

応用プログラミング A 第 10 回演習問題 関数オーバーロード 解答例

問題 1 (必須) オブジェクト作成と対応するコンストラクタの確認

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

class myclass {
    int x;
public:
    myclass() {
        x = 0;
        cout << "初期化なしのコンストラクタ\n";
    }
    myclass(int n) {
        x = n;
        cout << "初期化ありのコンストラクタ\n";
    }
    int getx() { return x; }
};

int main() {
    myclass o1(0);
    cout << "o1: " << o1.getx() << "\n";
    myclass o2;
    cout << "o2: " << o2.getx() << "\n";
    myclass o3[2];
    cout << "o3[0]: " << o3[0].getx() << "\n";
    cout << "o3[1]: " << o3[1].getx() << "\n";
    myclass o4[2] = {0, 0};
    cout << "o4[0]: " << o4[0].getx() << "\n";
    cout << "o4[1]: " << o4[1].getx() << "\n";
    myclass *o5;
    o5 = new myclass[2];
    if (!o5) {
        cout << "メモリ割り当てエラー\n";
        return 1;
    }
    cout << "o5[0]: " << o5[0].getx() << "\n";
    cout << "o5[1]: " << o5[1].getx() << "\n";

    return 0;
}
```

問題 2 (必須) コンストラクタ関数のオーバーロード

```
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;

class strtype {
    char *p;
public:
    strtype(char *s);
    strtype(char *s, int n);
    char *getstring() { return p; }
};

strtype::strtype(char *s)
{
    int l;
    l = strlen(s) + 1;

    p = new char [l];
    if(!p) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    strcpy(p, s);
}

strtype::strtype(char *s, int n)
{
    int l;
    l = strlen(s)*n + 1;

    p = new char [l];
    if(!p) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    *p = '\0';
    for (int i = 0; i < n; i++)
        strcat(p, s);
}

int main()
{
    strtype s1("Good!");
    strtype s2("Good!", 5);

    cout << "s1: " << s1.getstring() << "\n";
    cout << "s2: " << s2.getstring() << "\n";

    return 0;
}
```

問題3 コピーコンストラクタ

```
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;

class strtype {
    char *p;
public:
    strtype(char *s);
    strtype(char *s, int n);
    strtype(const strtype &o); // コピーコンストラクタ
    ~strtype() { delete [] p; }
    char *getstring() { return p; }
};

strtype::strtype(char *s)
{
    int l;
    l = strlen(s) + 1;

    p = new char [l];
    if(!p) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    strcpy(p, s);
}

strtype::strtype(char *s, int n)
{
    int l;
    l = strlen(s)*n + 1;

    p = new char [l];
    if(!p) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    *p = '\0';
    for (int i = 0; i < n; i++)
        strcat(p, s);
}

strtype::strtype(const strtype &o)
{
    int l;

    l = strlen(o.p) + 1;

    p = new char [l];
    if(!p) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    strcpy(p, o.p);
}

int ccount(strtype x, char c)
{
    int count = 0;
    char *p;

    p = x.getstring();

    while (*p) {
        if (*p == c) count++;
        p++;
    }

    return count;
}
```

```
int main()
{
    strtype s1("Good!");
    strtype s2("Good!", 5);

    cout << "s1 の o の数: " << ccount(s1, 'o') <<
        "\n";
    cout << "s2 の o の数: " << ccount(s2, 'o') <<
        "\n";
    cout << "s1: " << s1.getstring() << "\n";
    cout << "s2: " << s2.getstring() << "\n";

    return 0;
}
```

問題4 デフォルト引数

```
#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;

class strtype {
    char *p;
public:
    strtype(char *s, int n = 1); // デフォルト引数を利用
    strtype(const strtype &o); // コピーコンストラクタ
    ~strtype() { delete [] p; }
    char *getstring() { return p; }
};

strtype::strtype(char *s, int n)
{
    int l;
    l = strlen(s)*n + 1;

    p = new char [l];
    if(!p) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    *p = '\0';
    for (int i = 0; i < n; i++)
        strcat(p, s);
}

