

Лабораторная работа №1

Программный инструментарий

1 Устанавливаем git

1.1 Для этого переходим по ссылке: <https://git-scm.com>

1.2 После загрузки страницы, нажимаем «Download for Windows» (рисунок 1)

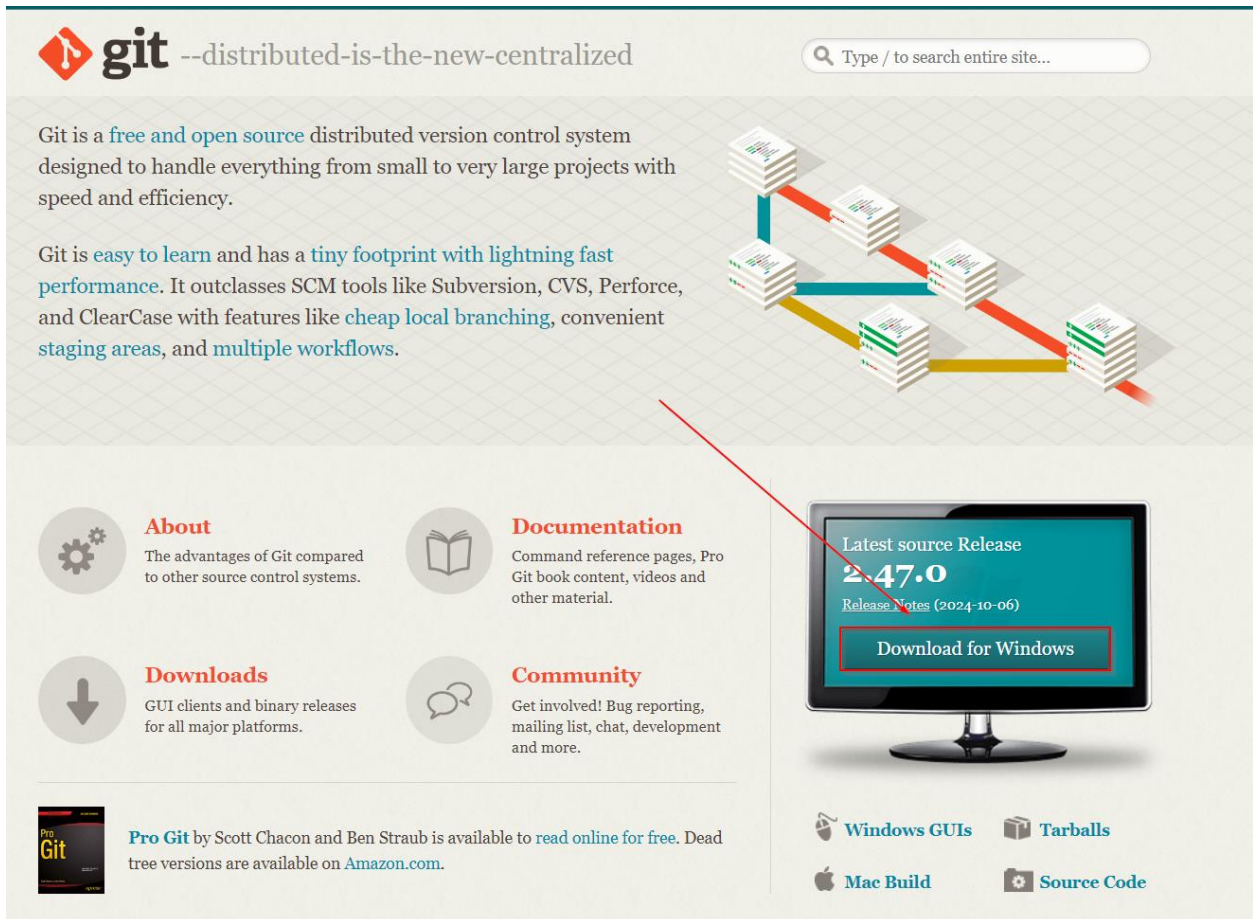


Рисунок 1 – Кнопка перехода на страницу загрузки

1.3 Скачиваем установщик последней версии для Windows X64, для этого нажимаем на кнопку «Click here to Download» (рисунок 2)

