

2021 年度形式言語とコンパイラレポート課題

1 課題 1 Quartz 言語の実現

本講義で実装したプログラミング言語 Quartz をテストするにあたって作成したテストファイルの身とそれらの実行結果を本章に示す.

a) 加減算, 単項加減算演算子

```
1. puts 1+1;
2. puts 1+1+1;
3. puts (1+3)/2;
4. puts +2;
5. puts -3;
6. puts --10;
7. puts 10--10;
```

図 1 任意個の加減算と単項加減算演算子のテスト

図 1 の実行結果を図 2 に示す.

```
[yui@HTT src]$ java Quartz ../tests/1-plusminus.txt
2
3
2
2
-3
10
20
```

図 2 図 1 の実行結果

b) 代入

```
8. a = 2;
9. puts a;
10. puts a + 3;
11. a = b = c = 10;
12. puts a;
13. puts b;
14. puts c;
```

図 3 代入のテスト

図 3 の実行結果を図 4 に示す.

```
[yui@HTT src]$ java Quartz ../tests/2-dainyu.txt  
2  
5  
10  
10  
10
```

図 4 図 3 の実行結果

c) 論理演算子, 比較演算子

```
15. puts 0||0;
16. puts 0||1;
17. puts 1||0;
18. puts 1||1;
19.
20. puts "-----";
21. puts 0&&0;
22. puts 0&&1;
23. puts 1&&0;
24. puts 1&&1;
25.
26. puts "-----";
27. puts 0==0;
28. puts 0==1;
29. puts 1==0;
30. puts 1==1;
31.
32. puts "-----";
33. puts 0<0;
34. puts 0<1;
35. puts 1<0;
36. puts 1<1;
37.
38. puts "-----";
39. puts 0>0;
40. puts 0>1;
41. puts 1>0;
42. puts 1>1;
43.
44. puts "-----";
45. puts 0<=0;
46. puts 0<=1;
47. puts 1<=0;
48. puts 1<=1;
49.
50. puts "-----";
51. puts 0>=0;
52. puts 0>=1;
53. puts 1>=0;
54. puts 1>=1;
```

図 5 論理演算子と比較演算子のテスト

図 5 の実行結果を図 6 に示す。

```

[yui@HTT src]$ java Quartz ../tests/3-hikaku.txt
0
1
1
1
-----
0
0
0
1
-----
1
0
0
1
-----
0
1
0
0
-----
0
0
1
0
-----
1
1
0
1
-----
1
0
1
1

```

図6 図5の実行結果

d) 条件分岐構文

```

55. x = 1;
56. y = 0;
57. if x == 0 then
58.   y = 1;
59. else
60.   y = 2;
61. end
62. puts y;

```

図7 条件分岐構文のテスト

図7の実行結果を図8に示す.

```
[yui@HTT src]$ java Quartz ../tests/4-if-test.txt  
2
```

図8 図7の実行結果

e) 繰り返し構文

```
63. i = 1;  
64. sum = 0;  
65. while i <= 10 do  
66.   sum = sum + i;  
67.   i = i + 1;  
68. end  
69. puts sum;
```

図9 繰り返し構文のテスト

図9の実行結果を図10に示す。

```
[yui@HTT src]$ java Quartz ../tests/5-while-test.txt  
55
```

図10 図9の実行結果

f) 条件分岐構文と繰り返し構文

```
70. odd_num = even_num = 0;  
71.   i = 0;  
72. while i <= 10 do #comment  
73.   if i % 2 == 0 then  
74.     even_num = even_num + 1;  
75.   else  
76.     odd_num = odd_num + 1;  
77.   end  
78.   i = i + 1;  
79. end  
80. puts odd_num;  
81. puts even_num;
```

図11 条件分岐構文と繰り返し構文を合わせたプログラムのテスト

図11の実行結果を図12に示す。

```
[yui@HTT src]$ java Quartz ../tests/6-while-test2.txt  
5  
6
```

図 12 図 11 の実行結果

g) 文字列

```
82. a = "hello";  
83. b = "kin iro";  
84. c = "mozaic";  
85.  
86. puts a + " " + b + " " + c;
```

図 13 文字列の代入, 結合, 表示のテスト

図 13 の実行結果を図 14 に示す.

```
[yui@HTT src]$ java Quartz ../tests/7-string.txt  
hello kin iro mozaic
```

図 14 図 13 の実行結果

h) コメント

```
87. # puts "hello";
```

図 15 コメントのテスト

図 15 の実行結果を図 16 に示す.

```
[yui@HTT src]$ java Quartz ../tests/8-comment.txt
```

図 16 図 15 の実行結果

i) 関数

```
88. def fib(n)
89. if n < 2 then
90. n;
91. else
92. fib(n-1) + fib(n-2);
93. end
94. end
95. i = 0;
96. while i <= 10 do
97. puts "fib(" + i + ") = " + fib(i);
98. i = i + 1;
99. end
```

