

Курс C++ && промышленное программирование

Лектор:

Илья Дединский, МФТИ, ded@ded32.ru

Менторы:

Олег Пономарев, Яндекс, oponomarev@gmail.com

Михаил Рахманов, Rambler, rakhmanov.m@gmail.com

- **Образовательные проекты Mail.Ru**
 - Техноатом, Технотрек, Технопарк, Техносфера...
- **Особенности**
 - Преодоление разрыва между учебой и работой
 - Много практики и проектной подготовки
 - Современные технологии
 - Есть преимущества, но нет обязательств по работе в Mail.Ru
 - Ценится работодателями

Отметьтесь на ТЕХНОАТОМЕ!

- Прямо сейчас
- Пожалуйста :)

<http://atom.mail.ru>

Наш курс

- 14 занятий, 1 семестр, 1 раз в неделю, по 4+ ак. часа
- C++ как развитие C
- Главный вопрос на букву «З»!

Особенности работы

- Команды по 2-3 чел.
- Баллы за каждое задание: $[3..10] * (\text{вес задания}) \rightarrow [0..100]$
- Итоговое задание – курсовой проект (вес = 50%)
 - Складывается из ранее сделанных заданий
- Рефакторинг и перфекционизм
- Взаимодействие с ментором
- Сроки – в конце лекции

Общее и базовое

- Б. Страуструп, «Программирование. Принципы и практика с использованием С++» (2-е издание)
- Б. Страуструп, «Дизайн и эволюция языка С++»
- С. Прата, «Язык С++ для начинающих»
- Дж. Лафоре, «Объектно-ориентированное программирование в С++»

Более глубокое

- С. Мейерс, «Эффективное использование C++» (все 3 книги)
- Г. Саттер, А. Александреску, «Стандарты программирования на C++»
- Г. Саттер, «Решение сложных задач на C++», «Новые сложные задачи на C++»
- Дж. Элджер, «Эффективный C++»
- Д. Вандевурд, Н. Джосаттис, «Шаблоны C++. Справочник разработчика»
- А. Александреску, «Современное проектирование на C++»
- Дж. Коплиен «Мультипарадигменное проектирование на C++»

Общие вещи

- С. МакКоннелл, «Совершенный код»
- М. Фаулер, «Рефакторинг»
- Ю. Хсих, «Наука отладки»

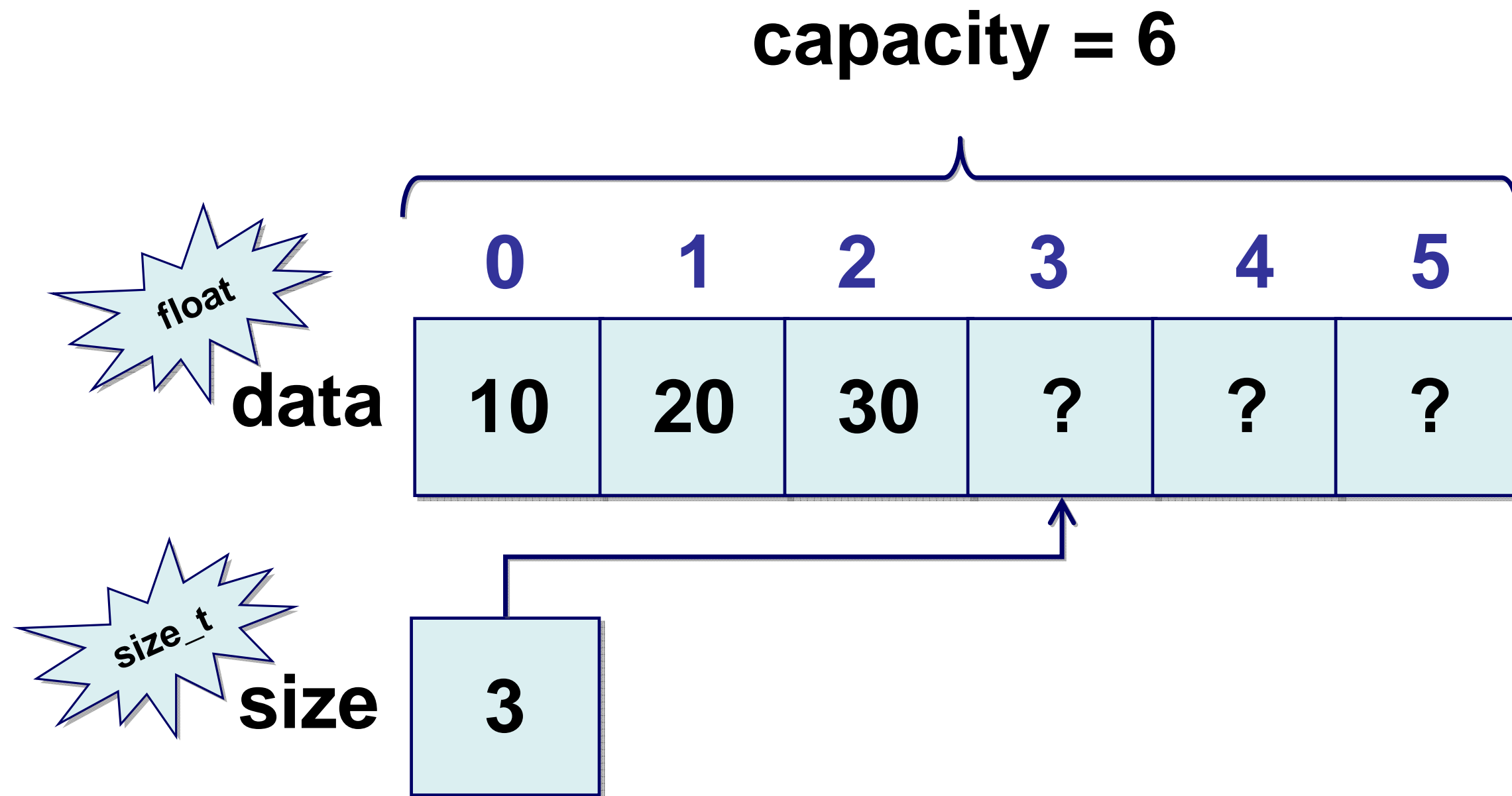
Паттерны проектирования

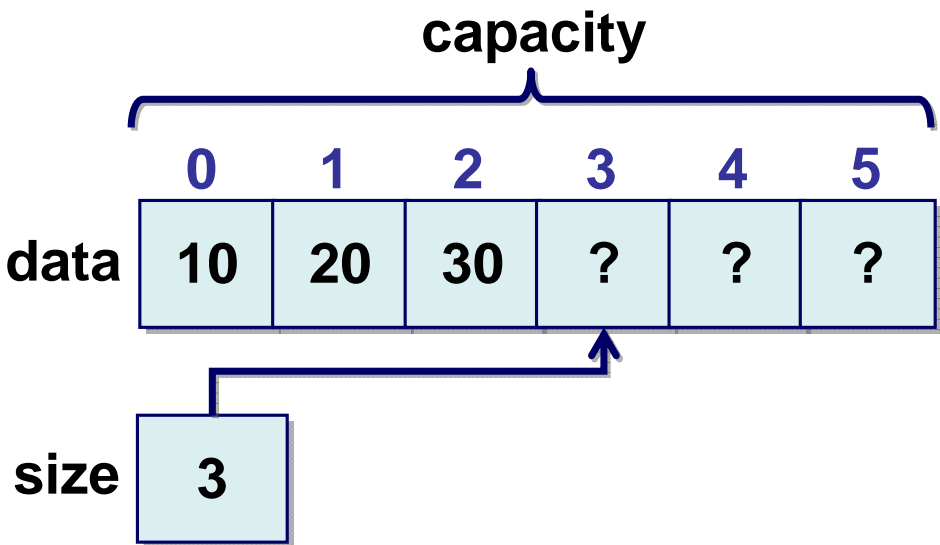
- Дж. Влиссидес, «Применение шаблонов проектирования»
- Э. Гамма, Г. Хелм, Г. Джонсон, Дж. Влиссидес, «Приемы объектно-ориентированного проектирования. Паттерны проектирования»

