Subject: Re: Make THISFN simpler and more powerful
Posted by Lance on Thu, 10 Nov 2022 20:41:07 GMT
View Forum Message <> Reply to Message

Some failed attempts.

```
template <class T>
concept UppMoveable = requires (T t){
   Upp::Vector<T> v;
   v.Add(t);
};
```

This will fail in compilation with a strange error message.

Turns out you cannot do variable declaration in a requires-expression.

Something like this

```
template <class T>
concept UppMoveable = requires (T t, Upp::Vector<T> v){
   v.Add(t);
};
```

compiles fine but doesn't give desired results. Compiler only check if expressions inside a requires-expression are well-formed. Evaluation doesn't take place.

So this won't work. Back to old conclusion: we need a way to check is a class T is declared/defined with something like

```
class MyString : public Moveable<MyString, optionalB>{...};
```

to determine if it can be placed in a Upp::Vector.

Here are some well written articles about concepts and requires-expression.

C++20 Concepts - a Quick Introduction

Requires-expression