

# デバッグ手法

デバッグとは、ソフトウェアのソースコードのエラーやバグを見つけて修正するプロセス

最も基本的なデバッグ手法

- ソースコードをよく確認する
- デバッグ用のコードを仕込む
  - 確認用に一時的な動作をさせる
  - 変数をコンソールに表示する
- ブレークポイントを設定する
  - 一時停止して変数の内容を確認する
  - ステップ実行して動作を確認する

# 変数をコンソールに出力

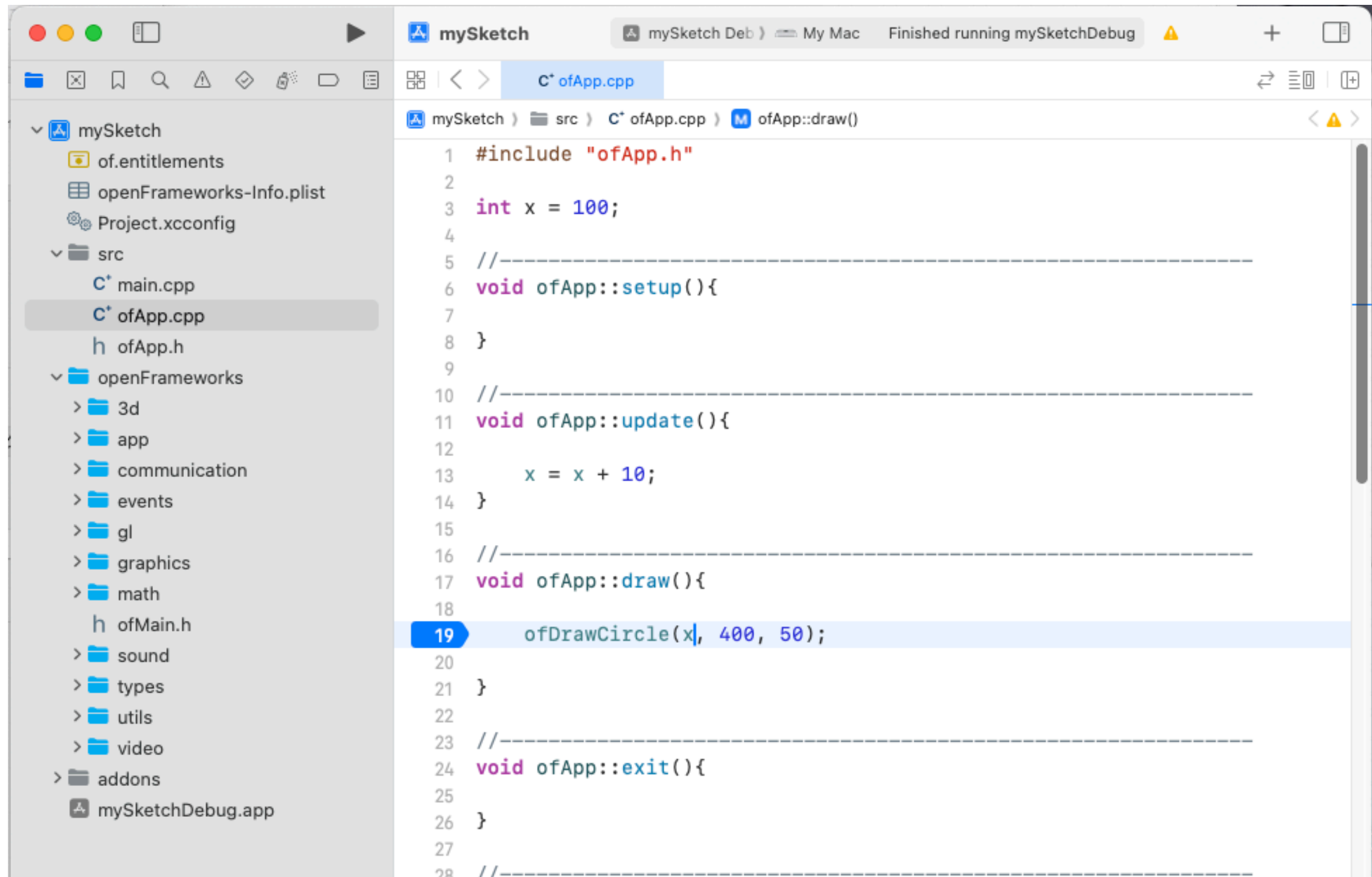
- ・ 変数の内容を見たいところに cout を仕込む

```
for (int x = 0 ; x < 10 ; x++) {  
    ofDrawCircle(x, 200, 100);  
    cout << x << endl;  
}
```