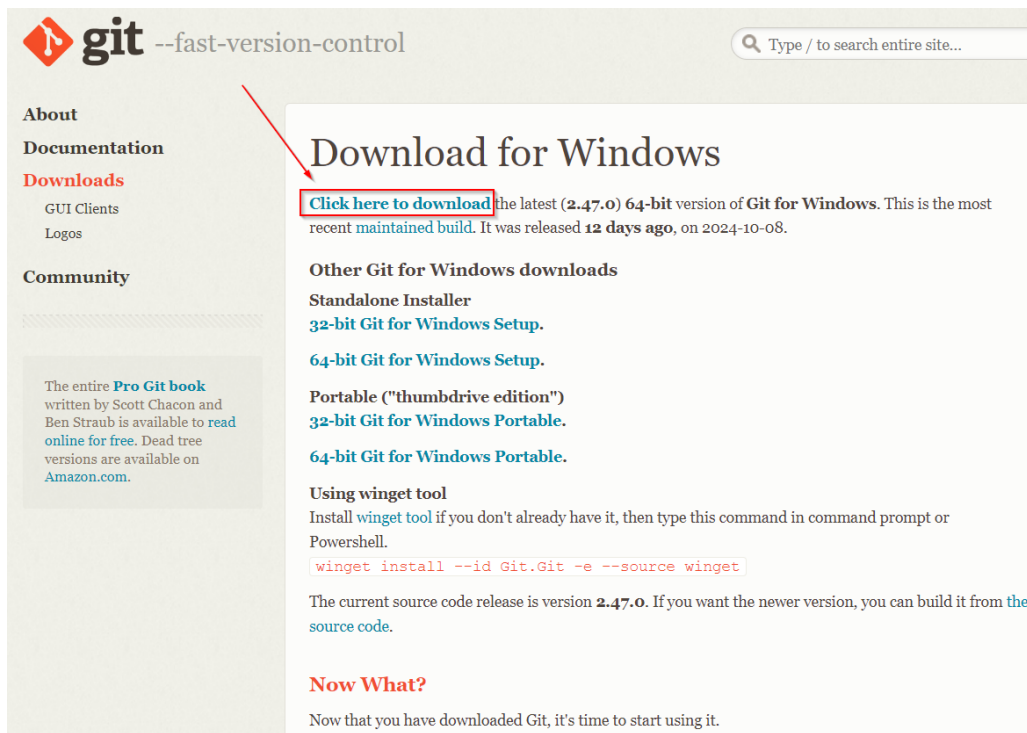


Рисунок 2 – Кнопка загрузки приложения

1.4 После загрузки файла, открываем его и в появившемся окне «Контроль учетных записей» разрешаем приложению вносить изменения на устройстве, нажав «Да»

1.5 Далее нажимаем «Next» -> ... «Next» и ждем завершения установки.

1.6 По окончании установки нажимаем «Finish» (рисунок 3)

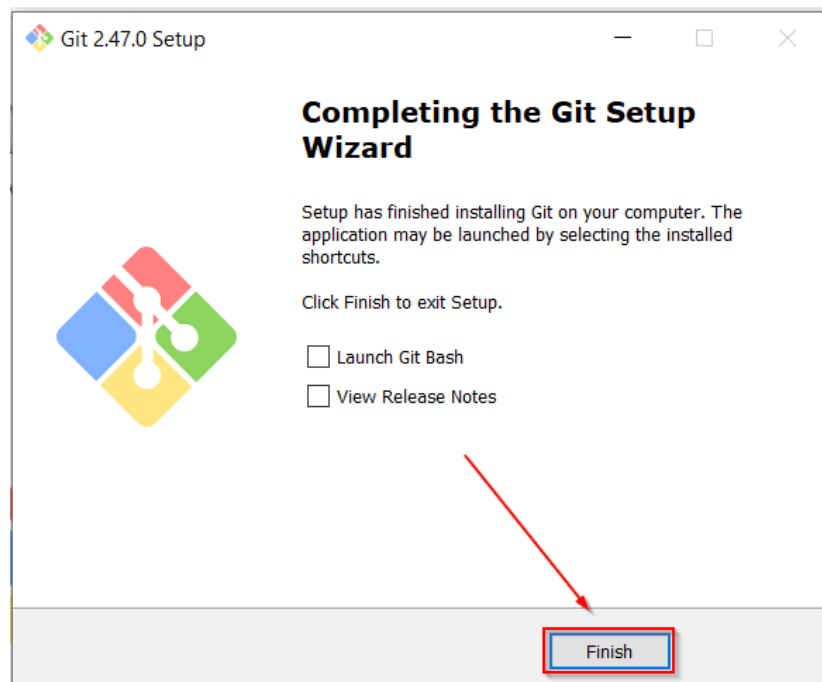


Рисунок 3 – Завершение установки

2 Входим в аккаунт github

2.1 Для этого переходим по ссылке: <https://github.com/logout> и нажимаем «Sign in» (рисунок 4)

