

# 2024 年ソフトウェア演習 2B 報告書

## 第 6, 7 回課題

	学籍番号	氏名
報告者	B243392	ALCANDER IMAWAN

提出日	2024 年 8 月 05 日
-----	-----------------

概要	このプログラムは、BMI (Body Mass Index) を計算するアプリケーションである。また、目標体重とダイエット期間を入力し、目標 BMI、予測される現在の 1 日の消費カロリー、および週ごとのカロリー減少量を計算するプログラム。
----	--

## 解説

最近、自分の体の健康が気になり始めた。去年の健康診断の結果で、BMI が大幅に増えていることに驚き、どうすれば健康な体を手に入れられるかを考えるようになった。ダイエットを決意し、さまざまな方法を調べたが、最終的にダイエットは「カロリーインとカロリーアウト」が基本であるという結論に至った。

しかし、どのくらいカロリーを減らせば良いのかがわからず、困っていた。そこで、このプログラムを考案した。Mifflin-St Jeor 方程式を使用して、1 日の消費カロリーを推定した。そして、1kg の体重減少に必要なカロリーが 7700 kcalであることを考慮し、週あたりのカロリー減少量を計算することにした。

このプログラムは、現在の体重、身長、目標体重、ダイエット期間を入力すると、次の情報を提供する：

1. 現在の BMI
2. 目標の BMI
3. 予測される現在の 1 日の消費カロリー
4. 週あたりのカロリー減少量

これにより、具体的な数値を把握でき、効果的なダイエット計画を立てる手助けとなる。

## 課題の仕様をどのように満たしたか

### 1. 独自に定義したクラスを使用すること

BMI クラスと BMICalculator クラスを定義している。BMI クラスは BMI を計算する機能を持ち、BMICalculator クラスは目標 BMI の計算や 1 日の消費カロリーの予測、週ごとのカロリー減少量の計算を行う。

### 2. 動的ライブラリを作成して使用すること

BMI クラスと BMICalculator クラスの実装は、bmicalc.dll という動的ライブラリとしてコンパイルされている。これにより、他のプログラムからこれらの機能を動的にリンクして使用できる。

### 3. 動的ロードを利用したプラグインを作成して使用すること

PluginInterface という抽象クラスを定義し、このインターフェースを実装する BMICalculatorPlugin クラスを作成している。このプラグインは動的ライブラリとしてコンパイルされ、実行時に動的にロードされる。main.cpp では、LoadLibrary 関数を使って動的にラ

イブラリをロードし、GetProcAddress 関数を使ってプラグインの関数を取得している。この方法により、プラグインの機能を動的に利用することができる。

これらの要素により、プログラムは課題の仕様を完全に満たしている。

## 作成したアプリケーションの工夫したところ

単に BMI を計算するだけではなく、目標体重に達成するためのカロリー減少量を計算する機能も備えている。また、1 日のカロリー摂取量の予測も行うことができる。このため、ユーザーは目標体重を達成するための具体的なカロリー管理が可能となる。

## 残り 30 点のアピール

広範囲のカロリー管理：

このプログラムは、単に BMI を計算するだけでなく、必要なカロリーに関する詳細な情報を提供する。希望する体重に到達するためには、毎週どれだけのカロリーを減らす必要があるのかを計算し、体重管理のための明確な道筋を示してくれる。

パーソナライズされた推奨：

年齢、性別、活動レベルなどの入力を考慮することで、プログラムはパーソナライズされた推奨を提供する。これにより、カロリーの必要量と削減量が個人のプロファイルに合わせて調整され、アドバイスがより適切で実行可能なものになる。

ユーザーフレンドリーなインターフェース：

このプログラムは、ユーザーに必要な情報の入力を促し、わかりやすい出力を提供する。そのため、技術に詳しくなくても、体重や健康を効果的に管理することに関心のある人でも利用しやすい。

教育的価値：

Mifflin-St.Jeor 方程式について学ぶ機会を提供し、それが 1 日に必要なカロリーと体重管理にどのように適用されるかを知することで、実用的な機能性に加えて教育的価値も提供する。

## アプリケーション実行例

```
D:\Toyohashi_Gikadai\TUT_Software_Enshuu2\report67>BMICalculator.exe
現在の体重 (kg) を入力してください: 72.3
身長 (m) を入力してください: 1.645
目標体重 (kg) を入力してください: 65
ダイエット期間 (日) を入力してください: 60
年齢を入力してください: 24
性別を入力してください (M または F): M
活動レベルを入力してください (1.2: Sedentary, 1.375: Lightly active, 1.55: Moderately active, 1.725: Very active, 1.9: Super active): 1.375

現在の BMI は: 26.7182
目標の BMI は: 24.0205
予測される現在の1日の消費カロリーは: 2249.67 kcal
過ごとの必要なカロリー減少量は: 6557.83 kcal
```

