

プログラム設計 レポート

std::string・ストリーム・静的メンバ

2115 小林恵都

1. シングルバイト文字・ワイド文字・マルチバイト文字について調査し、説明せよ

- シングルバイト文字

文字コードのうち、8ビットので表した文字のこと

- ワイド文字

およびC++における文法要素で、1文字あたりのバイト数を通常より多くしたデータ型のこと

- マルチバイト文字

文字コードのうち、2バイト以上の大きさで表した文字のこと

2. std::string を用いた次のプログラムを完成させよ.

```
// TODO: 必要なヘッダをインクルード
using namespace std;
int main()
{
    // TODO: string オブジェクト str1 を作成
    // TODO: string オブジェクト str2 を作成し"Program Design"で初期化
    // TODO: string オブジェクト str3 を作成
    // TODO: 標準入力から str1 に文字列を入力
    // TODO: str2 の文字列を反転
    // TODO: str3 に str1 と str2 を連結した文字列をセット
    // TODO: str3

    return 0;
}
```

解答

```

#include<iostream>
#include<string>
#include<algorithm>

using namespace std;

int main(){
    string str1;
    string str2 = "Program Design";
    string str3;

    cin >> str1;
    reverse(str1.begin(), str1.end());
    cout << str1 << endl;

    str3 = str1 + str2;

    cout << str3;

    return 0;
}file.close();
}

```

3. 標準入力から数値(double 型)を入力が終わるまで受け取り, ファイルに保存するプログラムを作成せよ。

解答

```

#include<fstream>
#include<iostream>
using namespace std;

int main(){
    ofstream outfile;
    outfile.open("data.txt");

    double dnum;

    cin >> dnum;

    outfile << dnum;

    outfile.close();
}

```

4. 問 3 で作成したファイル"data.txt"を読み込み, 合計を表示するプログラムを作成せよ。

```

#include<fstream>
#include<iostream>
#include<string>
using namespace std;

int main(){
    ifstream infile;
    infile.open("data.txt");

    string x;

    getline(infile, x);

    cout << x ;

    infile.close();

    return 0;
}

```

5. stringstream を用いて、文字列"Department of Electronic and Information Engineering"を単語に分割し、ファイル"output.txt"にその単語を1行に1単語ずつ保存せよ。

解答

```

#include<fstream>
#include<string>
#include<sstream>
using namespace std;

int main(){
    ofstream outfile;
    outfile.open("output.txt");
    stringstream str1;
    str1 << "Department of Electronic and Information Engineering";

    while (!str1.eof()) {
        string text;
        str1 >> text;
        outfile << text << endl;
    }

    return 0;
}

```

6. 次のプログラムを完成せよ。ただし、main 関数内を変更してはいけない。

```

#include <iostream>
using namespace std;
class Point {
    double x, y, z;
};

int main()
{
    Point p;
    cout << "入力: ";
    cin >> p;
    cout << "出力: " << p << endl;
}

```

解答

```

#include <iostream>
using namespace std;

class Point {
    double x, y, z;
public:
    Point(): x(0),y(0),z(0){};
    friend istream& operator>>(istream& i, Point& p){
        i >> p.x >> p.y >> p.z;
        return i;
    }

    friend ostream& operator<<(ostream& o, Point& p) {
        o << p.x << ' ' << p.y << ' ' << p.z;
        return o;
    }
};

int main(){
    Point p;
    cout << "入力: ";
    cin >> p;
    cout << "出力: " << p << endl;
}

```

7. 体内時計の正確さを調べるために、100 分の 1 秒まで計測できるストップウォッチを用いて、目を閉じて 10 秒計測してみた。その結果を以下のファイルに保存した。

```
1 10.27
2 9.40
3 11.17
4 8.97
```

このファイルを読み込み、マニピュレータを用いて以下のように表示せよ。

```
-----
| n|    s|
| 1| 10.27|
| 2|  9.40|
| 3| 11.17|
| 4|  8.97|
-----
```

解答

```
#include<iostream>
#include<fstream>
#include<string>
#include<iomanip>
using namespace std;

int main(){
    ifstream infile;
    infile.open("watch.txt");

    string column1, column2;

    cout << "-----" << endl;

    cout.setf(ios::right, ios::adjustfield);
    cout << "|" << setw(2) << "n" << "|" << setw(7) << "s" << "|" << endl;

    while (!infile.eof())
    {
        infile >> column1 >> column2 ;
        cout << "|" << setw(2) << column1 << "|" << setw(7) << column2 << "|" << endl;
    }

    cout << "-----" << endl;
}
```

8. 次のプログラムについて答えよ。

```
#include<iostream>
#include<string>
using namespace std;
class Pet {
    string name;
public:
    Pet(const string& n) {
        name = n;
    }
    void Name() {
        cout << name << endl;
    }
};
```

(ア) ペットの数の合計を記録するための静的メンバ変数を追加するためにどうすればよいか

解答

```
class Pet {
    string name;
    static int petcount;
```

(イ) ペットの数の合計を返却するための静的メンバ関数を追加するためにどうすればよいか.

```
class Pet {
public:
    static int TotalPet(){
        return petcount;
    }
}
```

(ウ) オブジェクトの増減に対応して、プログラム中のペットの数を正しく数えるための仕組みを追加するためにどうすればよいか.

```
class Pet{  
public:  
    Pet(const string& n) {  
        name = n;  
        petcount++;  
    }  
    void Name() {  
        cout << name << endl;  
        petcount--;  
    }  
}
```