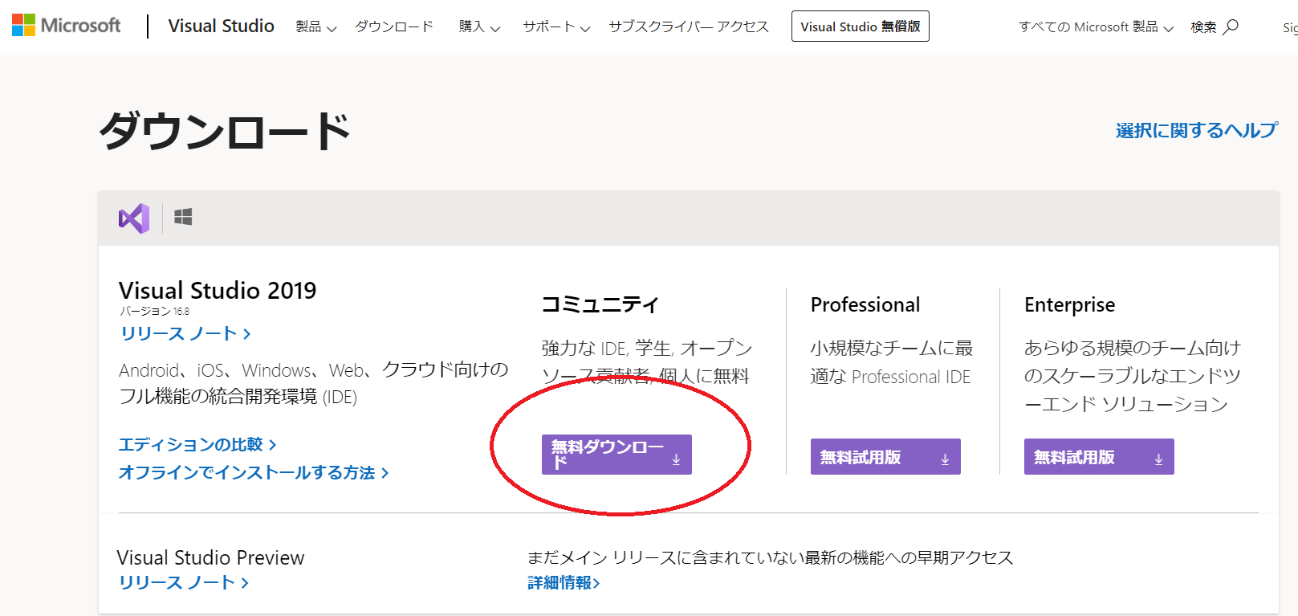
研修環境セットアップ

|  |  |  |
| --- | --- | --- |
| 日付 | 変更内容 | 変更者 |
| 2021/02/04 | 新規作成 | 庭 |
| 2021/04/02 | git登録時の設定を追記 | 庭 |

# Visual Studio 2019のセットアップ

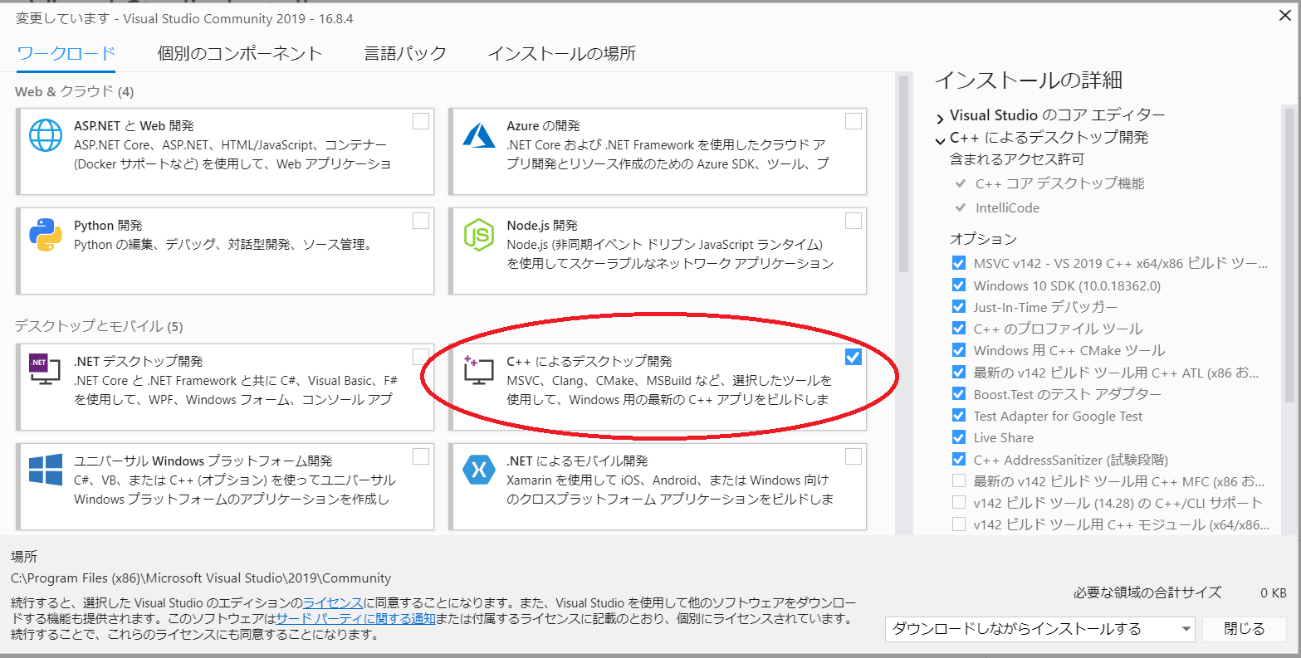
<https://visualstudio.microsoft.com/ja/downloads/>