図1 男性、24歳、身長164.5cm、体重72.3kg、60日のダイエットの予測

```
D:\Toyohashi_Gikadai\TUT_Software_Enshuu2\report67>BMICalculator.exe
現在の体重 (kg) を入力してください: 55
身長 (m) を入力してください: 1.58
目標体重 (kg) を入力してください: 50
ダイエット期間 (日) を入力してください: 42
年齢を入力してください: 22
性別を入力してください (M または F): F
活動レベルを入力してください (1.2: Sedentary, 1.375: Lightly active, 1.55: Moderately active, 1.725: Very active, 1.9: Super active): 1.375

現在の BMI は: 22.0317
目標の BMI は: 20.0288
予測される現在の1日の消費カロリーは: 1741.44 kcal
過ごとの必要なカロリー減少量は: 6416.67 kcal
```

図2 女性、22歳、身長158cm、体重55kg、42日のダイエットの予測

## ソースコード

BMI.hpp

```
1: #ifndef BMI_HPP
2: #define BMI_HPP
3:
4: class BMI {
5: public:
6:     BMI(double weight, double height);
```

```

7:     double calculate() const;
8:
9: private:
10:     double weight;
11:     double height;
12: };
13:
14: class BMICalculator {
15: public:
16:     BMICalculator(double weight, double height, double desiredWeight, int dietPeriod, int
age, char gender, double activityLevel);
17:     double currentBMI() const;
18:     double desiredBMI() const;
19:     double dailyCalorieConsumption() const;
20:     double weeklyCalorieReduction() const;
21:
22: private:
23:     double weight;
24:     double height;
25:     double desiredWeight;
26:     int dietPeriod;
27:     int age;
28:     char gender;
29:     double activityLevel; // Activity level factor
30: };
31:
32: #endif // BMI_HPP

```

BMI.cpp

```

1: #include "BMI.hpp"
2:
3: BMI::BMI(double weight, double height) : weight(weight), height(height) {}
4:

```

```

5: double BMI::calculate() const {
6:     return weight / (height * height);
7: }
8:
9: BMICalculator::BMICalculator(double weight, double height, double desiredWeight, int
dietPeriod, int age, char gender, double activityLevel)
10:     : weight(weight), height(height), desiredWeight(desiredWeight),
dietPeriod(dietPeriod), age(age), gender(gender), activityLevel(activityLevel) {}
11:
12: double BMICalculator::currentBMI() const {
13:     BMI bmi(weight, height);
14:     return bmi.calculate();
15: }
16:
17: double BMICalculator::desiredBMI() const {
18:     BMI bmi(desiredWeight, height);
19:     return bmi.calculate();
20: }
21:
22: double BMICalculator::dailyCalorieConsumption() const {
23:     double bmr;
24:     if (gender == 'M') {
25:         bmr = 10 * weight + 6.25 * height * 100 - 5 * age + 5;
26:     } else if (gender == 'F') {
27:         bmr = 10 * weight + 6.25 * height * 100 - 5 * age - 161;
28:     } else {
29:         // Handle invalid gender input
30:         return 0;
31:     }
32:     return bmr * activityLevel; // Activity level factor
33: }
34:
35: double BMICalculator::weeklyCalorieReduction() const {

```

```

36:     double totalCaloriesToLose = (weight - desiredWeight) * 7700; // 1kg の体重減少に必要なカロリーは 7700 kcal
37:     return totalCaloriesToLose / (dietPeriod / 7.0); // 週あたりのカロリー減少量
38: }

```

PluginInterface.hpp

```

1: #ifndef PLUGIN_INTERFACE_HPP
2: #define PLUGIN_INTERFACE_HPP
3:
4: class PluginInterface {
5: public:
6:     virtual ~PluginInterface() = default;
7:     virtual double calculate(double weight, double height) = 0;
8:     virtual double calculateDesiredBMI(double weight, double height, double
desiredWeight, int dietPeriod, int age, char gender, double activityLevel) = 0;
9:     virtual double predictDailyCalories(double weight, double height, int age, char gender,
double activityLevel) = 0;
10:    virtual double calculateWeeklyCalorieReduction(double weight, double desiredWeight,
int dietPeriod) = 0;
11: };
12:
13: extern "C" __declspec(dllexport) PluginInterface* create();
14: extern "C" __declspec(dllexport) void destroy(PluginInterface*);
15:
16: #endif // PLUGIN_INTERFACE_HPP

```

BMICalculatorPlugin.hpp

```

1: #ifndef BMI_CALCULATOR_PLUGIN_HPP
2: #define BMI_CALCULATOR_PLUGIN_HPP
3:
4: #include "PluginInterface.hpp"
5: #include "BMI.hpp"
6:

```

```

7: class BMICalculatorPlugin : public PluginInterface {
8: public:
9:     double calculate(double weight, double height) override;
10:    double calculateDesiredBMI(double weight, double height, double desiredWeight, int
dietPeriod, int age, char gender, double activityLevel) override;
11:    double predictDailyCalories(double weight, double height, int age, char gender, double
activityLevel) override;
12:    double calculateWeeklyCalorieReduction(double weight, double desiredWeight, int
dietPeriod) override;
13:
14: private:
15:     BMI* bmiInstance;
16: };
17:
18: extern "C" __declspec(dllexport) PluginInterface* create();
19: extern "C" __declspec(dllexport) void destroy(PluginInterface* plugin);
20:
21: #endif // BMI_CALCULATOR_PLUGIN_HPP

