システムデザイン学部

情報科学科　２年

学習番号：２２１４０００３

佐倉仙汰郎

**言語処理系 (字句解析課題問題)**

1. ) go run main.go によりプログラムを実行し 3.14 と入力した際の実行結果

>>go run main.go

Hello sentooooooon! This is the Monkey programming language!

Feel free to type in commands

>> 3.14

{Type:INT Literal:3}

{Type:ILLEGAL Literal:.}

{Type:INT Literal:14}

まず最初のメッセージは、fmt.Printf()でメッセージを表示。user, err := user.Current()でユーザー名（sentooooooon）を表示させている。

3.14と入力すると、トークンが読み込まれ、それぞれの型が変換される。

テキスト

自動的に生成された説明

1. repl/repl.go 11 行目の const PROMPT = ">> " の ">> " を "Input> " に変え るとプログラムを実行した際に、どの部分が変化するかを説明せよ。

main.go を実行したときに表示される、入力を求める文字列が変化する。

1. lexer/lexer test.go の35行目の {token.LET, "let"}, を{token.LET, "LET"}, に変更した後に go test ./lexer を実行せよ。変更前と変更後で結果がどの様 に変化するかを示し、なぜ変化が起きたかを説明せよ。

テキスト

自動的に生成された説明

変更する前はエラーが出ないが変更するとエラーが出る。

1. Monkey 言語では、@ はトークンとして認識されず、ILLEGAL として処理される。 @ を新しく 1 文字トークンとして追加し、lexer で処理される様にするため、以 下の手順でプログラムを変更せよ。

テキスト

自動的に生成された説明

1. Monkey 言語での識別子名には、アルファベットのみ使用可能である（例えば a100 は識別子 a と整数 100 の 2 つのトークンとして処理される）。以下の手順でプロ グラムを変更し、2 文字以降には数字文字も使用可能な様にせよ。

テキスト

自動的に生成された説明

*func* (l \*Lexer) *readIdentifier*() string {

    position := l.position

*for* isLetterNum(l.ch) {

        l.readChar()

    }

*return* l.input[position:l.position]

}

*func* *isLetterNum*(ch byte) bool {

*return* isLetter(ch) || isDigit(ch)

}

1. 2 文字トークン "++" を追加して lexer が認識するようにプログラムを変更せ よ。トークンタイプは++ とする。（token/token.go に定数を追加する際には、 PLUSPLUS = "++" などとする。）

テキスト

自動的に生成された説明

*case* '+':

*if* l.peekChar() == '+' {

            ch := l.ch

            l.readChar()

            literal := string(ch) + string(l.ch)

            tok = token.Token{Type: token.PLUSPLUS, Literal: literal}

        } *else* {

            tok = newToken(token.PLUS, l.ch)

        }

1. 関数を記述する際のキーワードを "fn" に加え、 "function"　でも可能となるよ うにしたい。キーワードに "function" を追加し、出力するトークンは "fn" と 同じとなるようにプログラムを変更せよ。

テキスト

自動的に生成された説明

*switch* l.readIdentifier() {

*case* "fn", "function":

*return* token.Token{Type: token.FUNCTION, Literal: "fn"}

        }

1. 整数リテラルとして、16 進数 (0x35AC など) が利用可能となるようにプログラム を変更せよ。
2. *func* (l \*Lexer) *readNumber*() string {
3. position := l.position
4. isHex := false
5. *for* {
6. *if* isHex && !isHexDigit(l.ch) {
7. *break*
8. }
9. *if* !isHex && !isDigit(l.ch) {
10. *break*
11. }
12. l.readChar()
13. *if* l.ch == 'x' {
14. isHex = true
15. }
16. }
17. *return* l.input[position:l.position]
18. }

*func* *isHexDigit*(ch byte) bool {

*return* '0' <= ch && ch <= '9' || 'a' <= ch && ch <= 'f' || 'A' <= ch && ch <= 'F'

}

{token.INT, "0x35AC"},