

wrapText(QGraphicsSimpleTextItem* inputItem, int maximumWidth)

lokale/globale Variablen und Attribute:

index: std::vector<int>
startingPos: GZ
occurencePos: GZ
startingPosSave: GZ
originalText: Text
newTex: Text
thereAreOccurences: bool
numberOfInserts: GZ
sizeOfString: GZ
sizeOfSubsting: GZ
positionToUse: GZ
diffVector: GZ
diffSave: GZ
aimedPosition: GZ

startingPos ← 0		
startingPosSave ← -1		
coriginalText ← inputItem.text()		
newText ← originalText		
thereAreOccurences ← false		
numberOfInserts ← 1		
(inputItem.boundingRect().width() > maximumWidth) and (numberOfInserts ≤ 2)		
startingPos > startingPosSave		
occurencePos ← originalText.indexOf(QRegExp("[,+->]"), startingPos+1)		
wahr occurencePos ≠ -1 falsch		
index.push_back(occurencePos)	∅	
thereAreOccurences ← true		
startingPosSave ← startingPos		
startingPos ← occurencePos		
wahr thereAreOccurences falsch		
sizeOfString ← originalText.size()	∅	
sizeOfSubsting ← std::round(sizeOfString/(numberOfInserts+1))		
positionToUse ← 0		
diffVector, diffSave		
aimedPosition ← 0		
int x ← 0 x < numberOfInserts x ← x + 1		
aimedPosition ← aimedPosition + sizeOfSubsting		
unsigned int i ← 0 i < index.size() i ← i + 1		
diffVector ← std::abs(aimedPosition - index[i])		
diffSave ← std::abs(aimedPosition - positionToUse)		
diffVector < diffSave		
wahr falsch		
positionToUse ← index[i]		
newText ← newText.insert(positionToUse+1, "\n")		
inputItem.setText(newText)		
numberOfInserts ← numberOfInserts + 1		