Философия :)

- Р. Брукс, «Мифический человеко-месяц»
- Э. Хант, Д. Томас, «Программист-прагматик»

- Софт-процессор (виртуальный CPU)
 - Ассемблер и дизассемблер для виртуального процессора
- ~~Мультиагентная система с графическим отображением информации и возможностью интерактивного изменения параметров~~ Игра :)
 - Графика
 - Моддинг и плагины
 - Сеть



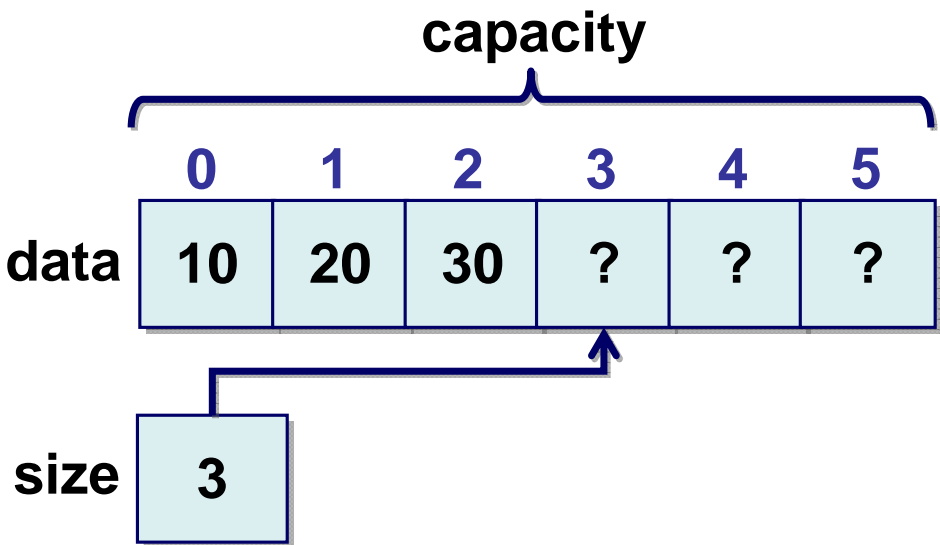


```
float d[6];
int s ;

int main() {
    d[s++]=42;
    float a=d[--s];
    if(a==42) abort; }
```



FFFFUUUUUUUU!!!11
FU FU



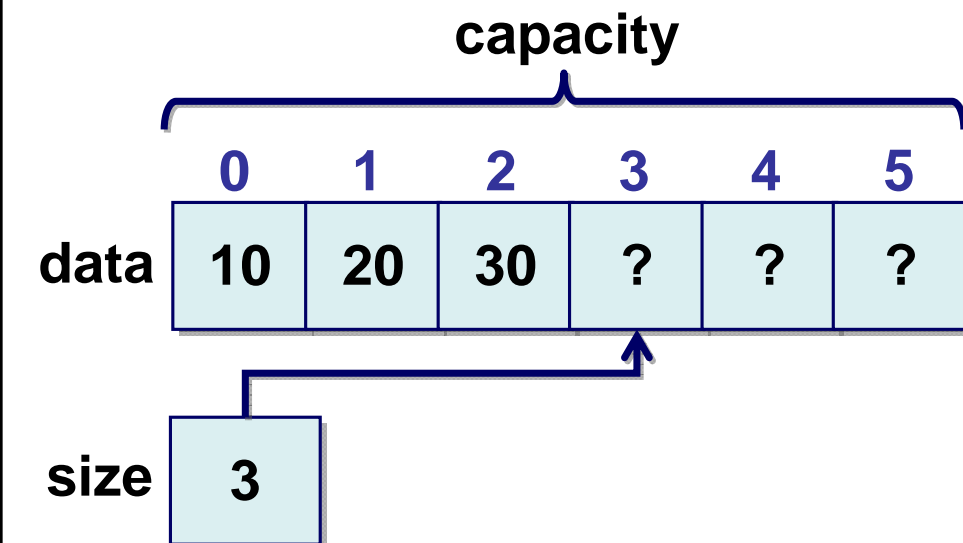
```
float d[6];
int s ;

void pu(float) ;
double po() ;

int main(){
    pu(42);
    float a=po();
    if(a==42)abort();}
```

```
void pu (float va)
{d[s++];
}
float pop() {
    return d [--s]; }
```

FFFFUUUUUUUU!!!11
FU FU



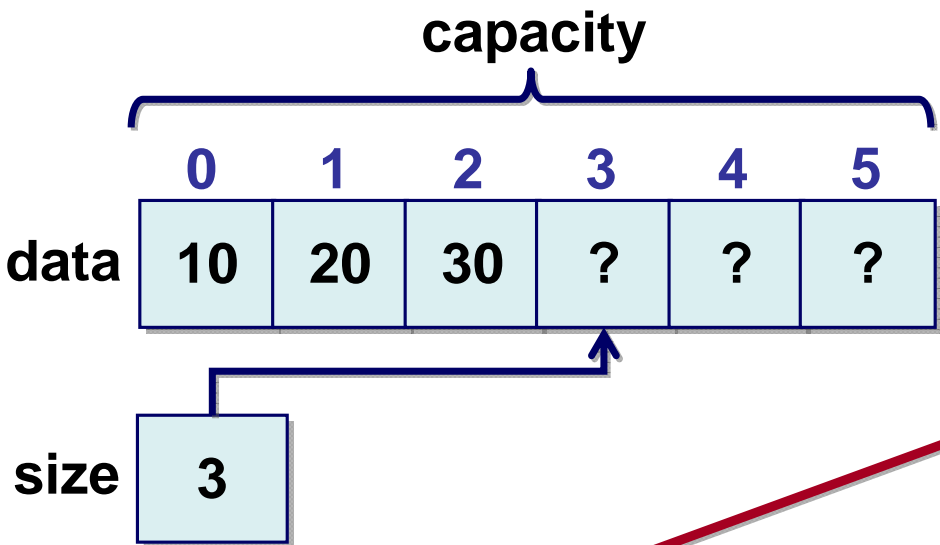
```
float data[6];
int size;

void push (float value);
double pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```



```
float data[6];
int size;