printf()も使えます。調べてみてください。

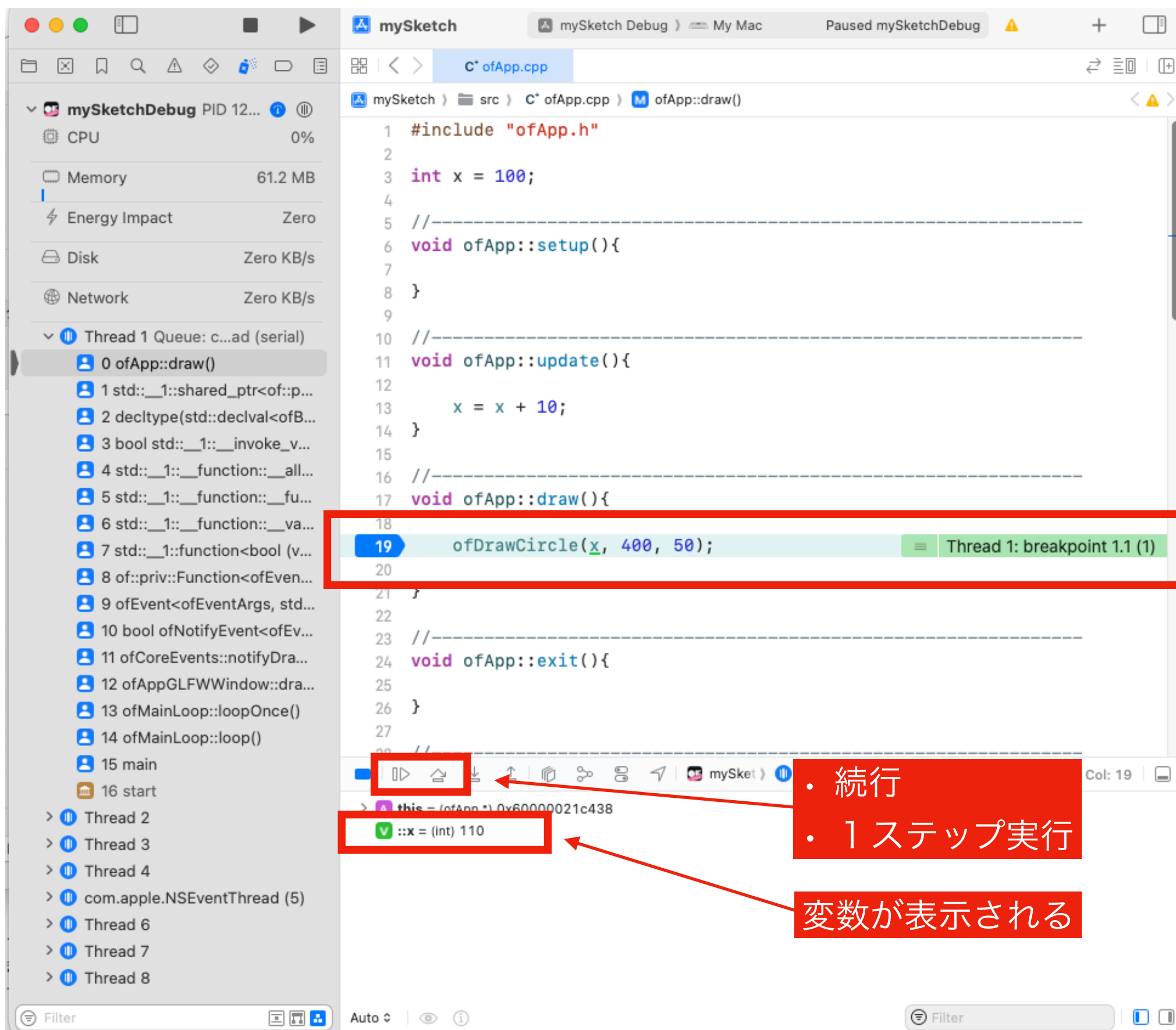
# ブレークポイントを設定

Xcode



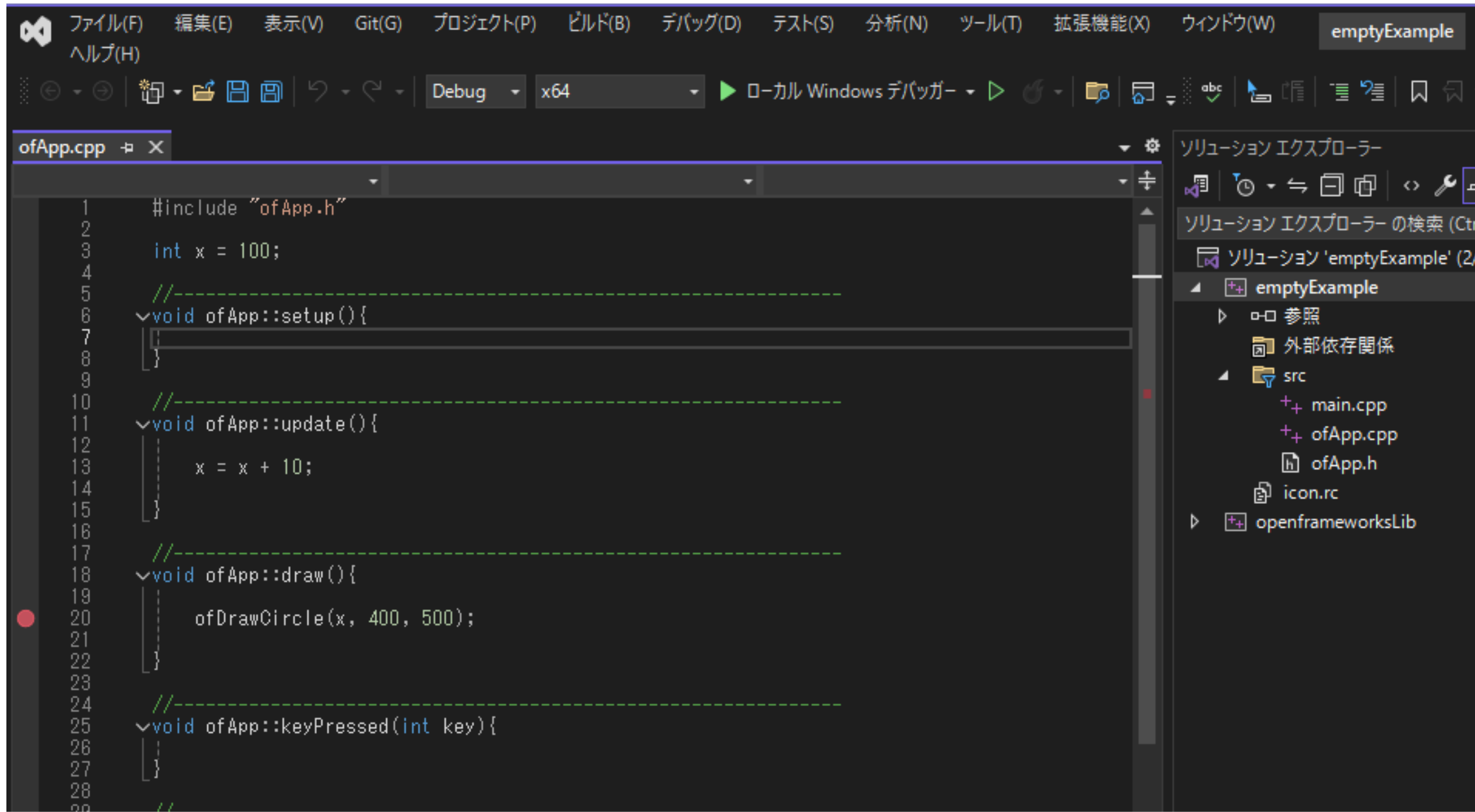
Xcode

# ブレークポイントで停止



# ブレークポイントを設定

Visual Studio



# ブレークポイントで停止

## Visual Studio

The screenshot shows the Visual Studio IDE in debug mode. The main window displays the source code of `ofApp.cpp`. A breakpoint is set at line 20, which is `ofDrawCircle(x, 400, 500);`. The call stack on the right shows the current function is `draw()`. The local variables window at the bottom shows the variable `x` with a value of 110.

Red boxes and arrows highlight the following elements:

- The **Continue (C)** button in the debug toolbar.
- The **Step Into** button in the debug toolbar.
- The breakpoint icon on line 20 of the source code.
- The variable `x` in the local variables window.

Annotations in red boxes:

- 続行
- 1 ステップ実行
- 変数が表示される