から無料版をダウンロードし、インストールする。

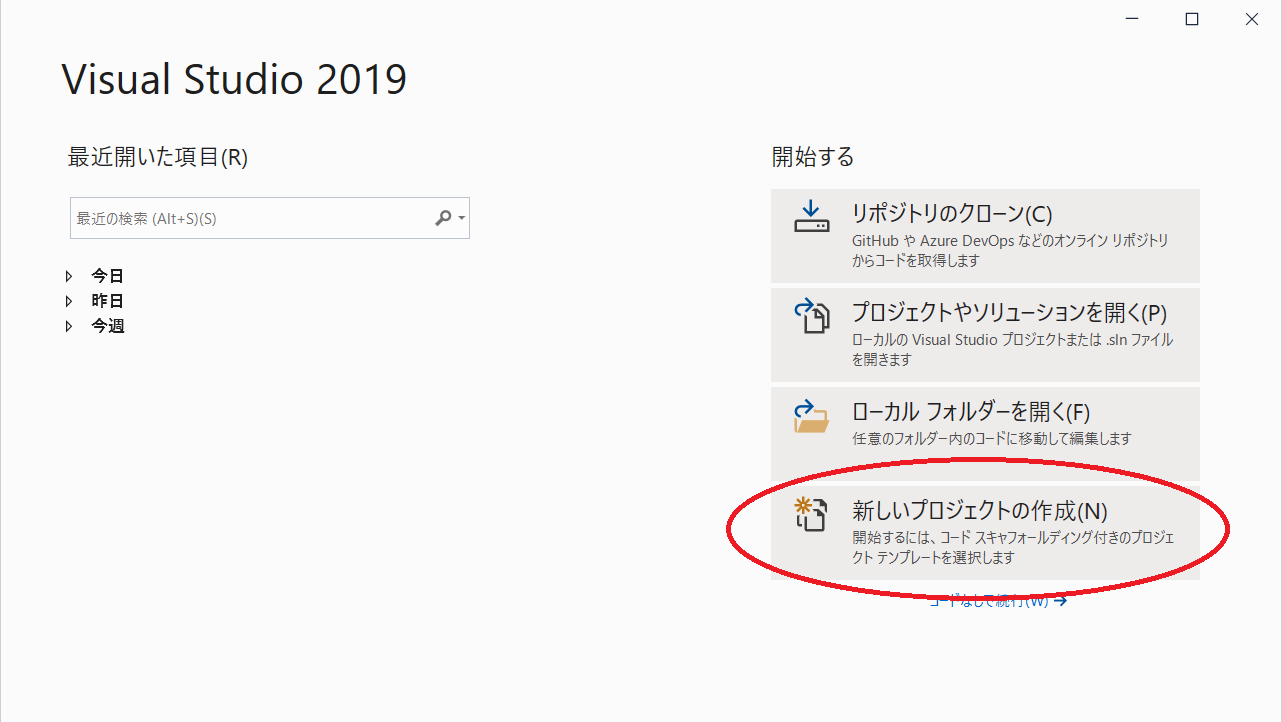


インストールするワークロードは、「C++によるデスクトップ開発」を選択

(C++のコンパイラでCのコンパイルも可能)



