

# 2025 年 ソフトウェア演習 2B

## 第 1 回課題

### 1 課題 1

以下の仕様を満たすクラス Message を実装し、動作確認を行いなさい。以下の仕様を満たしていれば、コンストラクタやその他のメンバ関数の実装については自由に行って良い。

- メンバ変数として、public ではない char 型のポインタ変数 message を持つ
- 引数を取らないデフォルトコンストラクタと、char 型のポインタ変数を引数に持つコンストラクタを持つ
- 引数として char 型のポインタ変数で文字列を与え、メンバ変数 message にコピーする関数をメンバ関数として持つ
- メンバ変数 message の値 (char 型のポインタ) を戻り値として返す関数をメンバ関数として持つ

#### リスト 1 課題 1 の動作確認用ソースコードの例

```
1 #include <iostream>
2 #include "Message.h"
3
4 int main (int argc, char *argv[]) {
5     Message obj;
6     obj.setMessage("Hello World.");
7     std::cout << obj.getMessage() << std::endl;
8
9     return 0;
10 }
```

#### 課題のねらい

- クラスの実装を復習する

#### 採点基準

- 課題の仕様を満たしている (20 点)
- クラスの汎用性があるか (クラスとして必要な機能を実装しているかなど) (10 点)
- 動作確認の様子を示しているか (5 点)

### 2 課題 2

課題 1 で実装したクラスに対して抽出演算子 (>>) と挿入演算子 (<<) を実装して、動作確認を行いなさい。

#### リスト 2 課題 2 の動作確認用ソースコードの例

```
1 #include <iostream>
2 #include "Message.h"
3
4 int main (int argc, char *argv[]) {
5     Message obj;
6     std::cout << "Input message:";
```

```

7  std::cin >> obj;
8  std::cout << "Output_message:" << std::endl;
9  std::cout << obj << std::endl;
10
11  return 0;
12 }

```

### 課題のねらい

- クラスに対する演算子の実装を復習する
- 変数の参照渡しについて理解する

### 採点基準

- 課題の仕様を満たしている (10 点 ×2)
- 動作確認の様子を示しているか (5 点)

## 3 課題 3

Message クラスを継承して、以下の仕様を満たすクラス RepeatMessage を実装して、動作確認を行いなさい。以下の仕様を満たしていれば、コンストラクタやその他のメンバ関数の実装については自由に行って良い。

- (クラス Message の) メンバ変数 message の値を表示する回数を表すメンバ変数 nloops を持つ
- 変数 nloops の値だけ変数 message の値を表示する挿入演算子 (<<) を持つ

### リスト 3 課題 3 の動作確認用ソースコードの例

```

1  #include <iostream>
2  #include "RepeatMessage.h"
3
4  int main (int argc, char *argv[]) {
5      RepeatMessage obj(3);
6      std::cout << "Input_message: ";
7      std::cin >> obj;
8      std::cout << "Output_message:" << std::endl;
9      std::cout << obj;
10
11  return 0;
12 }

```

### 課題のねらい

- クラスの継承について復習する

### 採点基準

- 課題の仕様を満たしている (20 点)
- クラスの汎用性があるか (クラスとして必要な機能を実装しているかなど) (5 点)
- 動作確認の様子を示しているか (5 点)

## 自己チェック項目

以下の項目について、1 から 4 までの 4 段階で自己評価しなさい。

4. 十分に理解した 3. 少し不安が残るが理解した 2. 十分には理解できていない 1. まったく理解できない

- ☐ クラスの実装の仕方を理解した。
- ☐ private, protected, public などのアクセス指定子の意味と使い方を理解した。
- ☐ コンストラクタ, デストラクタの実装方法を理解した。
- ☐ メンバ変数の実装方法を理解した。
- ☐ クラスに対する演算子の実装方法を理解した。
- ☐ 変数の参照渡しについて理解し、値渡しとの違いを説明できる。
- ☐ クラスの継承について理解し、既存クラスを継承した別のクラスを実装することができる。
- ☐ インデント（字下げ）など、一貫したスタイルでプログラムが書ける。
- ☐ プログラムに適切なコメントを入れることができる。
- ☐ 適切な変数名を用いることができる。