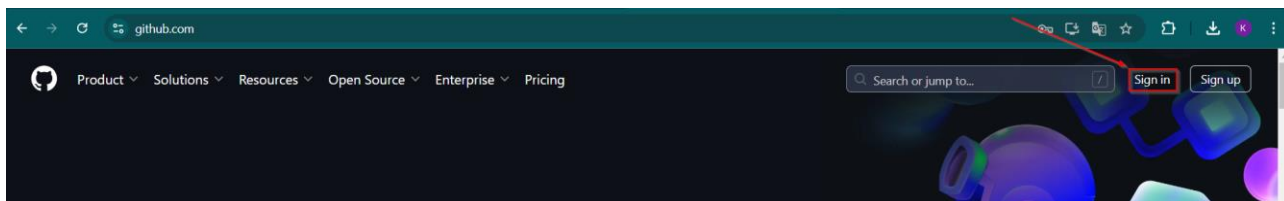


Рисунок 4 – Кнопка входа в аккаунт github

2.2 Далее вводим логин и пароль и нажимаем «Sign in» (рисунок 5)

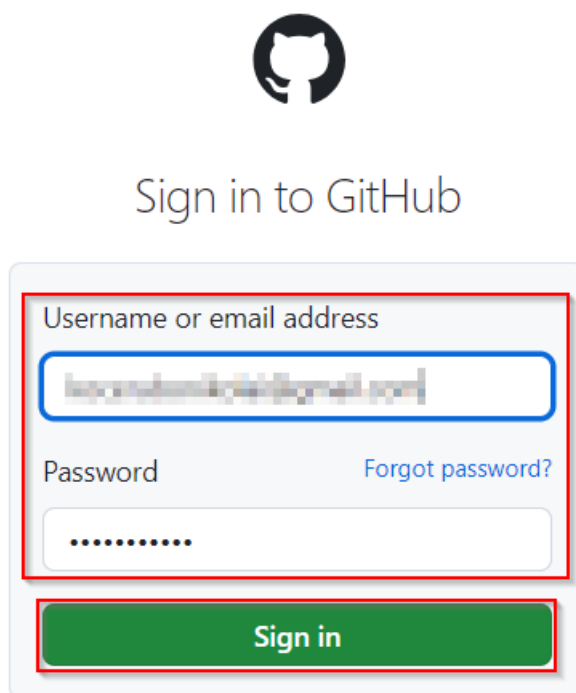


Рисунок 5 – Вход в аккаунт

3 Загрузка файла в репозиторий на github

3.1 В папке с проектом открываем Windows PowerShell (Shift + ПКМ)

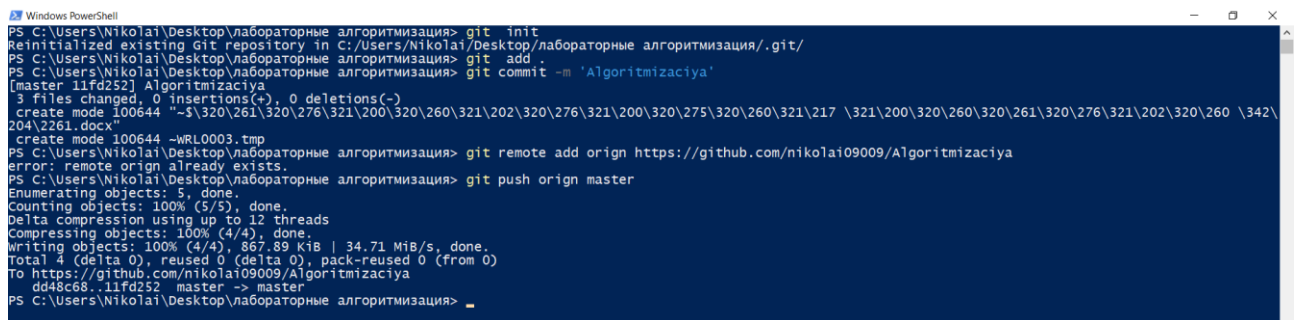
3.2 Инициализируем git: `git init`

3.3 Добавляем в локальный репозиторий файл с помощью команды: `git add`.