インストール後、Visual studio 2019を起動し、「新しいプロジェクトの作成」を選択



「コンソールアプリ」を選択し、「次へ」

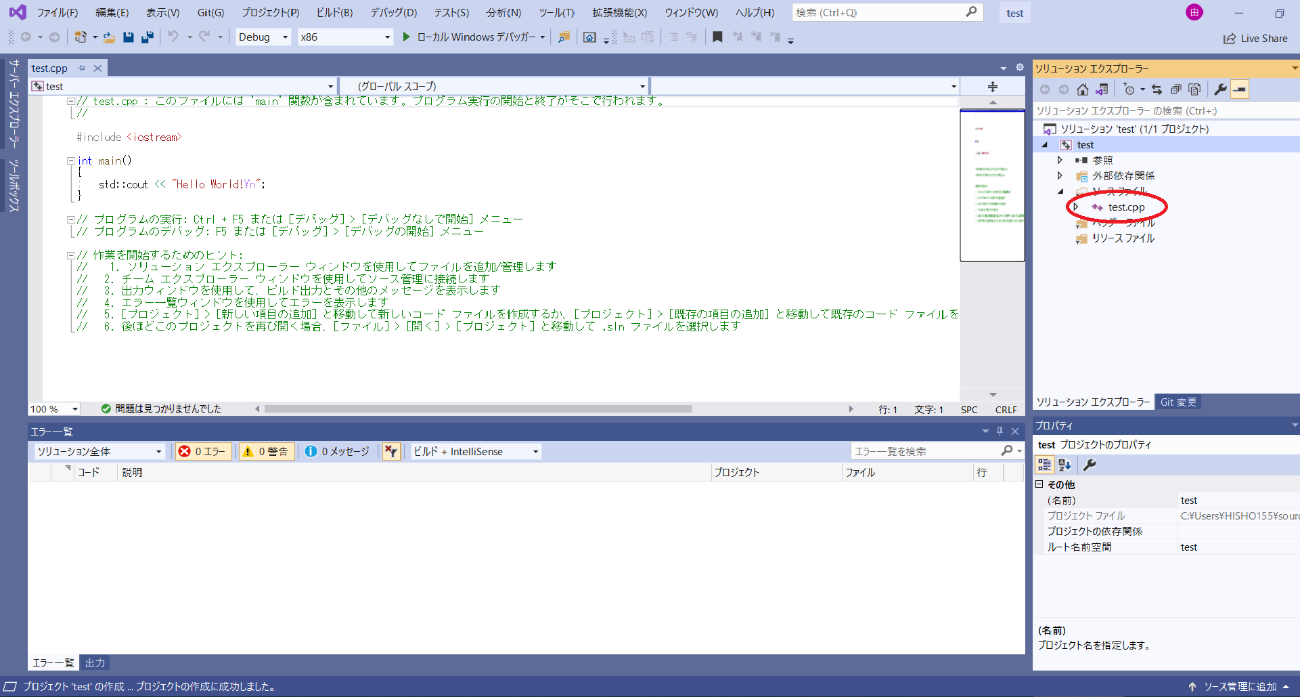


「プロジェクト名」に任意の名前を入力し、「場所」には保存先を入力し、作成

C++で作成されたHello Worldのサンプルプログラムが作成されるので、サンプルコードを削除する。

プロジェクトが保存されているフォルダへソースコードを格納し、

「ソリューションエクスプローラー」の「ソースコード」を右クリック→追加→既存の項目を選択



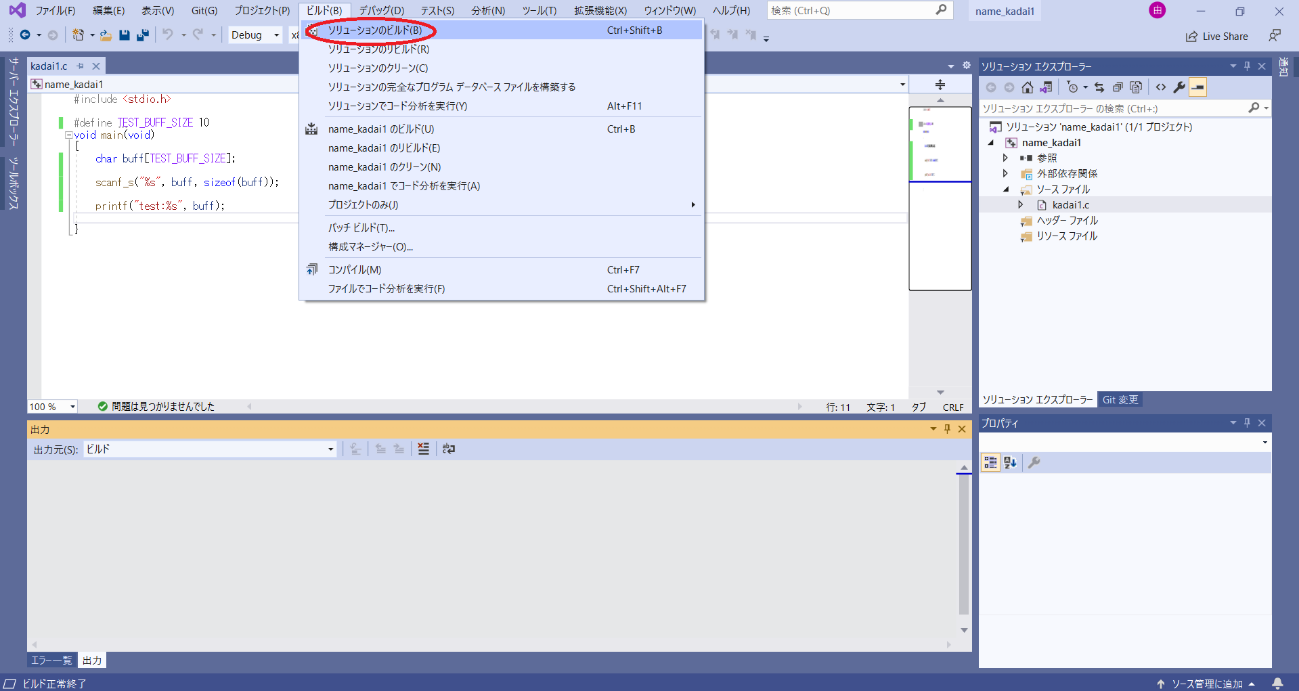