strtype::strtype(const strtype &o)
{
    int l;

    l = strlen(o.p) + 1;

    p = new char [l];
    if(!p) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    strcpy(p, o.p);
}

int ccount(strtype x, char c)
{
    int count = 0;
    char *p;

    p = x.getstring();

    while (*p) {
        if (*p == c) count++;
        p++;
    }

    return count;
}

int main()
{
    char c, s[100];
    int n;

    cout << "文字列 s1 を設定します。¥n 文字列を入力してください: ";
    cin >> s;
    strtype s1(s);
    cout << "繰り返し文字列 s2 を設定します。¥n 文字列を入力してください: ";
    cin >> s;
```

```
    cout << "繰り返し回数を入力してください: ";
    cin >> n;
    strtype s2(s, n);

    cout << "文字列 s1 と s2 に含まれるある文字の数を数えます。¥n 文字を入力してください: ";

    cin >> c;
    cout << "s1 の" << c << "の数: " << ccount(s1, c) << "¥n";
    cout << "s2 の" << c << "の数: " << ccount(s2, c) << "¥n";
    cout << "s1: " << s1.getstring() << "¥n";
    cout << "s2: " << s2.getstring() << "¥n";

    return 0;
}
```

問題 5 予定表

```
#include <iostream>
#include <ctime>
using namespace std;

class Schedule {
    int mon;
    int day;
    char *str;
public:
    Schedule() { mon = 0; day = 0; str = NULL; }
    Schedule(const Schedule &o);
    ~Schedule() { if (str) delete [] str; }
    void set(int m, int md, char *s);
    int get_mon() { return mon; }
    int get_day() { return day; }
    char *get_str() { return str; }
};

Schedule::Schedule(const Schedule &o) {
    int l;
    l = strlen(o.str) + 1;
    str = new char [l];
    if(!str) {
        cout << "メモリ割り当てエラー\n";
        exit(1);
    }
    strcpy(str, o.str);
    mon = o.mon;
    day = o.day;
}

void Schedule::set(int m, int d, char *s) {
    int flag = 0;

    if (m < 1 || m > 12 || d < 1) {
        flag = 1;
    } else {
        switch(m) {
            case 1: case 3: case 5: case 7: case 8: case
10: case 12:
                if (d > 31) {
                    flag = 1;
                }
                break;
            case 4: case 6: case 9: case 11:
                if (d > 30) {
                    flag = 1;
                }
                break;
            case 2:
                if (d > 28) {
                    flag = 1;
                }
                break;
        }
    }
    if (flag) {
        cout << "日付が不正です\n";
    } else {
        int l;

        if (str) {
            delete [] str;
        }
        l = strlen(s) + 1;
        str = new char [l];
        if(!str) {
            cout << "メモリ割り当てエラー\n";
            exit(1);
        }
        strcpy(str, s);
        mon = m;
    }
}
```

```
        day = d;
    }
}

bool today(Schedule ob) {
    time_t now = time(NULL);
    struct tm *date = localtime(&now);

    if (ob.get_mon()==date->tm_mon+1 &&
ob.get_day()==date->tm_mday)
        return true;
    else
        return false;
}

int main() {
    Schedule obj[5];

    obj[0].set(4, 4, "入学式");
    obj[1].set(5, 7, "創立記念日");
    obj[2].set(6, 20, "世界難民の日");
    obj[3].set(7, 17, "海の日 (補講日)");
    obj[4].set(8, 11, "山の日");

    int flag = 0;
    cout << "今日の予定\n";
    for (int i = 0; i < 5; i++) {
        if (today(obj[i])) {
            cout << "¥t" << obj[i].get_str() << "¥n";
            flag = 1;
        }
    }
    if (!flag) {
        cout << "¥t何也没有什么¥n";
    }

    cout << "全予定表示¥n";
    for (int i = 0; i < 5; i++) {
        cout << obj[i].get_mon() << "月" <<
obj[i].get_day() << "日¥t";
        cout << obj[i].get_str() << "¥n";
    }

    return 0;
}
```