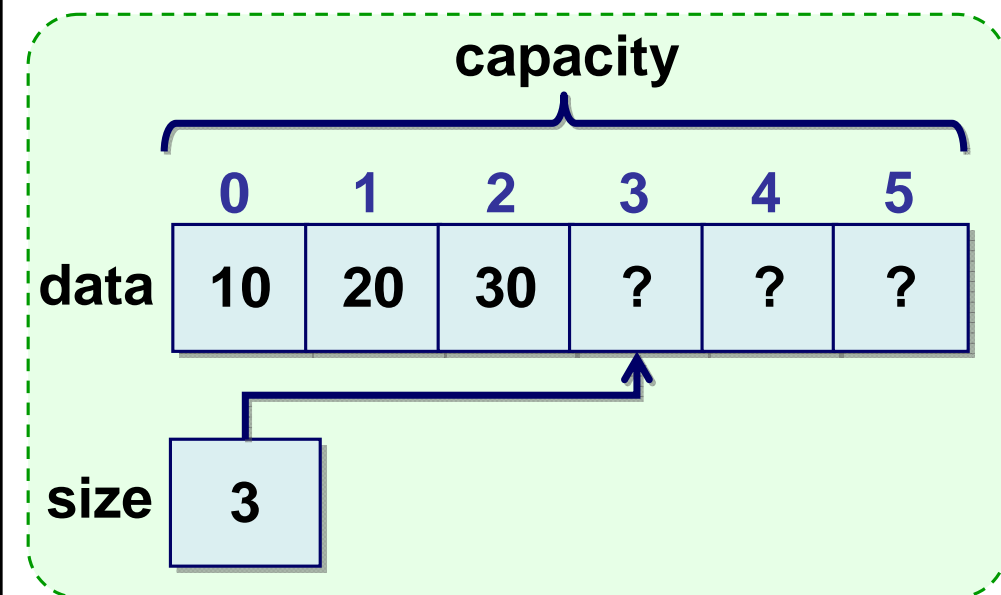
void push (float value);
float pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```

- Проблемы кода:
- Фиксированный тип элементов
- «Магические числа»
- Глобальные данные
- Неявная группировка данных и функций
- Нет индикации ошибок
- Нет проверок на переполнение
- Неясно, что возвращать в случае ошибки
- Что, если данные испорчены?



```
struct Stack
{
    float data [StkCapacity];
    size_t size;
};
```

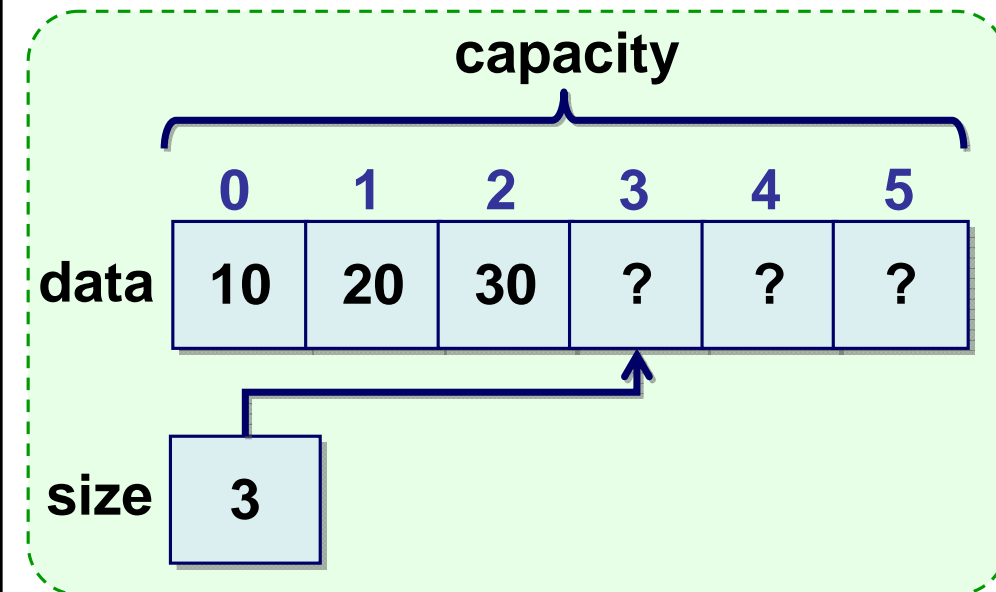
```
float data[6];
int size;

void push (float value);
float pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```

```
float data[6];
int size;

void push (float value);
float pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```

```
const size_t StkCapacity = 6;
typedef float StkValue_type;

//-----

struct Stack
{
    StkValue_type data [StkCapacity];
    size_t        size;
};

//-----

bool          StkPush (Stk* thisStk);
StkValue_type StkPop  (Stk* thisStk);

//-----

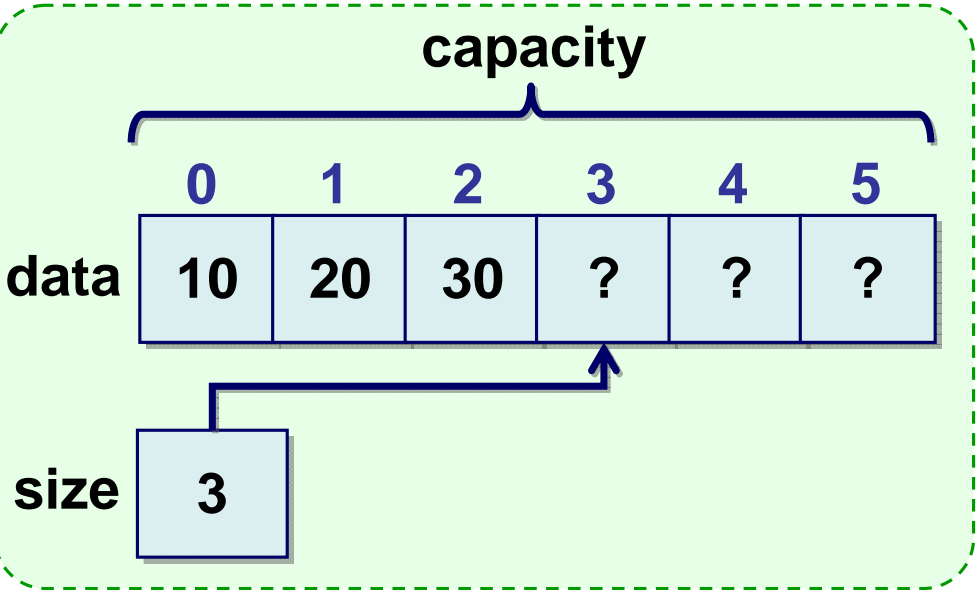
int main()
{
    Stack s1 = {}; InitStack (&s1);
    Stack s2 = {}; InitStack (&s2);

    StkPush (&s1, 42);
    StkPush (&s2, StkPop (&s1));
    TEST (StkPop (&s2) == 42);
}

//-----

bool push (Stk* thisStk, ...)
{
    if (thisStk->size >= StkCapacity)
        return false;

    thisStk->data [thisStk->size++]...;
    ...
}
```



```
float data[6];
int size;

void push (float value);
float pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```

```
const size_t StkCapacity = 6;
typedef float StkValue_type;

//-----
struct Stack
{
    StkValue_type data [StkCapacity];
    size_t size;
};