# ビルド

ビルド→ソリューションのビルド

でビルドします。

※リビルドはビルドによって生成されたオブジェクトを削除し、ビルドしなおします。

※クリーンはビルドによって生成されたオブジェクトを削除します。



コンパイルエラーは「出力」や「エラー一覧」に表示されます。

# デバッグ

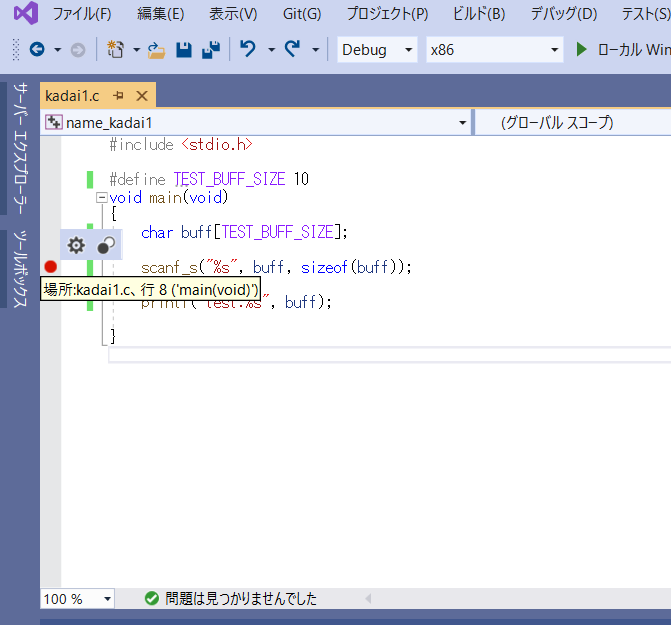
## ブレークポイントの設定

ブレークポイントを設定すると、プログラムを実行した際に任意の場所で止めることができます。

コードの左側をクリックすると●がつきます。

または右クリック→ブレークポイントブレークポイントの挿入