3.4 Создаем первый локальный коммит в репозитории: `git commit -m "Algoritmizaciya"`

3.5 Добавляем локальному репозиторию ссылку на удаленный репозиторий `github: github git remote add origin https://github.com/nikolai09009/Algoritmizaciya` (так как удаленный репозиторий уже существует высветилась ошибка: `error: remote origin already exists`)

3.6 Далее загружаем файлы с помощью команды: `git push origin master` (рисунок 6)



```
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> git init
Reinitialized existing git repository in C:\Users\Nikolai\Desktop\лабораторные алгоритмизация/.git/
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> git add .
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> git commit -m 'Algoritmizaciya'
[master 11fd252] Algoritmizaciya
3 files changed, 0 insertions(+), 0 deletions(-)
create mode 100644 "-$320\261\320\276\321\200\320\260\321\202\320\276\321\200\320\275\320\260\321\217 \321\200\320\260\320\261\320\276\321\202\320\260 \342\204\2261.docx"
create mode 100644 -WRL0003.tmp
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> git remote add origin https://github.com/nikolai09009/Algoritmizaciya
error: remote origin already exists.
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 12 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 867.89 KiB | 34.71 MiB/s, done.
Total 4 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/nikolai09009/Algoritmizaciya
dd48668..11fd252 master -> master
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> _
```

Рисунок 6 – Загрузка файлы в репозиторий github

3.7 После этого появилось окно авторизации github, в котором необходимо ввести данные от аккаунта github

3.8 После авторизации файлы загрузились в ветку github

4 Загрузка компилятора

4.1 Скачиваем zip архив по прямой ссылке:

4.2 Распаковываем его в C:\Users\Nikolai

4.3 Открываем параметры переменных сред и добавляем путь в файлу bin (рисунок 7)

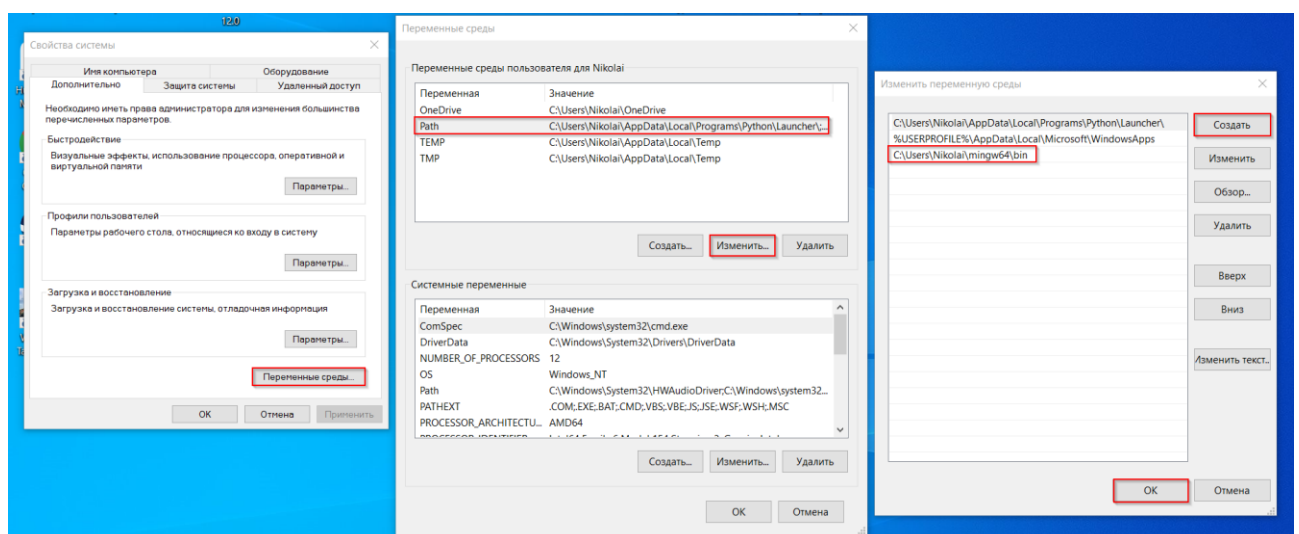


Рисунок 7 – Изменение параметров переменных сред

5 Установка Visual Studio Code и расширения C/C++

5.1 Переходим по ссылке: <https://code.visualstudio.com/Download#>

5.2 Скачиваем установщик

5.3 После загрузки открываем его и принимаем лицензионное соглашение и устанавливаем программу