//-----
bool StkPush (Stk* thisStk);
StkValue_type StkPop (Stk* thisStk);

//-----
int main()
{
    Stack s1 = {}; InitStack (&s1);
    Stack s2 = {}; InitStack (&s2);

    StkPush (&s1, 42);
    StkPush (&s2, StkPop (&s1));
    TEST (StkPop (&s2) == 42);
}

//-----
bool push (Stk* thisStk, ...)
{
    if (thisStk->size >= StkCapacity)
        return false;

    thisStk->data [thisStk->size++]...;
    ...
}
```

Проблемы кода:

Неявная группировка элементов кода

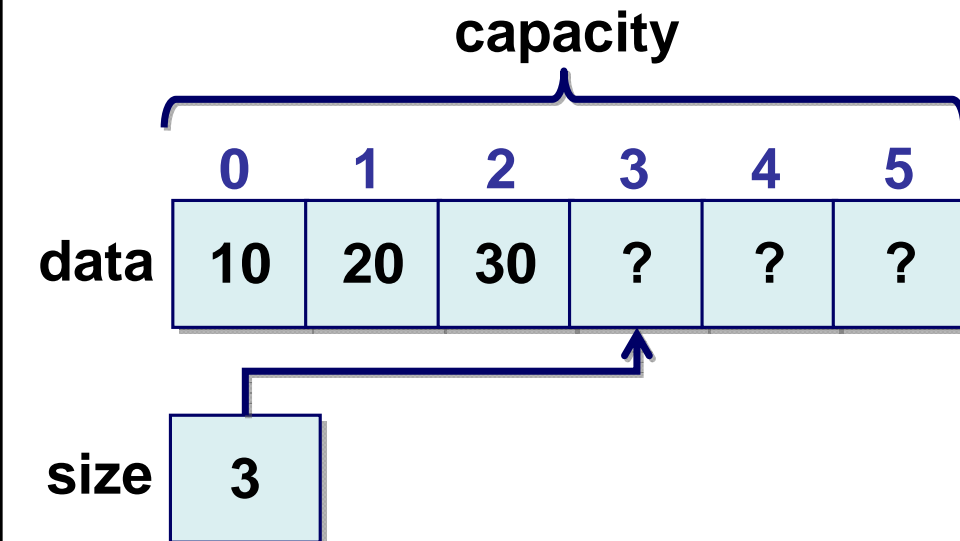
Необходимость вызова инициализатора

Навязчивая передача переменной объекта

Навязчивое указание имени объекта

Что, если данные испорчены?

Неясно, что возвращать в случае ошибки pop()



```
float data[6];
int size;

void push (float value);
float pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```

```
const size_t StkCapacity = 6;
typedef float StkValue_type;
//-----

struct Stack
{
    StkValue_type data [StkCapacity];
    size_t size;
};
//-----

bool StkPush (Stk* thisStk);
StkValue_type StkPop (Stk* thisStk);
//-----

int main()
{
    Stack s1 = {}; InitStack (&s1);
    Stack s2 = {}; InitStack (&s2);

    StkPush (&s1, 42);
    StkPush (&s2, StkPop (&s1));
    TEST (StkPop (&s2) == 42);
}
//-----

bool push (Stk* thisStk, ...)
{
    if (thisStk->size >= StkCapacity)
        return false;

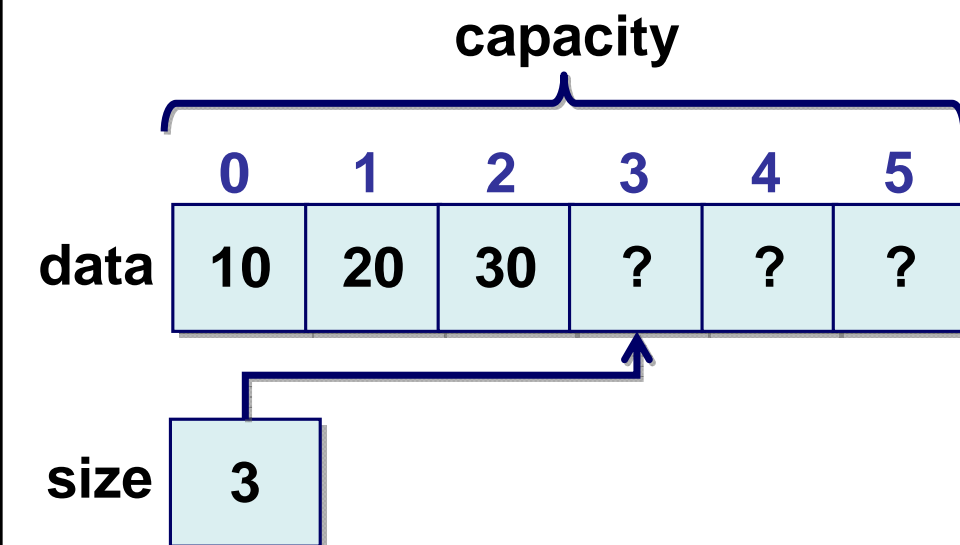
    thisStk->data [thisStk->size++]...;
    ...
}
```

```
struct Stack
{
    static const size_t capacity = 6;

    typedef float value_type;

    value_type data [capacity];
    size_t size;

    bool push (value_type value);
    value_type pop ();
};
```



```
float data[6];
int size;

void push (float value);
double pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```

```
const size_t StkCapacity = 6;
typedef float StkValue_type;
//-----

struct Stack
{
    StkValue_type data [StkCapacity];
    size_t size;
};
//-----

bool StkPush (Stk* thisStk);
StkValue_type StkPop (Stk* thisStk);
//-----

int main()
{
    Stack s1 = {}; InitStack (&s1);
    Stack s2 = {}; InitStack (&s2);

    StkPush (&s1, 42);
    StkPush (&s2, StkPop (&s1));
    TEST (StkPop (&s2) == 42);
}
//-----

bool push (Stk* thisStk, ...)
{
    if (thisStk->size >= StkCapacity)
        return false;

    thisStk->data [thisStk->size++]...;
    ...
}
```

```
struct Stack
{
    static const size_t capacity = 6;

    typedef float value_type;

    value_type data [capacity];
    size_t size;