図 17 関数定義と関数呼び出しのテスト

図 17 の実行結果を図 18 に示す。

```
[yui@HTT src]$ java Quartz ../tests/9-fibonacci.txt
fib(0) = 0
fib(1) = 1
fib(2) = 1
fib(3) = 2
fib(4) = 3
fib(5) = 5
fib(6) = 8
fib(7) = 13
fib(8) = 21
fib(9) = 34
```

図 18 図 17 の実行結果

2 クロージャの理解

本章で書き記すことは無い。

3 配列の実現

a) 配列の代入と表示

```
100.a = [1,2,3];
101.puts a;
102.
103.puts a[0];
104.puts a[1];
105.puts a[2];
106.
107.w = [{"one", 1}, {"two", 2}];
108.puts w;
109.puts w[1];
110.
111.v = [[["deep"]]];
112.puts v[0];
113.puts v[0][0];
114.puts v[0][0][0];
115.puts v[0][0][0][0];
```

図 19 配列の代入と表示のテスト

図 19 の実行結果を図 20 に示す。

```
[yui@HTT src]$ java Quartz ../tests/12-arraytest.txt
[ 1, 2, 3, ]
1
2
3
[ [ one, 1, ], [ two, 2, ], ]
[ two, 2, ]
[ [ [ deep, ], ], ]
[ [ deep, ], ]
[ deep, ]
deep
```

図 20 図 19 の実行結果

4 配列操作の実現

a) 配列操作


```
116. a = [1, 2, 3, 4, 5];  
117. b = [7, 6, 4, 2];  
118. puts a + b;  
119. puts a - b;  
120. puts a * b;
```

図 21 配列操作のテスト

図 21 の実行結果を図 22 に示す。

```
[yui@HTT src]$ java Quartz ../tests/13-arraytest2.txt  
[ 1, 2, 3, 4, 5, 7, 6, 4, 2, ]  
[ 1, 3, 5, ]  
[ 2, 4, ]
```

図 22 図 21 の実行結果

5 for 文の実現

a) for 文

```
121. for i in [1, 2, 3] do  
122. for j in ["a", "b", "c"] do  
123. puts i + "," + j;  
124. end  
125. end  
126. puts i;
```

図 23 for 文のテスト

図 23 の実行結果を図 24 に示す。

```
[yui@HTT src]$ java Quartz ../tests/14-for-test.txt  
1,a  
1,b  
1,c  
2,a  
2,b  
2,c  
3,a  
3,b  
3,c
```

図 24 図 23 の実行結果

6 クラスとオブジェクトの実現

a) クラスとオブジェクト

```
127.class Complex
128.   re = 0;
129.   im = 0;
130.
131.   def initialize(r, i)
132.       re = r;
133.       im = i;
134.   end
135.
136.   def add(c)
137.       Complex.new(re + c.get_re(), im + c.get_im());
138.   end
139.
140.   def add2()
141.       Complex.new(re, im);
142.   end
143.
144.   def to_s()
145.       re + " + " + im + "i";
146.   end
147.
148.   def show()
149.       env
150.   end
151.
152.   def get_re()
153.       re;
154.   end
155.
156.   def get_im()
157.       im;
158.   end
159.end
160.
161.c1 = Complex.new(10, 20);
162.#c1.show();
163.c2 = Complex.new(30, 40);
164.c3 = c1.add(c2);
165.puts c3.to_s();
```

図 25 for 文のテスト

図 25 の実行結果を図 26 に示す。

```
[yui@HTT src]$ java Quartz ../tests/15-class-test.txt  
40 + 60i
```

図 26 図 25 の実行結果

7 独自機能の実現

a) 変数テーブルの表示機能

現在定義されている変数・関数・クラスの一覧を表示する命令”env”を実装した。図 27 に使用例を示す。また、図 27 の実行結果を図 28 に示す。

167 行目の”env”ではそれまでに定義されている a:10 のみ表示される。一方で、170 行目では局所変数 m,n に加えグローバルに定義されていた a と print 関数も表示される。

```
166.a = 10;  
167.env  
168.  
169.def print(n, m)  
170.    env  
171.    puts n;  
172.end  
173.print(10, 20);
```

図 27 変数テーブルの表示機能を仕様するプログラム

```
[yui@HTT src]$ java Quartz ../tests/16-fun-print-test.txt  
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  
a : 10  
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^  
^^^^^^^^^^^^^^^^^^^^^OUTER^^^^^^^^^^^^^^^^^^^^  
a : 10  
print : Function@3abfe836  
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  
m : 20  
n : 10  
vvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvv  
10
```

図 28 図 27 の実行結果