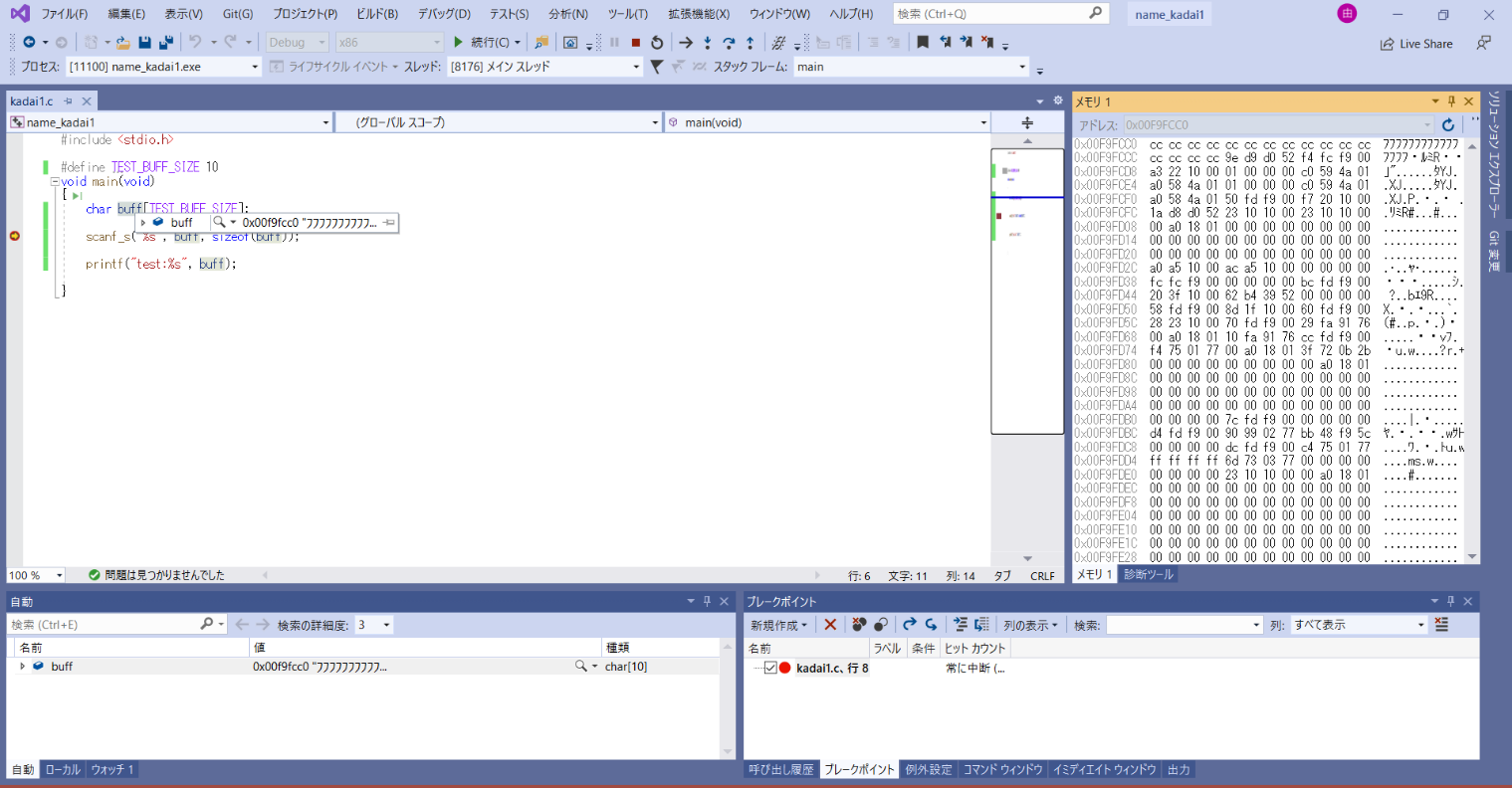
でもできます。



## デバッグの開始

デバッグ→デバッグの開始

またはF5キーでデバッグを開始します。



・ブレークポイントで止まったら、F10キーを押すことで次の行へ進み、プログラムを1行ずつ実行できます。

・関数を実行する行でF11キーを押すことでその関数の中へ進みます。

・右クリック→次のステートメントの設定とすると次に実行する行を設定できます。処理を飛ばしたい場合に便利です。

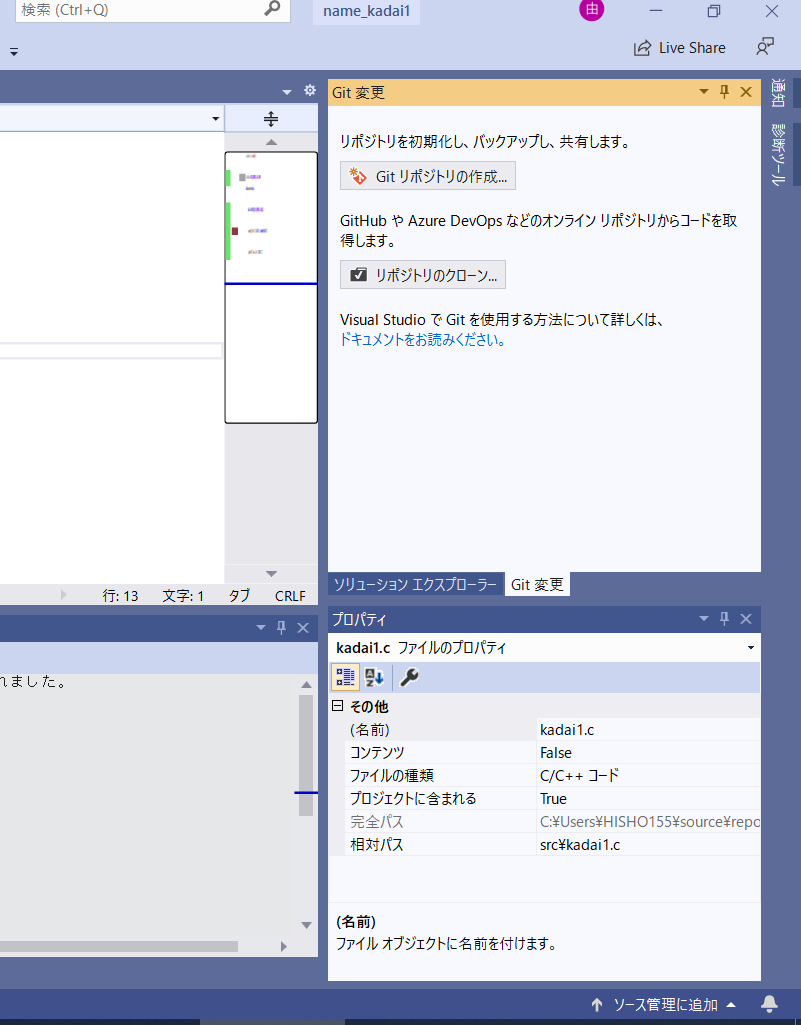
・「自動」や「ローカル」、「ウォッチ」ウィンドウで変数の値を確認したり、変更したりできます。