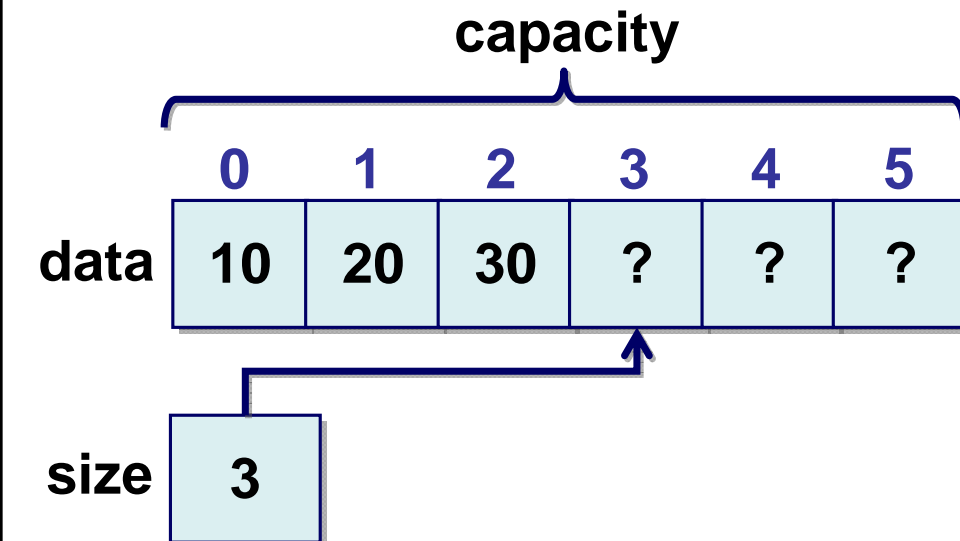
    bool push (value_type value);
    value_type pop ();
};
//-----

int main()
{
    Stack s1 = {}; InitStack (&s1);
    Stack s2 = {}; InitStack (&s2);

    s1.push (42);
    s2.push (s1.pop());
    TEST (s2.pop() == 42);
}
//-----

bool Stack::push (value_type value)
{
    if (this->size >= this->capacity)
        return false;

    this->data [this->size++] = value;
    ...
}
```

```
float data[6];
int size;

void push (float value);
double pop();

int main()
{
    push (42);
    float val = pop();
    TEST (val == 42);
}

void push (float value)
{
    data [size++];
}

float pop()
{
    return data [--size];
}
```

```
const size_t StkCapacity = 6;
typedef float StkValue_type;
//-----

struct Stack
{
    StkValue_type data [StkCapacity];
    size_t        size;
};
//-----

bool          StkPush (Stk* thisStk);
StkValue_type StkPop  (Stk* thisStk);
//-----

int main()
{
    Stack s1 = {}; InitStack (&s1);
    Stack s2 = {}; InitStack (&s2);

    StkPush (&s1, 42);
    StkPush (&s2, StkPop (&s1));
    TEST (StkPop (&s2) == 42);
}
//-----

bool push (Stk* thisStk, ...)
{
    if (thisStk->size >= StkCapacity)
        return false;

    thisStk->data [thisStk->size++]...;
    ...
}
```

```
struct Stack
{
    static const size_t capacity_ = 6;

    typedef float value_type;

    value_type data_ [capacity_];
    size_t      size_;

    bool          push (value_type value);
    value_type pop  ();
};
//-----

int main()
{
    Stack s1 = {}; InitStack (&s1);
    Stack s2 = {}; InitStack (&s2);

    s1.push (42);
    s2.push (s1.pop());
    TEST (s2.pop() == 42);
}
//-----

bool Stack::push (value_type value)
{
    if (size_ >= capacity_)
        return false;

    data_ [size_++] = value;
    ...
}
```

```
//-----  
//! @file Stack.h  
//! Implements a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
//-----  
//! Stack class  
//-----  
  
class Stack  
{  
public:  
    typedef float value_type; //!< Elements type  
  
    Stack();  
    // Stack (size_t capacity);  
  
    ~Stack();  
  
    //-----  
    //!< Pushes a value to the stack  
    //!< @param value is an element to be pushed  
    //!< @return success of operation  
    //-----  
  
    bool push (value_type value);  
  
    //...  
  
    value_type pop();  
    size_t capacity();  
  
private:  
    static const size_t capacity_ = 6;  
  
    value_type data_ [capacity];  
    size_t size_;  
};
```

Документация

Общедоступные члены
(Интерфейс класса)

Конструктор(ы)

Деструктор


Другие методы класса

Геттер

Члены, доступные
только внутри класса

Статический член
1 экз. для всех объектов


```
//-----  
//! @file Stack.h  
//! Implements a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
//-----  
//! Stack class  
//-----  
  
class Stack  
{  
public:  
    typedef bloat value_type; //!< Elements type  
  
    Stack();  
    // Stack (size_t capacity);  
  
    ~Stack();  
  
    //-----  
    //! Pushes a value to the stack  
    //! @param value is an element to be pushed  
    //! @return success of operation  
    //-----  
  
    bool push (value_type value);  
  
    //...  
  
    value_type pop();  
    size_t      capacity();  
  
private:  
    static const size_t capacity_ = 6;  
  
    value_type data_ [capacity];  
    size_t      size_;  
};
```



The Nuclear Stack Library (NSL) 1.0

[Main Page](#) [Classes](#) [Files](#)

The Nuclear Stack Library (NSL)

Classes

Class List

Stack

Class Index

Class Members

Files

Stack Class Reference

Public Types | Public Member Functions | Static Public Attributes | Private Attributes | List of all members.

Implements a stack class. More...

#include <Stack.h>

Collaboration diagram for Stack:

Stack
+ capacity_
- data_
- size_
+ Stack()
+ ~Stack()
+ push()
+ pop()
+ clear()
+ size()
+ ok()
+ dump()

Public Types


typedef double value_type
Elements type. More...

Public Member Functions

Stack ()
~Stack ()
bool push (value_type value)
value_type pop ()
void clear ()
size_t size () const
bool ok () const
bool dump () const

Static Public Attributes

```
//-----  
//! @file Stack.h  
//! Implements a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
//-----  
//! Stack class  
//-----  
  
class Stack  
{  
public:  
    typedef goat value_type;  //!< Elements type  
  
    Stack();  
    // Stack (size_t capacity);  
  
    ~Stack();  
  
    //-----  
    //!< Pushes a value to the stack  
    //!< @param value is an element to be pushed  
    //!< @return success of operation  
    //-----  
  
    bool push (value_type value);  
  
    //...  
  
    value_type pop();  
    size_t      capacity();  
  
private:  
    static const size_t capacity_ = 6;  
  
    value_type data_ [capacity];  
    size_t      size_;  
};
```



The Nuclear Stack Library (NSL) 1.0

[Main Page](#) [Classes](#) [Files](#)

The Nuclear Stack Library (NSL)

- Classes
 - Class List
 - Stack
 - Class Index
 - Class Members
 - Files

Public Types | Public Member Functions | Static Public Attributes | Private Attributes | List of all members.

Stack Class Reference

Implements a stack class. [More...](#)

Stack

push()

bool Stack::push (value_type value)

Pushes a value to the stack

Parameters
value is an element to be pushed

Returns
success of operation

Adds an element to the stack.

Definition at line 97 of file [Stack.h](#).

98 {
99 ASSERT_OK();
100
101 // TODO Feed the cats
102
103 ASSERT_OK();
104 return true;
105 }

Here is the caller graph for this function:

Stack::push

ALMOST_GOOGLETEST_StackTest_42

main

Generated by [doxygen](#) 1.8.13

23



```
//-----  
//! @file Stack.h  
//! Implements a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
//-----  
//! Stack class  
//-----  
  
class Stack  
{  
public:  
    typedef coat value_type; //!< Elms type  
  
    Stack();  
    // Stack (size_t capacity);  
  
    ~Stack();  
  
    //-----  
    //! Pushes a value to the stack  
    //! @param value is an element to be pushed  
    //! @return success of operation  
    //-----  
  
    bool push (value_type value);  
  
    //...  
  
    value_type pop();  
    size_t capacity();  
  
private:  
    static const size_t capacity_ = 6;  
  
    value_type data_ [capacity];  
    size_t size_;  
};
```

```
bool Stack::Stack() :  
    size_ (0)  
{  
    printf (__PRETTY_FUNCTION__ "\n");  
}  
  
