```cpp
void push_42 (cow_vector<int>::inout v)
{
    std::thread t ([v]
                   {
                       // create a copy
                       auto vec2 = v;
                   });

    v.push_back (42); // no data-race as v has
                      // an internal copy
}
```

Modelled **`inout`**

86

```
class inout
{
public:
    ~inout()
    {
        *cow_vector_arg = cow_vector; // copy back modifications
    }

    // reflect to generate and forward all
    // functions to internal copy

private:
    friend class cow_vector;
    cow_vector* cow_vector_arg; // pointer to original
    cow_vector cow_vector;      // copy, safe to modify

    inout (cow_vector* v)
        : cow_vector_arg (v),
          cow_vector (*v)
    {}
};

inout make_inout()
{
    return inout (this);
}
```

```
cow_vector<int> vec;
vec.push_back (40);
vec.push_back (41);

push_42 (vec.make_inout());

//... vec contains 42
```

```
vec = vec_func;
```

```
void push_42 (cow_vector<int>::inout v)
```

{

```
// create a copy
```

{

```
v.push_back (42); // no data-race as v has
```

}

```
});
```

```
auto vec2 = v;
```

```
std::thread t ([v]
// an internal copy
```

Modelled inout

{

```
class inout
```

```
~inout()
```

```
public:
```

{

```
cow_vector* cow_vector_arg; // pointer to original
```

```
friend class cow_vector;
```

```
*cow_vector_arg = cow_vector; // copy back modifications
```

```
return inout (this);
```

```
inout make_inout()
```

```
// functions to internal copy
// reflect to generate and forward all
```

```
inout (cow_vector* v)
```

```
};
```

```
private:
```

```
: cow_vector_arg (v),
cow_vector (*v)
```

{ }

}

```
cow_vector cow_vector;      // copy, safe to modify
```

}

```
vec.push_back (40);
```

```
cow_vector<int> vec;
```

```
push_42 (vec.make_inout());
```

```
vec.push_back (41);
```

```
//... vec contains 42
```

```
vec = vec_func;
```

```
inout (cow_vector* p)
```

```
: cow_vector_arg (p),
cow_vector (*p)
```

```
return inout (this);
```

```
auto vec2 = *v;
```

```
push_42 (&vec);
```

```
inout operator&()
```

```
return inout (this);
```

{

}