・デバッグ→ウィンドウ→メモリ→メモリ1を選択するとメモリウィンドウが表示され、メモリの値を確認できます。確認したい変数をドラッグ＆ドロップすると確認できます

# githubへのアップロード

## 初めての登録の場合

1. Git変更タブを選択し、Gitリポジトリの作成を選択



1. アカウント欄に研修用に作成したアカウントを設定する

アカウント名：hishosoftinfo@gmail.com

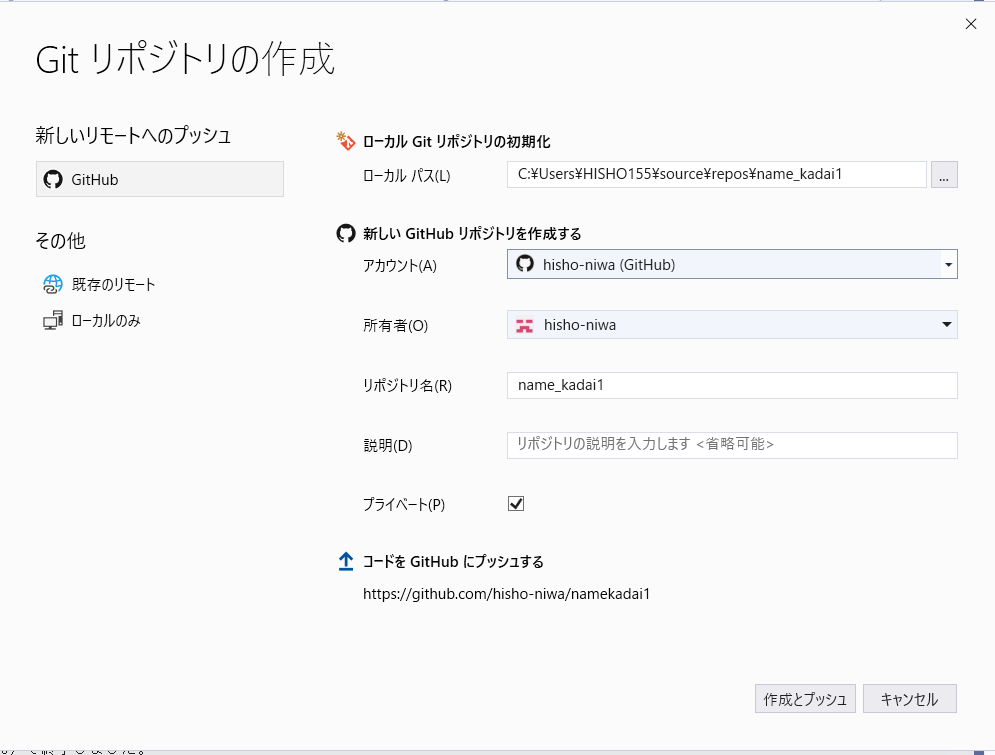
パスワード：hisho5761git

所有者は自動で決まります。

1. 同じアカウントで多数の新人の課題を管理するので、リポジトリ名はわかりやすいものにしてください。

例)