5.4 По завершении установки, нажимаем «Завершить» (рисунок 8)

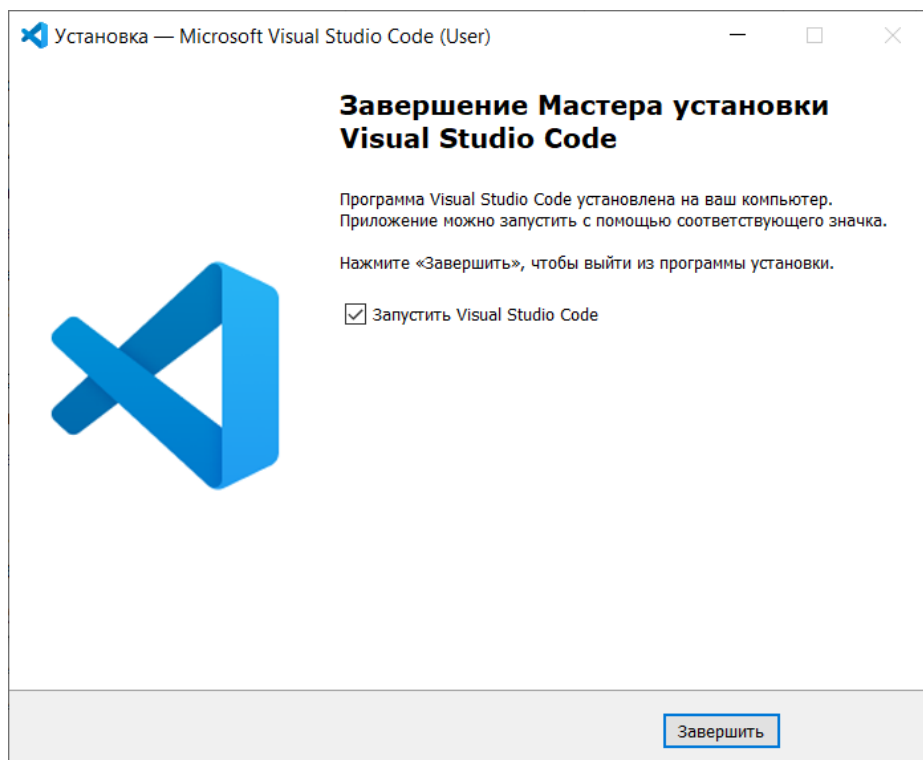


Рисунок 8 – Завершение установки Visual Studio Code

5.5 Открываем окно Extensions

5.6 Находим и устанавливаем расширение C/C++ (рисунок 9)

