[prog2] Programming C++ (C6) Exercise Guide (Ex12)

11/16, Thursday 3rd period.

Ex12 について (基本的にEx11と同じ)

- 教科書を読みながら写してください
- 答え自体は全部書いてあります
- ▶ 「任意の長さの文字列を格納」という意味では、前回のVecクラスを そのまま流用できます
- Vecクラスは、最初にそのまま貼りつけてOK (ちゃんとできてるならば...)

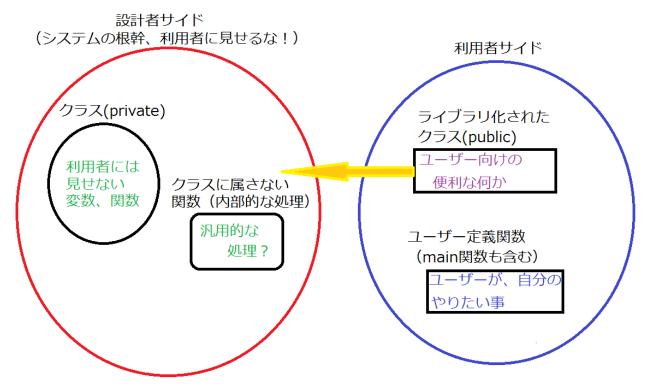
string型の基本設計

- ▶ Cでは文字(リテラル)の配列だった
- C++のstringもだいたい同じ仕様
 - string型の変数は添え字指定でI文字取り出せる
 - 文字列長を意識しないで、自由に文字列を格納可能
 - イテレータによるアドレス参照が可能

文字を要素とするvectorコンテナ

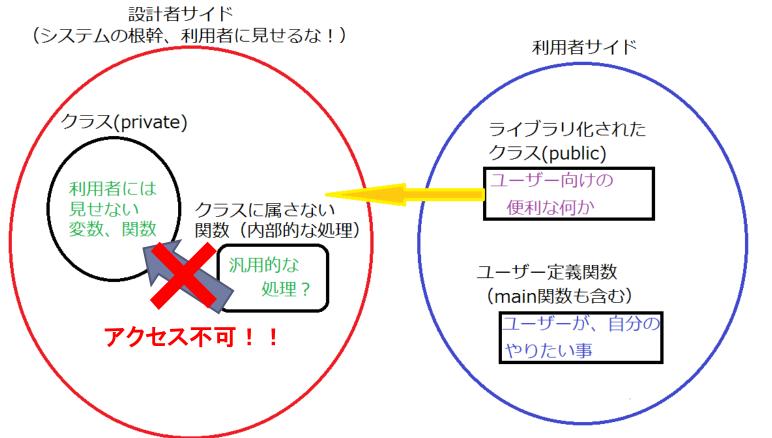
ExllのVecクラスとの組み合わせで、完全自前の 文字列を扱うクラス Strクラス を作ろう

