

実装について色々検討して案を書いているのはいいのですが、的外れなコメントばかりで大変心配です。実際のプログラムが行なっている処理を把握していないせいだと思います。

プロジェクト演習第4周日個人レポート

744177

上田 晃

・自分の行なった作業

今回、私の担当しているソースコードには変更する所がないと判断し、repeat文のサンプルソースコードの作成を行なった。作成したソースコードは、

- ・通常通りに機能するrepeat.dus
- ・repeat.dusにuntilの条件をなくしたrepeat.dus
- ・repeat.dusにuntilの条件式をずっと偽にするrepeat_forever.dus
- ・repeat_forever.dusにbreak文を追加したrepeat_never.dus
- ・repeat文を2つ用意して二重ループをするrepeat_twoloop.dus
- ・repeat_twoloop.dusの二重ループにbreak文を追加したrepeat_twoloop_break.dus

の6つである。

このうち、repeat_not.dusはuntilの式が無いため実行できないプログラムと考えた。

・実装方針に基づいて説明

実装方針については以下の通りである。

- ・予約語の追加をする
- ・シンボル（トークン）の追加をする
- ・repeatのnodeを作成
- ・repeatのstatementを追加
- ・execute内で実装する

repeat,untilの予約語を追加を行い、シンボルの追加を行うことで、repeat,untilの文字を文として読み込ませる。

untilが読み込めない場合はエラーを示す。

executeで実装を行う。

repeat文はwhile文と似ている構文図なので、while文と同じようにexecute.cで実装をすればうまくいくだろう。 **その通りです。で、なぜ動作しないのだろうか？**

・他2つの拡張機能の実装方針について

他2つは

- ・組込関数
- ・配列の導入

である。

組込関数は、repeat文と同じように予約語の追加、シンボルの追加などを行えば実装が行えるであろう。その際、構文図はサブルーチン呼び出しと同じようなもので、処理をそれぞれ絶対値、大きいほう、小さいほう、乱数値で変えればいいだろう。

単にサブルーチンに repeat-until だけが存在しているのではなく、untilの式の後に別の文（例えば代入文）があっても動作することを確認したり、while文とは異なり、どんな条件でも必ず1回は繰り返しを実行することを確認する例も欲しいですね。

組込関数の名前は予約語にしない、と一番最初に説明しました。で、構文図も書き直す必要はありません。関数なので。

配列の導入では、statements.cの関数varsDeclareList()で変数宣言が行われているため、変数名を読み込んで変数名の中に'['があるならば変数名の数字を別の変数に代入し、malloc関数でその数字分メモリを確保し、その配列が読み込まれるたびにメモリ領域の番号を参照すればいいのではないかと。 **配列ではないグローバル変数、ローカル変数がどのように実装されているかを調べてください。**

・まだ理解できていない部分

(1)前回のレポートのフィードバックで私の担当するソースコードに組込関数の式の評価が直接関係してくるとあったが、どのようにして式の評価をしているのかというのがわからなかった。おそらくexpression.cの関数oppPutOperator()で式の評価を行なっていると思うがこの関数では、こういった流れで式の評価が行われているかよく理解できなかった。

(2)max関数を機能させたいときにはmaxの予約語、sym_maxを追加し、simplestat.cの部分にmaxの処理を付け加えれば機能するのではないかと思う。しかし、拡張するのは関数としての機能のため、function.cに何か変更する部分があるのかが疑問である。function.cではfunc,proc,var,declareの他に関数を読み込んでいるのかどうかはまだよくわかっていない。

oppPutOperator() は二項演算子の評価に関係する部分なので、組込関数には関係ありません。

組込関数が一番関係しそうなのは exptree.c の関数 term() です。これは構文図の「項」と「因子」に相当する部分で、式の中に含まれる関数呼び出しも処理しています。

simplestat.c は「文」の解析を行う部分です。組込関数は「式」なので、このファイルは関係がありません。

組込関数について、大体こんな感じかな、という実装の概要は：

関数のテーブルに普通の関数と大体同じように名前や引数の個数を登録しておきますが、特殊な関数であることを示す情報を付加しておきます。ここに関数の本体はありません。

プログラムの実行時にすべき処理は別途（演算子の実装のように）準備しておく必要があります。

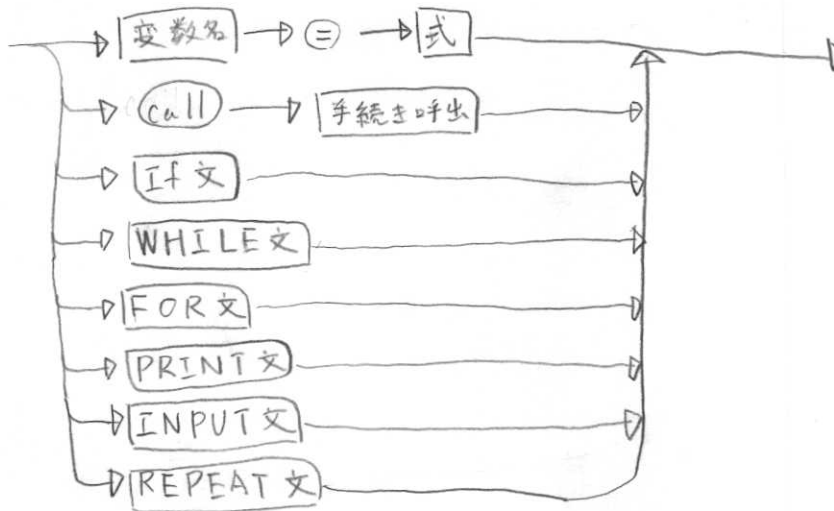
なお、function.c に手を入れる必要はありません。

G07の報告書

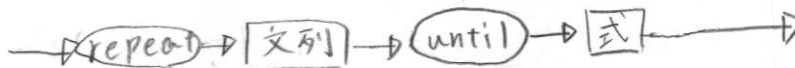
修正した構文図

文

実行文



REPEAT文



(1)repeat文の実装

予約語の追加をする

シンボル (トークン) の追加をする

repeatのnodeを作成

repeatのstatementを追加

execute内で実装する

実装の方針を説明して欲しいのです。

例えば「repeatのnode」とは何？

(4)動作しませんでした

動作しない、だけでは情報が不足。

コンパイルは通ったのか、実行時にエラーが出るのか。

実行はできるが想定と違う動作なのか。

G7