//-----  
bool Stack::~Stack()  
{  
    size_ = POISON_INT;  
    ...  
}  
  
//-----  
bool Stack::push (value_type value)  
{  
    if (size >= capacity)  
        return false;  
    data [Stack->size++];  
    ...  
}  
  
//-----  
...
```

```
//-----  
//! @file StackTest.cpp  
//! Tests a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
#include "Stack.h"  
  
//-----  
//! A simple unit-test  
//! @param condition is a test expr  
//-----  
  
#define TEST( condition ) \  
    if (! (condition)) \  
        printf ("Test FAILED: " #condition);  
  
//-----  
//! main  
//-----  
  
int main()  
{  
    Stack s1;    // Stack::Stack() for s1  
  
    {  
        Stack s2;    // Stack::Stack() for s2  
  
        s1.push (42);  
        s2.push (s1.pop());  
        TEST (s2.pop() == 42);  
    }    // Stack::~Stack() for s2  
}  
    // Stack::~Stack() for s1
```



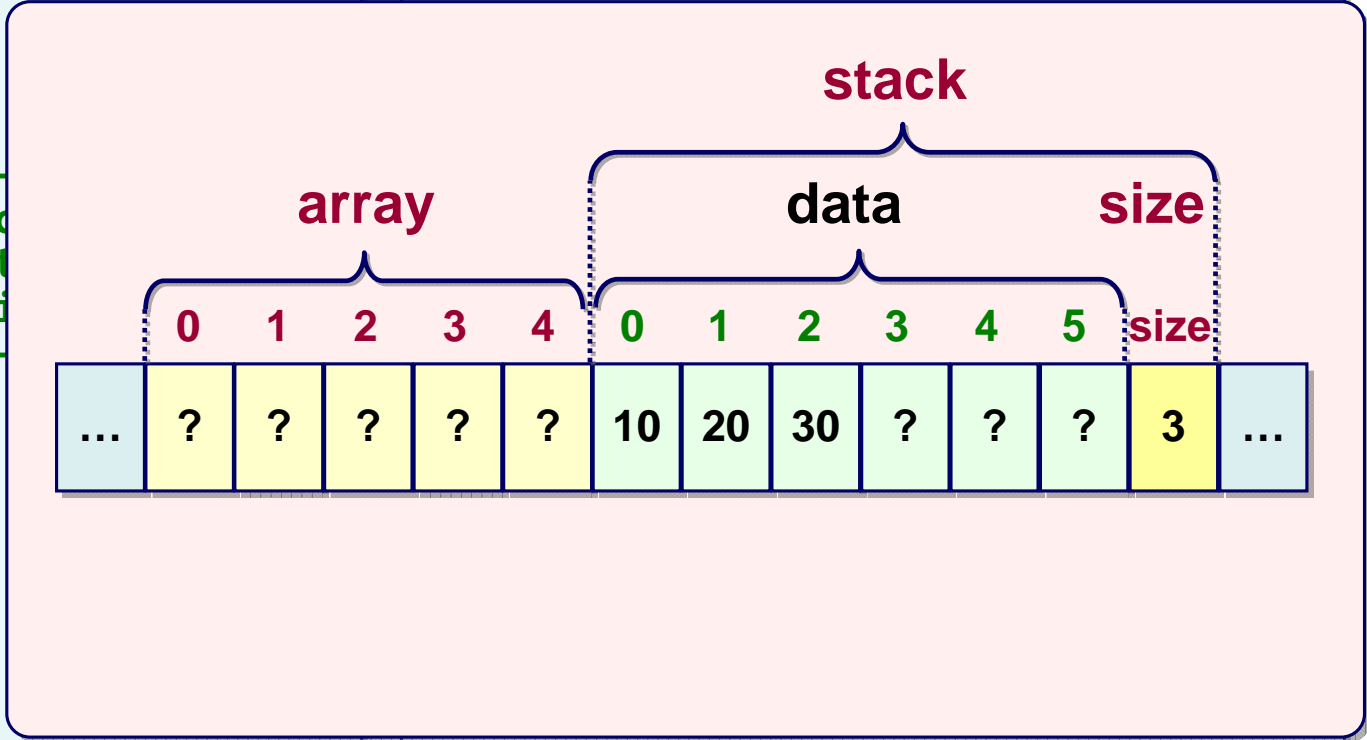
```
//-----  
//! @file Stack.h  
//! Implements a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
//-----  
//! Stack class  
//-----  
  
class Stack  
{  
public:  
    typedef throat value_type; //!< Elms type  
  
    Stack();  
    // Stack (size_t capacity);  
  
    ~Stack();  
  
    //-----  
    //! Pushes a value to the stack  
    //! @param value is an element to be pushed  
    //! @return success of operation  
    //-----  
  
    bool push (value_type value);  
  
    //...  
  
    value_type pop();  
    size_t      capacity();  
  
private:  
    static const size_t capacity_ = 6;  
  
    value_type data_ [capacity];  
    size_t      size_;  
};
```

```
bool Stack::Stack() :  
    size_ (0)  
{  
    printf (__PRETTY_FUNCTION__ "\n");  
}  
  
//-----  
bool Stack::~Stack()  
{  
    size_ = POISON_INT;  
    ...  
}  
  
//-----  
bool Stack::push (value_type value)  
{  
    if (size >= capacity)  
        return false;  
    data [Stack->size++];  
    ...  
}  
  
//-----  
...
```

```
//-----  
//! @file StackTest.cpp  
//! Tests a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
#include "Stack.h"  
  
//-----  
//! A simple unit-test  
//! @param condition is a test expr  
//-----  
  
#define TEST( condition ) \  
    if (! (condition)) \  
        printf ("Test FAILED: " #condition);  
  
//-----  
//! main  
//-----  
  
int main()  
{  
    Stack s1;    // Stack::Stack() for s1  
  
    int array[5];  
    array[11] = 100500;    // 64-bit: [14]  
  
    {  
        Stack s2;    // Stack::Stack() for s2  
        s1.push (42);  
        s2.push (s1.pop());  
        TEST (s2.pop() == 42);  
    }    // Stack::~Stack() for s2  
}  
    // Stack::~Stack() for s1
```

```
//-----  
//! @file Stack.h  
//! Implements a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
//-----  
//! Stack class  
//-----  
  
class Stack  
{  
public:  
    typedef borat value_type; //!< Elems type  
  
    Stack();  
    // Stack (size_t capacity);  
  
    ~Stack();  
  
    //-----  
    //! Pushes a value to the stack  
    //! @param value is an element  
    //! @return success of operation  
    //-----  
  
    bool push (value_type value);  
  
    //...  
  
    value_type pop();  
    size_t capacity();  
  
private:  
    static const size_t capacity_ = 6;  
  
    value_type data_ [capacity];  
    size_t size_;  
};
```

```
bool Stack::Stack() :  
    size_ (0)  
{  
    printf (__PRETTY_FUNCTION__ "\n");  
}  
  
//-----  
bool Stack::~~Stack()  
{  
    size_ = POISON_INT;  
    ...  
}  
  
//-----  
...
```



```
//-----  
//! @file StackTest.cpp  
//! Tests a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
#include "Stack.h"  
  