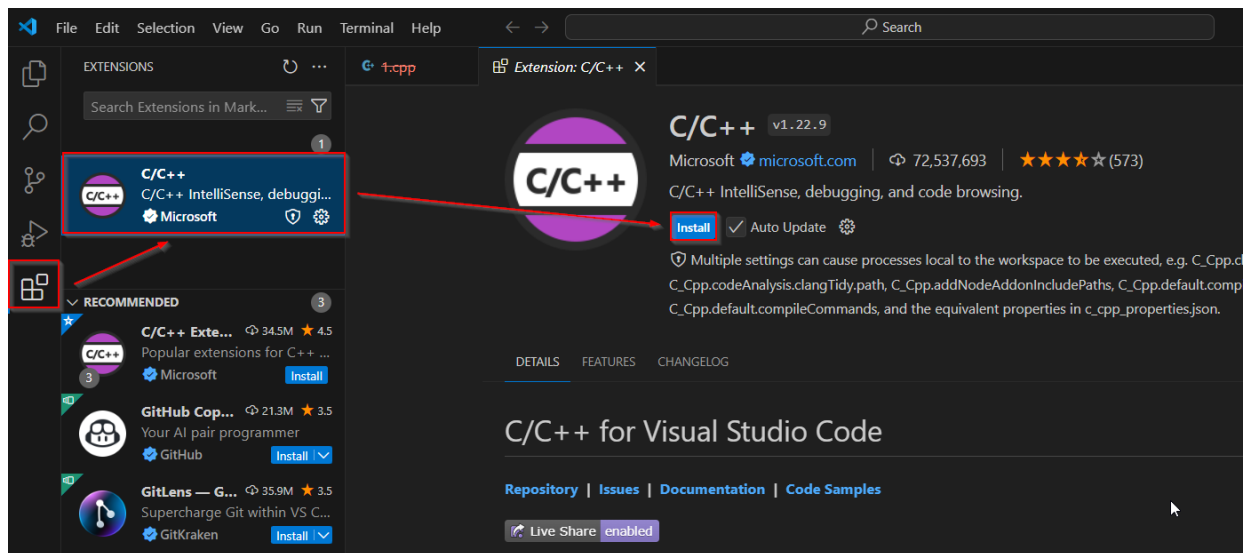


Рисунок 9 – Установка расширения

6 Написание программы, компиляция и запуск

6.1 Нажимаем «File» -> «New File»

6.2 Вводим имя файла и нажимаем «Create File»

6.3 Пишем программу, которая должна вывести в терминал: Hello, World!

6.4 Компилируем и запускаем (рисунок 10)

```
Extension: C/C++ Extension Pack  lab1.cpp x
C: > Users > Nikolai > Desktop > лабораторные алгоритмизация > лабораторная работа №1 > lab1.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      cout << "Hello, World!";
7  }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\Nikolai> cd "C:\Users\Nikolai\Desktop\лабораторные алгоритмизация"
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> g++ lab1.cpp -o app
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> ./app
Hello, World!
PS C:\Users\Nikolai\Desktop\лабораторные алгоритмизация> 
```

Рисунок 10 – Компиляция и запуск программы

7 Установка Visual Studio Community 2022

7.1 Переходим по ссылке: <https://visualstudio.microsoft.com/ru/>

7.2 Загружаем установщик Visual Studio Community 2022

7.3 После загрузки открываем файл, принимаем лицензионное соглашение, а также разрешаем приложению вносить изменения на устройстве, нажав «Да»

7.4 Во время установки выбираем пакеты, которые будут установлены дополнительно (рисунок 11)

