

75 STRUCTURED BINDINGS in C++

1. 结构化绑定（只针对C++17）

Structured binding（结构化绑定）是一个新特性，让我们更好地处理多返回值（多返回值可参考[52 C++处理多返回值](#)），这是在52课方法基础上拓展的一种处理多返回值的新方法，特别是如何处理 *tuple*（元组）和 *pairs*（对组）以及返回诸如此类的东西。因为结构化绑定简化了我们的代码，让它比以前的做法更简洁。

在52课中提到Cherno喜欢用返回结构体的实例来处理多返回值，随着结构化绑定的引入，这种情况可能会发生改变。

以前的使用方法

```
C++

#include <tuple>

std::tuple<std::string,int> CreatePerson() // 由于只有两个参数, 也可以用std::pair, 但元组可继续拓展
{
    return { "Cherno", 24 };
}

int main()
{
    auto person = CreatePerson();
    std::string& Name = std::get<0>(person);
    int age = std::get<1>(person);
}
```

或者用 `std::tie` 方法，这里不用创建实例，可以更漂亮一些，但还是结构体用起来更好：

```
C++


std::string Name;
int age;
std::tie(Name, age) = CreatePerson();
```

而结构化绑定解决了这些问题，让代码看起来很好（VS默认C++14，所以要在项目属性里设C++语言标准为17）：

```
C++

auto [name, age] = CreatePerson();
std::cout << name << ":" << age << "\n";
```

```
#endif #if 0
    auto [name:string, age:int] = CreatePerson();
    std::cout << name << ":" << age << "\n";
}
```

 Microsoft Visual Studio Debug Console

Cherno:24

C:\Dev\HelloWorld\bin\Win32\Debug\HelloWorld.exe (p
To automatically close the console when debugging s
le when debugging stops.
Press any key to close this window . . .