//-----  
//! A simple unit-test  
//! @param condition is a test expr  
//-----  
  
#define TEST( condition ) \  
    if (! (condition)) \  
        printf ("Test FAILED: " #condition);  
  
//-----  
//! main  
//-----  
  
int main()  
{  
    Stack s1;    // Stack::Stack() for s1  
  
    int array[5];  
    array[11] = 100500;    // 64-bit: [14]  
  
    {  
        Stack s2;    // Stack::Stack() for s2  
  
        s1.push (42);  
        s2.push (s1.pop());  
        TEST (s2.pop() == 42);  
    }    // Stack::~~Stack() for s2  
}  
    // Stack::~~Stack() for s1
```

```
//-----
//! @file Stack.h
//! Implements a stack class
//!
//! @author Ded, 2017
//-----

//-----
//! Stack class
//-----

class Stack
{
public:
    typedef borat value_type; //!< Elems type

    Stack();
    // Stack (size_t capacity);

    ~Stack();

    //-----
    //! Pushes a value to the stack
    //! @param value is an element
    //! @return success of operation
    //-----

    bool push (value_type value);

    //...

    value_type pop();
    size_t capacity();

private:
    static const size_t capacity_ = 6;

    value_type data_ [capacity];
    size_t size_;
};
```

```
bool Stack::Stack() :
    size_ (0)
{
    printf (__PRETTY_FUNCTION__ "\n");
}

//-----
bool Stack::~~Stack()
{
    size_ = POISON_INT;
    ...
}

//-----
...

```

```
//-----
//! @file StackTest.cpp
//! Tests a stack class
//!
//! @author Ded, 2017
//-----

#include "Stack.h"

//-----
//! A simple unit-test
//! @param condition is a test expr
//-----

#define TEST( condition ) \
    if (! (condition)) \
        printf ("Test FAILED: " #condition);

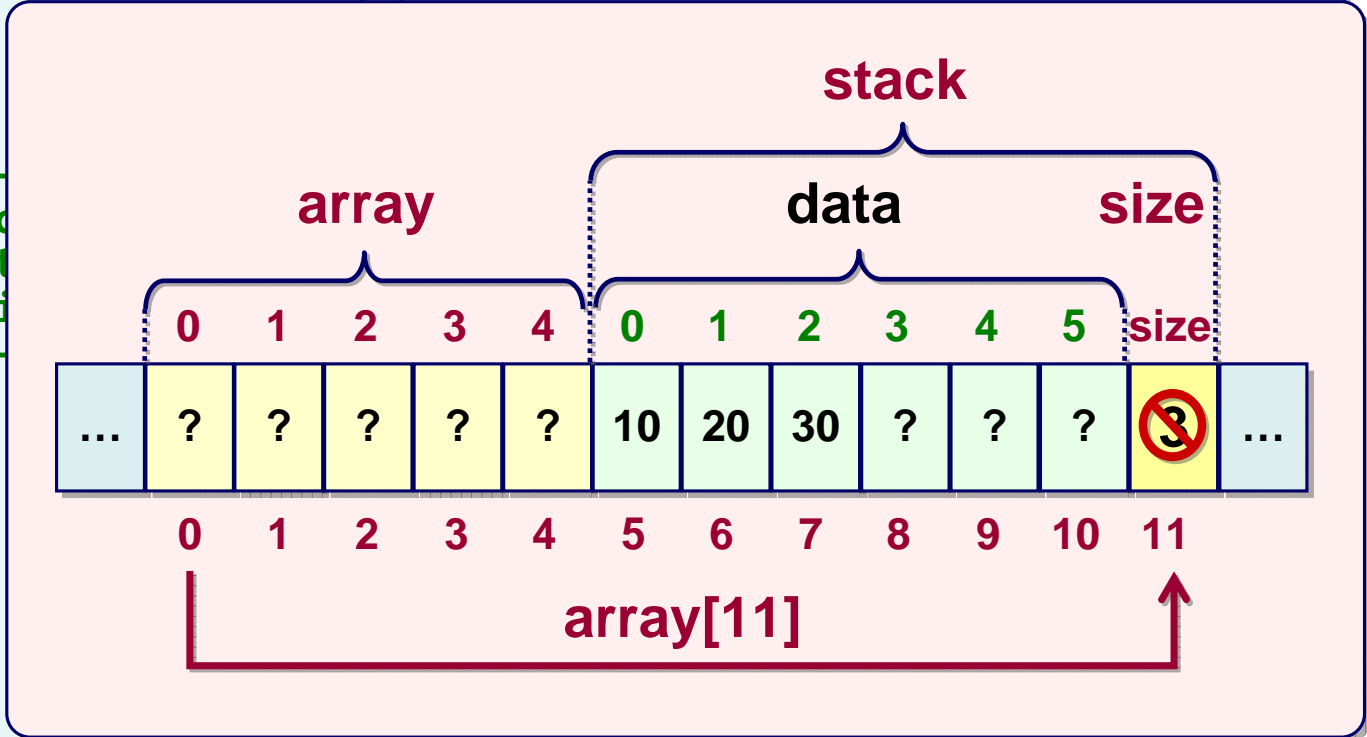
//-----
//! main
//-----

int main()
{
    Stack s1;    // Stack::Stack() for s1

    int array[5];
    array[11] = 100500;    // 64-bit: [14]

    {
        Stack s2;    // Stack::Stack() for s2

        s1.push (42);
        s2.push (s1.pop());
        TEST (s2.pop() == 42);
    }    // Stack::~~Stack() for s2
}    // Stack::~~Stack() for s1
```





```
//-----  
//! @file Stack.h  
//! Implements a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
//-----  
//! Stack class  
//-----  
  
class Stack  
{  
public:  
    typedef your_cat value_type; //!< Elms type  
  
    Stack();  
    // Stack (size_t capacity);  
  
    ~Stack();  
  
    //-----  
    //! Pushes a value to the stack  
    //! @param value is an element to be pushed  
    //! @return success of operation  
    //-----  
  
    bool push (value_type value);  
  
    //...  
  
    value_type pop();  
    size_t      capacity();  
  
private:  
    static const size_t capacity_ = 6;  
  
    value_type data_ [capacity];  
    size_t      size_;  
};
```

```
bool Stack::Stack() :  
    size_ (0)  
{  
    printf (__PRETTY_FUNCTION__ "\n");  
}  
  
//-----  
bool Stack::~Stack()  
{  
    size_ = POISON_INT;  
    ...  
}  
  
//-----  
bool Stack::push (value_type value)  
{  
    if (size >= capacity)  
        return false;  
    data [Stack->size++];  
    ...  
}  
  
//-----  
//! Silent verifier  
//-----  
bool Stack::ok() const  
{  
    return size_ <= capacity_;  
}  
  
//-----  
//! Dumper  
//-----  
bool Stack::dump() const  
{  
    ...  
}
```

```
//-----  
//! @file StackTest.cpp  
//! Tests a stack class  
//!  
//! @author Ded, 2017  
//-----  
  
#include "Stack.h"  
  