NiwaYutaka\_kadai01



1. 承認されたユーザーのみが閲覧できるようにするため、プライベートにします。
2. 作成とプッシュを選択します。

※gitのアカウント名とメールアドレスが設定されていない場合、

プッシュができませんので、入力を促された場合は

ユーザー名：任意の名前

メールアドレス：niwa@hi-soft.co.jp

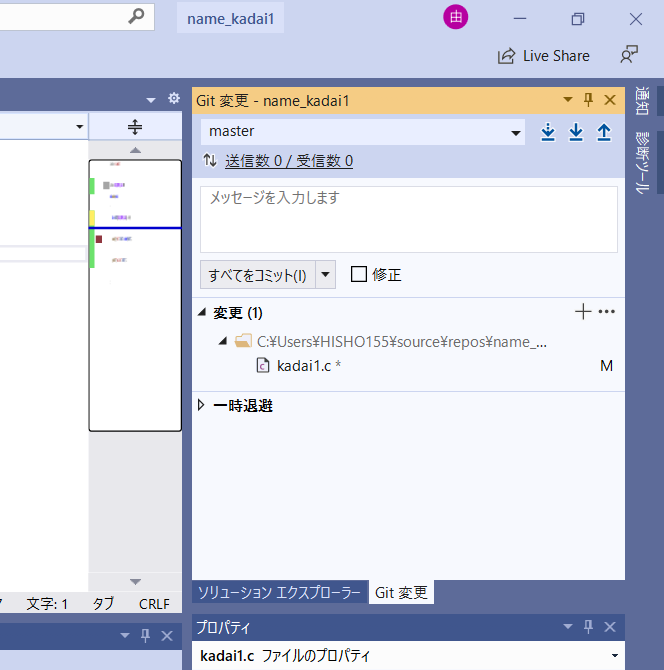
としてください

## 2回目以降の登録(更新)の場合

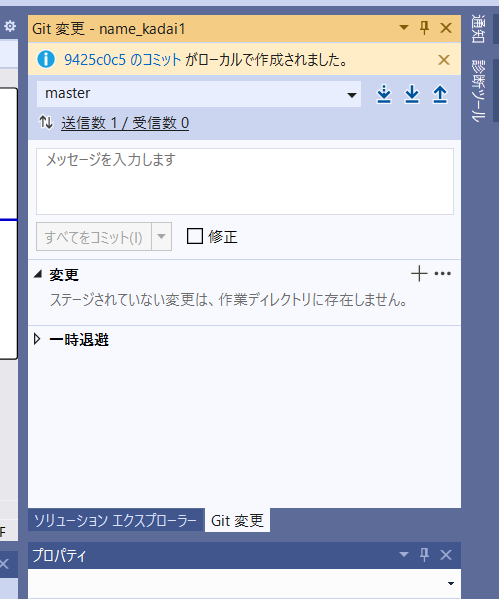
1. 変更の欄に変更したファイルが表示されます。
2. “メッセージを入力します“と書かれた欄に変更内容を記載してください。

