egin = '=' fal egin = ',' fal egin = ',' fal egin, end, "override") fal gin, end, "const") fal egin, end, "const") fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr continue wahr skipWhitespaces(begin, end) match(begin, end, "0") match(begin, end, "0") false	gin = end fal egin = '=' fal egin = ',' fal egin = ',' fal egin, end, "override") fal gin, end, "const") fal egin, end, "const") fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr continue wahr skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) match(begin, end, "0") fals	gin = end fal egin = '=' fal egin = ',' fal egin = ',' fal egin, end, "override") fal gin, end, "const") fal egin, end, "const") fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr continue wahr skipWhitespaces(begin, end) match(begin, end, "0") match(begin, end, "0") false	gin = end fal egin = '=' fal egin = ',' fal egin = ',' fal egin, end, "override") fal gin, end, "const") fal egin, end, "const") fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr skipWhitespaces(begin, end) wahr abstract ← true continue match(begin, end, "0") fals abstract ← true continue	gin = end fal egin = '=' fal pegin = ',' fal pegin = ',' fal pegin, end, "override") fal gin, end, "const") fal pegin, end, "const") fal pegin, end, "=") fal
skipWhitespaces(begin, end) *begin ≠ j'' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin ≠ 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr abstract ← true continue wahr abstract ← true continue match(begin, end, "default") wahr fals fals	gin = end fal egin = '=' fal pegin = ',' fal pegin = ',' fal pegin, end, "override") fal gin, end, "const") fal pegin, end, "const") fal pegin, end, "=") fal
skipWhitespaces(begin, end) *begin ≠]'' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin ≠ end skipWhitespaces(begin, end) wahr continue wahr abstract ← true continue wahr fals abstract ← true continue	gin = end fal egin = '=' fal pegin = ',' fal pegin = ',' fal pegin, end, "override") fal gin, end, "const") fal pegin, end, "const") fal pegin, end, "=") fal
skipWhitespaces(begin, end) *begin ≠ 'j' argBegin ← begin begin ≠ end and *begin ≠ 'j' and *begin ≠ 'j' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ← begin ≠ l skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr abstract ← true continue wahr fals continue wahr continue wahr fals continue	gin = end fal egin = '=' fal obegin = ',' fal obegin, end, "override") fal obegin, end, "const") fal obegin, end, "const") fal obegin, end, "=") fal
skipWhitespaces(begin, end) *begin ≠ ')' argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr abstract ← true continue match(begin, end, "0") match(begin, end, "default") wahr continue match(begin, end, "default") wahr continue match(begin, end, "delete") wahr skip ← true continue match(begin, end, "delete")	gin = end fal egin = '=' fal obegin = ',' fal obegin, end, "override") fal obegin, end, "const") fal obegin, end, "const") fal obegin, end, "=") fal
skipWhitespaces(begin, end) *begin ≠ pi argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr continue wahr abstract ← true continue match(begin, end, "0") fals abstract ← true continue wahr continue match(begin, end, "default") fals skip ← true	gin = end fal egin = '=' fal obegin = ',' fal obegin, end, "override") fal obegin, end, "const") fal obegin, end, "const") fal obegin, end, "=") fal
skipWhitespaces(begin, end) *begin ≠ pegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 wahr begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr abstract ← true continue match(begin, end, "0") fals abstract ← true continue match(begin, end, "default") wahr continue match(begin, end, "delete") fals skip ← true continue	gin = end fal egin = '=' fal obegin = ',' fal obegin, end, "override") fal obegin, end, "const") fal obegin, end, "const") fal obegin, end, "=") fal
skipWhitespaces(begin, end) *begin ≠ pi argBegin ← begin begin ≠ end and *begin ≠ ',' and *begin ≠ ')' and *begin ≠ '=' begin ← begin + 1 wahr throw std::runtime_error("Invalid operation") parseTypeAndName(argBegin, begin, argName, argType, true) arguments.emplace_back(argName, Type::createFromCppName(argType)) wahr begin ≠ end and *begin ≠ ',' and *begin ≠ ')' begin ← begin + 1 skipWhitespaces(begin, end) begin ← begin + 1 abstract ← false begin ≠ end skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr continue wahr skipWhitespaces(begin, end) wahr continue wahr abstract ← true continue match(begin, end, "0") fals abstract ← true continue wahr continue match(begin, end, "default") fals skip ← true	egin = end fat egin = '=' fat pegin = ';' fat pegin, end, "override") fat pegin, end, "const") fat pegin, end, "=") fat fat fat fat fat fat fat fa