```

BMICalculatorPlugin.cpp

```

1: #include "BMICalculatorPlugin.hpp"
2:
3: double BMICalculatorPlugin::calculate(double weight, double height) {
4:     bmiInstance = new BMI(weight, height);
5:     double result = bmiInstance->calculate();
6:     delete bmiInstance;
7:     return result;
8: }
9:
10: double BMICalculatorPlugin::calculateDesiredBMI(double weight, double height, double
desiredWeight, int dietPeriod, int age, char gender, double activityLevel) {
11:     BMICalculator calculator(weight, height, desiredWeight, dietPeriod, age, gender,
activityLevel);

```



```

12:     return calculator.desiredBMI();
13: }
14:
15: double BMICalculatorPlugin::predictDailyCalories(double weight, double height, int age,
char gender, double activityLevel) {
16:     BMICalculator calculator(weight, height, 0, 0, age, gender, activityLevel); //
desiredWeight と dietPeriod はここでは不要
17:     return calculator.dailyCalorieConsumption();
18: }
19:
20: double BMICalculatorPlugin::calculateWeeklyCalorieReduction(double weight, double
desiredWeight, int dietPeriod) {
21:     BMICalculator calculator(weight, 0, desiredWeight, dietPeriod, 0, '', 0); // height, age,
gender, activityLevel はここでは不要
22:     return calculator.weeklyCalorieReduction();
23: }
24:
25: extern "C" __declspec(dllexport) PluginInterface* create() {
26:     return new BMICalculatorPlugin();
27: }
28:
29: extern "C" __declspec(dllexport) void destroy(PluginInterface* plugin) {
30:     delete plugin;
31: }

```

main.cpp

```

1: #include <iostream>
2: #include <windows.h>
3: #include "PluginInterface.hpp"
4:
5: typedef PluginInterface* (*create_t)();
6: typedef void (*destroy_t)(PluginInterface*);
7:

```

```

8: int main() {
9:     HINSTANCE handle = LoadLibrary(TEXT("bmicalc.dll"));
10:    if (!handle) {
11:        std::cerr << "ライブラリを開けません¥n";
12:        return 1;
13:    }
14:
15:    create_t create_plugin = (create_t) GetProcAddress(handle, "create");
16:    if (!create_plugin) {
17:        std::cerr << "シンボル 'create' を読み込めません¥n";
18:        FreeLibrary(handle);
19:        return 1;
20:    }
21:
22:    destroy_t destroy_plugin = (destroy_t) GetProcAddress(handle, "destroy");
23:    if (!destroy_plugin) {
24:        std::cerr << "シンボル 'destroy' を読み込めません¥n";
25:        FreeLibrary(handle);
26:        return 1;
27:    }
28:
29:    PluginInterface* plugin = create_plugin();
30:
31:    double weight, height, desiredWeight, activityLevel;
32:    int dietPeriod, age;
33:    char gender;
34:
35:    std::cout << "現在の体重 (kg) を入力してください: ";
36:    std::cin >> weight;
37:    std::cout << "身長 (m) を入力してください: ";
38:    std::cin >> height;
39:    std::cout << "目標体重 (kg) を入力してください: ";
40:    std::cin >> desiredWeight;

```

```

41:     std::cout << "ダイエット期間（日）を入力してください: ";
42:     std::cin >> dietPeriod;
43:     std::cout << "年齢を入力してください: ";
44:     std::cin >> age;
45:     std::cout << "性別を入力してください（M または F）: ";
46:     std::cin >> gender;
47:     std::cout << "活動レベルを入力してください（1.2: Sedentary, 1.375: Lightly active,
1.55: Moderately active, 1.725: Very active, 1.9: Super active）: ";
48:     std::cin >> activityLevel;
49:
50:     double bmi = plugin->calculate(weight, height);
51:     double desiredBMI = plugin->calculateDesiredBMI(weight, height, desiredWeight,
dietPeriod, age, gender, activityLevel);
52:     double dailyCalories = plugin->predictDailyCalories(weight, height, age, gender,
activityLevel);
53:     double weeklyCalorieReduction = plugin->calculateWeeklyCalorieReduction(weight,
desiredWeight, dietPeriod);
54:
55:     std::cout << "¥n 現在の BMI は: " << bmi << "¥n";
56:     std::cout << "目標の BMI は: " << desiredBMI << "¥n";
57:     std::cout << "予測される現在の 1 日の消費カロリーは: " << dailyCalories << "
kcal¥n";
58:     std::cout << "週ごとの必要なカロリー減少量は: " << weeklyCalorieReduction << "
kcal¥n";
59:
60:     destroy_plugin(plugin);
61:     FreeLibrary(handle);
62:     return 0;
63: }
64:

```