メッセージは以下のフォーマットで記載してください。

|  |
| --- |
| コミット内容の要約（タイトル・概要）  (空行)  本文 （内容・詳細・変更理由）  (空行)  修正者 |

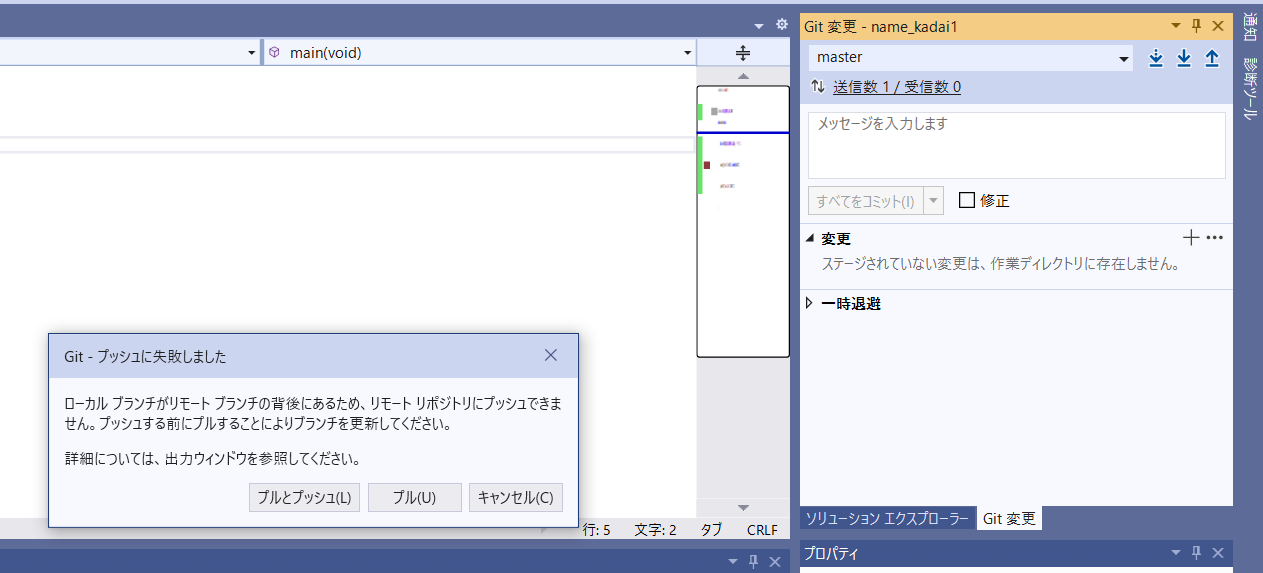


1. すべてをコミットを選択します
2. ↑のアイコンを選択し、コミットした修正をgithubへプッシュします。



## コンフリクトの解消

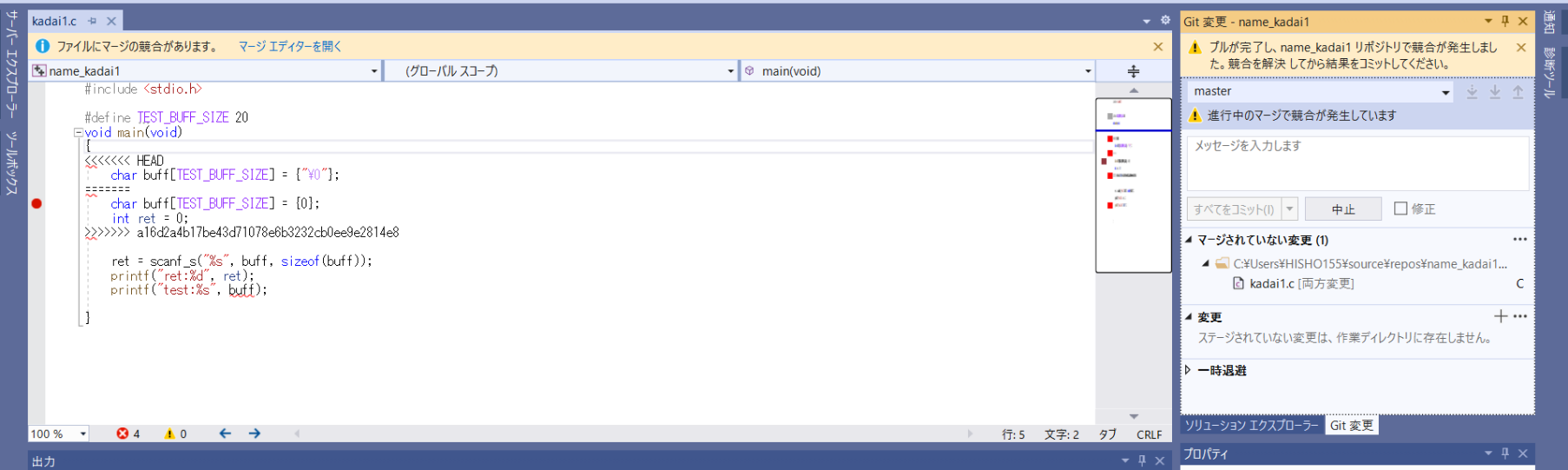
複数人で同じファイルを更新していると、修正内容が衝突してpushに失敗することがあります。これをコンフリクトといいます。



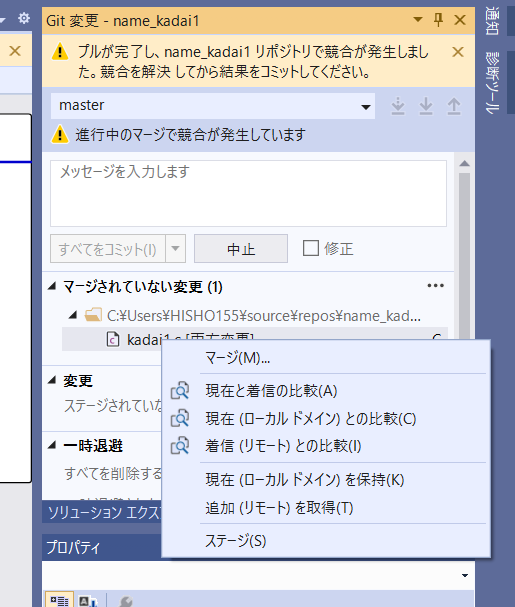
1. まずはリモートブランチ（github）から最新をプルしましょう。プルとプッシュorプルを選択します。
2. コンフリクトが発生した場所にコンフリクトマーカー※が発生します。適切に修正してください。

※コンフリクトマーカー

|  |
| --- |
| <<<<<<< HEAD  自分の修正  =======  リモート側の修正  >>>>>>> XXXXXX |



1. 修正できたら、マージされていない変更に入っているコードを右クリック→ステージを選択



1. 通常と同じようにコミットメッセージを入力してpush

# githubで確認

<https://github.com/>

にアクセスし、Sign inをクリックしてサインインする

Repositoriesから登録したリポジトリを選択すると、コードや履歴を見ることができます。

(編集もできます)

ネット環境があればいつでもコードが確認できます。家のPCでも作業できます。