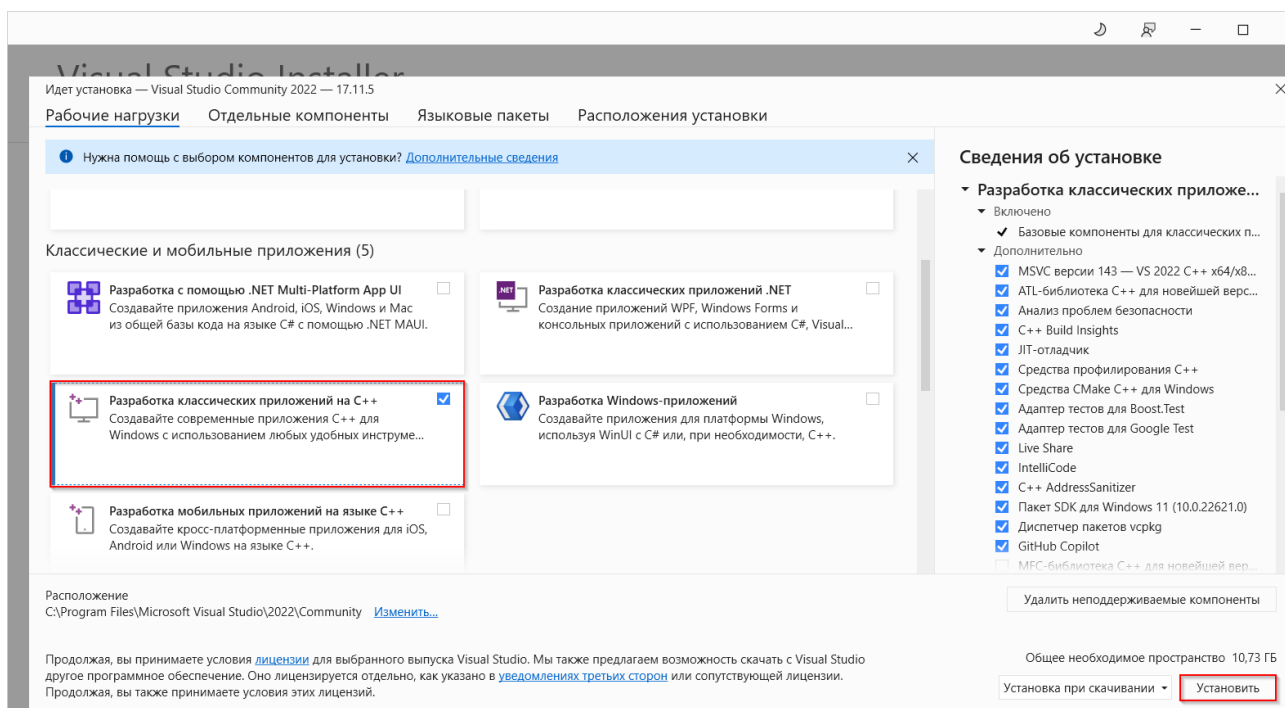


Рисунок 11 – Установка дополнительных пакетов

7.5 После установки приложения создаем консольное приложение и пишем программу

7.6 Далее нажимаем F5 для запуска отладчика (рисунок 12)

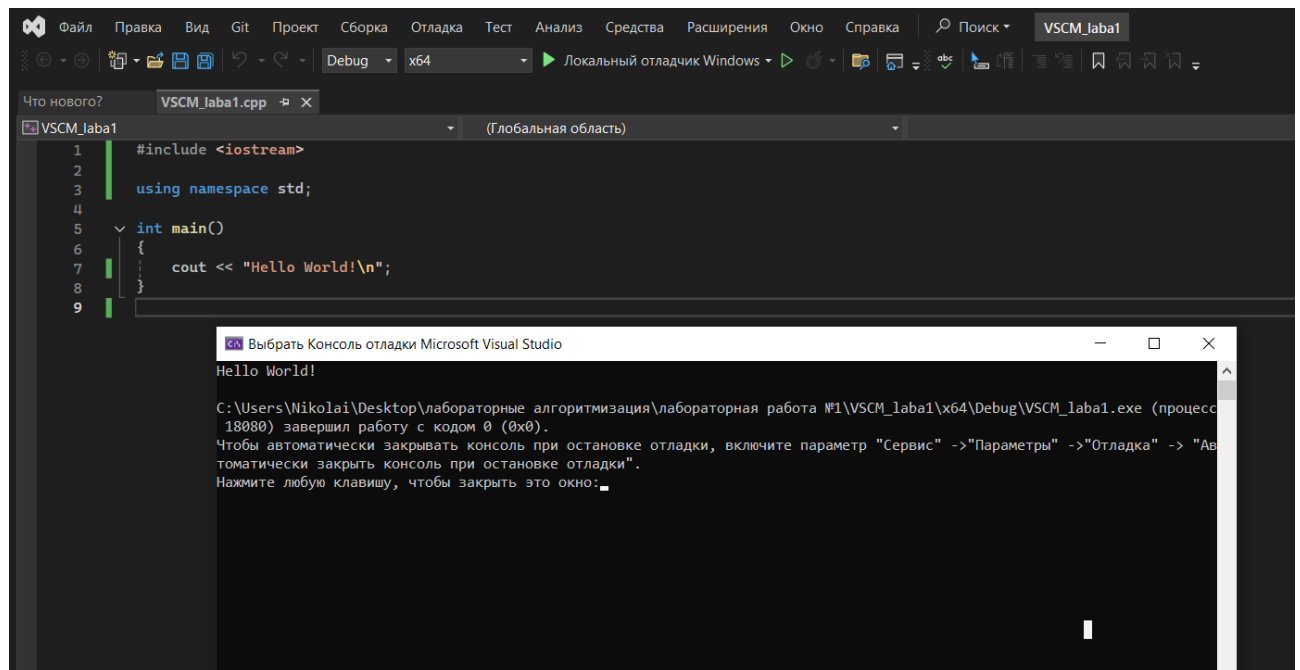


Рисунок 12 – Отладка программы