クラス設計者側の都合で、クラスのメンバーには含めない(内部処理的な)ただの関数があるとする

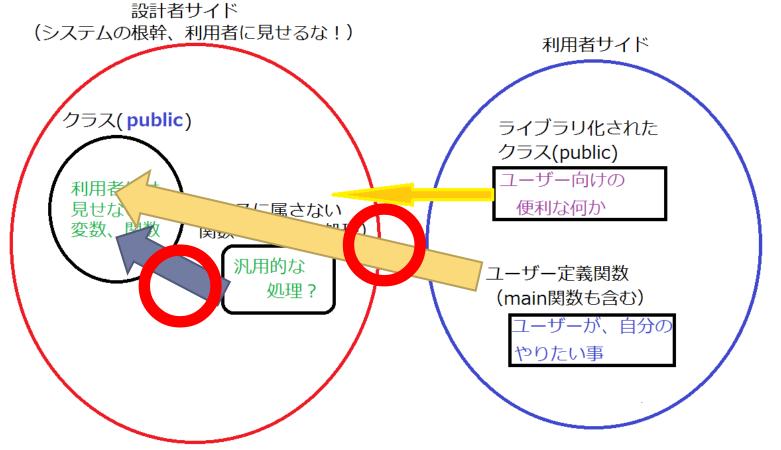


ただの関数 ⇒ クラスのメンバーにするほどではない、あるいは特定のメンバーにはしない方がいいような汎用的な関数など

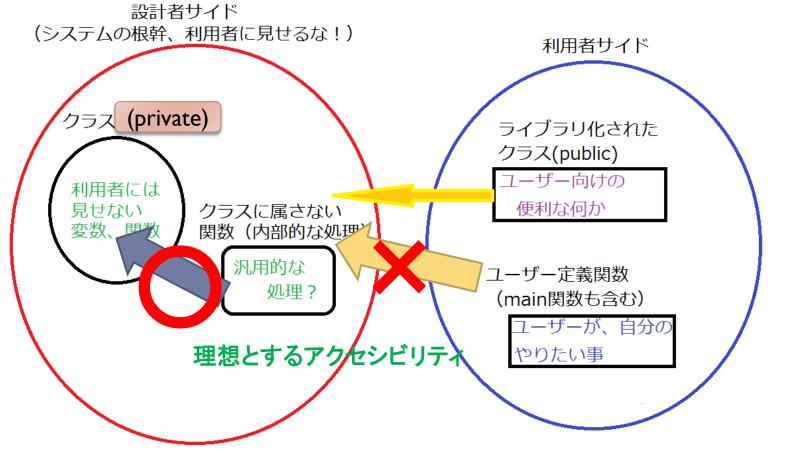
C++のオブジェクト指向プログラミングスタイルに従えば、 クラスの内部的な処理は private 宣言で隠蔽されている



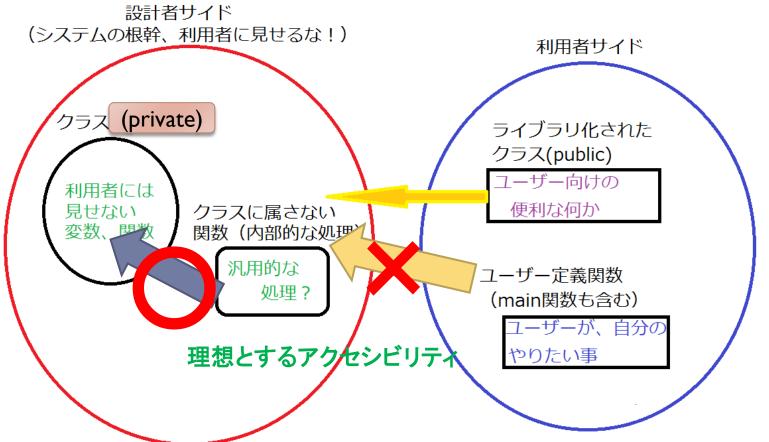
▶ では public 宣言するか? ⇒ 見られる、触られるとヤバい部分をユーザーに好き放題にされてしまう!!!



▶ あくまで、ユーザー(ユーザー定義の処理)には内部処理を触られたくない(結果的にバグの温床にもなるし...)



▶「内部的な処理」とやらもクラスに属するメンバー関数っ ぽい働きにできれば ⇒ フレンド関数



フレンド関数 (例題1)

▶「内部的な処理」とやらをクラスに友達登録させる クラス定義内で friend【クラス外の関数のプロトタイプ】

```
#include <iostream>
using namespace std;
class Test{
//フレンド関数
//外部の通常の関数「setval()」からは、あたかもsetval()がクラスのメンバー関数であるかのように
//private宣言されたこのクラスのメンバーにアクセスができる
//この1行をコメントアウトすれば、「valltprivateだからsetval()からはアクセスできない」とコンバイルエラーが出る
friend int setval(Test*,int); // int setval(Test*,int); をこのクラスの友達として登録する
private:
 int val;
 void checkfunc(void){//この関数もprivateなので、単純に外からは見えない
   cout << "We are friends!!! YEAR!!!!!!" << endl;</pre>
public:
 Test(): val(1){//1でvalが初期化されている
  cout << "Test class object" << endl;</pre>
 int getval(void){//ゲッターはpublicで宣言されているので、どこからもアクセス可能
  return val:
                                   元々の関数のプロトタイプ宣言を消しては
};
                                   ダメ(あくまでフレンド登録はオプショナル)
//setval()はTestクラスとは無関いのただの関数
```

int setval(Test*,int); //通常の関数なので、(Testクラス内のfriend宣言とは別に)ブロトタイプ宣言は必要

フレンド関数 (例題1)

友達になっていれば、クラス外のただの関数でも クラス内のprivate宣言されたメンバーにアクセス可能

```
int main(){
                                        ここからは、privateなメンバー
 Test a:
                                        にはアクセス不可能
 setval(&a,3);
 //a.val = 3; //これはもちろんダメ(valはprivate)
 //a.checkfunc(): //これもダメ(checkfunc()はprivate)
 cout << a.getval() << endl; // getval()|lpublic
 return 0;
//アドレスが指すクラスオブジェクトのvallCsetの値を代入する
//普通はこんなことはできないが、setval()がTestクラスにお友達認定されていれば
//クラスのメンバー関数であるかのように扱える
int setval(Test* a,int set){
                          ただの関数だが、Testクラスに友達登
 a->val = set;
 a->checkfunc();
                          録されていれば、privateなメンバーを
 return 1:
                          参照できる
```