

『プログラム言語論』 期末試験

解答用紙は2枚であり、表と裏の両方を使用できる。(2枚でおさまらないときは、もう1枚を要求してもよい。なお、下書き用紙は回収しないので、そちらに答案を書かないこと。)すべての解答用紙の上部に、学籍番号と氏名を明記すること。

問題用紙と下書き用紙は持ち帰ってよい。問題の順番通りに解答する必要はないが、必ず、問題番号(「1-a」など)を記載してから解答を書くこと。

問 1. (配点 40 点+ボーナス加点 5 点) MiniML 言語の具体構文(あるいは、OCaml 言語)で書かれた次の2つのプログラムについて考える。

```
let x = 10 in
let f y = y * y + y in
  f (f x)
```

```
let x = 3 in
let f y = x + y in
let g y = y * (f y) in
let x = 5 in
  g (f x) ;;
```

上記の2つのプログラムを以下のそれぞれの方式で実行したとき、プログラムが返す値を示しなさい。また、それぞれの方式での実行において、関数 f が呼ばれる回数を求めなさい。

1-a. 静的束縛かつ値呼び

1-b. 静的束縛かつ名前呼び

1-c. 静的束縛かつ必要呼び

1-d. 動的束縛かつ値呼び

1-e. この授業の演習で利用した MiniML 言語の処理系(インタープリタ)は、上記 1-a (静的束縛かつ値呼び)の実行方式であり、また、環境を受け渡す方式で実装をしている。このことを念頭において、上記の1つ目のプログラムを、1-a 方式で実行する過程で、環境がどのように変化するかを書きなさい。(環境が変化するとに記載しなさい。)

なお、環境の表記は、OCaml の文法に従う必要はなく、 $[x=10; y=20]$ や $(x=10, y=20)$ のように、内容がわかる書き方であればよい。

1-f. 前問と同様に、上記の2つ目のプログラムを 1-a 方式で実行する過程で、環境がどのように変化するかを書きなさい。

ここでは、高階関数の処理において、関数クロージャが必要となることに注意せよ。関数クロージャの表記も、OCaml の文法に従う必要はなく、内容がわかる書き方をすればよい。

1-g. この授業の演習で利用した MiniML 言語の処理系(インタープリタ)を、上記 1-d (動的束縛かつ値呼び)の実行方式に変更するために、インタープリタを修正したい。どのような修正をすればよいか、言葉で述べよ。(コードを書く必要はなく、インタープリタのどの部分を、どう修正すればよいかを述べなさい。)

問 2. (配点 30 点+ボーナス加点 6 点)

Java 言語で、2 つのクラス ClassA, ClassB を、以下のように定義する。

```
class ClassA {
    public String toString () { return "ClassA";      }
    public String method1 () { return "Method1";     }
    ClassA ()          {}
}
class ClassB extends ClassA {
    public String toString () { return "ClassB";      }
    public String method2 () { return "Method2";     }
    ClassB ()          {}
}
```

ClassB は ClassA を継承した子クラスである。また、それぞれのクラスは toString というメソッドを持ち、それが呼ばれると、クラス名を返す。そのほかに、method1 と method2 というメソッドが定義されている。

このとき、以下のプログラムを実行したとする。ただし、いくつかの行はコンパイルエラー (型エラーを含む) を起こす可能性があり、その場合は、他の行を試すときは、その行をコメントにしているものとする。

また、new ClassA() は、そのクラスのオブジェクトを新たに 1 つ作る式であり、x.method1() は、変数 x に格納されたオブジェクトに method1 というメソッドを呼び出すものであり、System.out.println は、引数の値を印刷するメソッドである。

```
class Test1 {
    public static void main(String args[]) {
        ClassA x; ClassB y;
        x = new ClassA();           // test1
        System.out.println(x.toString()); // test2
        System.out.println(x.method1()); // test3
        System.out.println(x.method2()); // test4
        y = x;                       // test5
        System.out.println(y.toString()); // test6
        System.out.println(y.method1()); // test7
        System.out.println(y.method2()); // test8
        y = new ClassB();           // test9
        System.out.println(y.toString()); // test10
        System.out.println(y.method1()); // test11
        System.out.println(y.method2()); // test12
    }
}
```

上記の test1 から test12 について、(1) その行を含めるとコンパイルエラーになるかどうか、(2) コンパイルエラーにならない場合、実行すると実行時エラーになるかどうか、(3) エラーがない場合、実行すると何が印刷されるかを答えなさい。ただし、test1, test5, test9 は印刷文ではないので、(3) を答える必要はない。また、たとえば、test1 がコンパイルエラーの場合、test2 から test4 まではテストできないので、それらも「コンパイルエラー」に分類せよ。

ヒント: Java の型付けは静的であり、メソッド呼び出しは動的ルックアップを用いる。

問 3. (配点 30 点+ボーナス加点 6 点)

以下の設問のうち 5 個以上 (1 個を除いて)、答えなさい。解答の分量の目安は、問題 1 つごとに 5 行である (最低 3 行とする)。

- 4-a. インタープリタとコンパイラの定義を述べよ。また、多くの場合、インタープリタで実行するより、コンパイルしてから実行する方が高速であるが、その理由の主なものはなにか。
- 4-b. 末尾再帰とは何か述べよ。また、いくつかのプログラム言語では、末尾再帰で記述した方が、非末尾再帰での記述よりも有利である。その理由の主なものはなにか。
- 4-c. 静的型付けと動的型付けの定義の違い (特にそれぞれの利点と欠点)、また、型検査と型推論の定義の違いを述べよ。
- 4-d. 多相型とは何か述べよ。(ここでは、一番基本的な多相、つまり、パラメータ多相について答えればよい。オブジェクト指向言語におけるサブタイプ多相や、アドホック多相について答える必要はない。) 多相型を用いることによるプログラミング上の利点と、利用例を 1 つ述べよ。
ヒント: 「利用例」というのは、たとえば、ソートアルゴリズムで多相型をどのように使うと良いか、利用シーンを 1 つ想定して書くとよい。
- 4-e. 情報の隠蔽 (情報のカプセル化、抽象化) とは何か述べよ。また、そのような考えがなぜ有用であるのか、現在どのように利用されているか、などを論じよ。
- 4-f. 静的な情報と動的な情報の違いの概要を答えなさい。また、次のうち、「静的に決定できる情報」がどれか答えなさい。(1) Java 言語あるいは C 言語において、プログラムで宣言されている変数がすべて、プログラムの中で一回以上出現するか? (2) Java 言語あるいは C 言語において、プログラムで宣言されている変数がすべて、プログラムの中で実際に一回以上使われているか? (3) OCaml 言語 (あるいは MiniML 言語) で、プログラムに出現する変数がすべて、int 型を持つか?

以上.