//-----  
//! A simple unit-test  
//! @param condition is a test expr  
//-----  
  
#define TEST( condition ) \  
    if (! (condition)) \  
        printf ("Test FAILED: " #condition);  
  
//-----  
//! main  
//-----  
  
int main()  
{  
    Stack s1;    // Stack::Stack() for s1  
  
    int array[5];  
    array[11] = 100500;    // 64-bit: [14]  
  
    {  
        Stack s2;    // Stack::Stack() for s2  
  
        s1.push (42);  
        s2.push (s1.pop());  
        TEST (s2.pop() == 42);  
    }    // Stack::~Stack() for s2  
}  
    // Stack::~Stack() for s1
```



```
//-----
//! @file Stack.h
//! Implements a stack class
//!
//! @author Ded, 2017
//-----

//-----
//! Stack class
//-----

class Stack
{
public:
    typedef meow value_type; //!< Elms type

    Stack();
    // Stack (size_t capacity);

    ~Stack();

    //-----
    //! Pushes a value to the stack
    //! @param value is an element to be pushed
    //! @return success of operation
    //-----

    bool push (value_type value);

    //...

    value_type pop();
    size_t capacity();

private:
    static const size_t capacity_ = 6;

    value_type data_ [capacity];
    size_t size_;
};
```

```
bool Stack::Stack() :
    size_ (0)
{
    printf (__PRETTY_FUNCTION__ "\n");
}


//-----
bool Stack::~~Stack()
{
    size_ = POISON_INT;
    ...
}

//-----
bool Stack::push (value_type value)
{
    if (size >= capacity)
        return false;
    data [Stack->size++];
    ...
}

//-----
//! Silent verifier
//-----

bool Stack::ok() const
{
    return size_ <= capacity_;
}

//-----
//! Dumper
//-----

bool Stack::dump() const
{
    {
        ... 
```

```
Stack (ok)
{
    size_      = 3
    capacity_  = 6
    data [6]:
    {
        * [0] = 10
        * [1] = 20
        * [2] = 30
        [3] = 100500 (POISON_INT)
        [4] = 100500 (POISON_INT)
        [5] = 100500 (POISON_INT)
    }
}
```

```
//-----
//! @file StackTest.cpp
//! Tests a stack class
//!
//! @author Ded, 2017
//-----

#include "Stack.h"

//-----
//! A simple unit-test
//! @param condition is a test expr
//-----

#define TEST( condition ) \
    if (! (condition)) \
        printf ("Test FAILED: " #condition);

//-----
//! main
//-----

int main()
{
    Stack s1;    // Stack::Stack() for s1
```

64-bit: [14]

Stack() for s2

Stack() for s2

Stack() for s1

```
//-----
//! @file Stack.h
//! Implements a stack class
//!
//! @author Ded, 2017
//-----

//-----
//! Stack class
//-----

class Stack
{
public:
    typedef meow! value_type; //!< Elms type

    Stack();
    // Stack (size_t capacity);

    ~Stack();

    //-----
    //! Pushes a value to the stack
    //! @param value is an element to be pushed
    //! @return success of operation
    //-----

    bool push (value_type value);

    //...

    value_type pop();
    size_t capacity();

private:
    static const size_t capacity_ = 6;

    value_type data_ [capacity];
    size_t size_;
};
```

```
bool Stack::Stack() :
    size_ (0)
{
    printf (__PRETTY_FUNCTION__ "\n");
}

//-----
bool Stack::~~Stack()
{
    size_ = POISON_INT;
    ...
}

//-----
bool Stack::push (value_type value)
{
    if (size >= capacity)
        return false;
    data [Stack->size++];
    ...
}

//-----
//! Silent verifier
//-----

bool Stack::ok() const
{
    return size_ <= capacity_;
}

//-----
//! Dumper
//-----

bool Stack::dump() const
{
    {
        ...
```

```
//-----
//! @file StackTest.cpp
//! Tests a stack class
//!
//! @author Ded, 2017
//-----

#include "Stack.h"

//-----
//! A simple unit-test
//! @param condition is a test expr
//-----

#define TEST( condition ) \
    if (! (condition)) \
        printf ("Test FAILED: " #condition);

//-----
//! main
//-----

int main()
{
    Stack s1;    // Stack::Stack() for s1
```

Stack (ERROR)

```
{
    size_      = 100500
    capacity_ = 6
    data [6]:
    {
        * [0] = 10
        * [1] = 20
        * [2] = 30
        [3] = 100500 (POISON_INT)
        [4] = 100500 (POISON_INT)
        [5] = 100500 (POISON_INT)
    }
}
```

64-bit: [14]

ack() for s2

ack() for s2

ack() for s1



```
//-----
//! @file Stack.h
//! Implements a stack class
//!
//! @author Ded, 2017
//-----

//-----
//! Stack class
//-----

class Stack
{
public:
    typedef feed_cat value_type; //!< Elem type

    Stack();
    // Stack (size_t capacity);

    ~Stack();

    //-----
    //! Pushes a value to the stack
    //! @param value is an element to be pushed
    //! @return success of operation
    //-----

    bool push (value_type value);

    //...

    value_type pop();
    size_t    capacity();

private:
    static const size_t capacity_ = 6;

    value_type data_ [capacity];
    size_t    size_;
};
```

```
bool Stack::Stack() :
    size_ (0)
{
    printf (__PRETTY_FUNCTION__ "\n");
}

//-----
bool Stack::~~Stack()
{
    size_ = POISON_INT;
    ...
}

//-----
bool Stack::push (value_type value)
{
    if (size >= capacity)
        return false;
    data [Stack->size++];
    ...
}

//-----
//! Silent verifier
//-----
bool Stack::ok() const
{
    return size_ <= capacity_;
}

//-----
//! Dumper
//-----
bool Stack::dump() const
{
    ...
}
```

```
//-----
//! Macro to test object integrity
//-----
#define ASSERT_OK() \
    if (!ok()) \
    { \
        dump(); \
        assert (!"Object is OK"); \
    }

//-----
bool Stack::push (value_type value)
{
    ASSERT_OK(); // Entry verification

    if (size >= capacity)
        return false;
    data [Stack->size++];
    ...

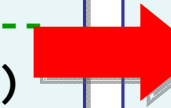
    ASSERT_OK(); // Exit verification
}

//-----
Stack::value_type Stack::pop()
{
    ASSERT_OK(); // Entry verification

    value_type result = ...
    ...

    ASSERT_OK(); // Exit verification

    return result;
}
```





```
//-----
//! @file StackTest.cpp
//! Implements a stack class tests
//!
//! @author Ded, 2017
//-----

#include "Stack.h"

//-----
//! Macro to test
//! @param what is the object to test
//! @param case is what test to run
//-----

#define TEST( what, case ) ...???...

//-----

int main()
{
    TEST (Stack, PushSingleNumber) // (+)
    {
        Stack s;
        s.push (3802);
        ASSERT_EQ (s.pop() == 0xEDA);
        ASSERT_EQ (s.size() == 0);
    }

    TEST (Stack, CannotPushMore) // (-)
    {
        Stack s;
        for (int i = 0; i < s.capacity(); i++)
            s.push (100 + i);

        ASSERT_EQ (s.push (0xF00D) == false);
    }

    ...
}
```

```
bool Stack::Stack() :
    size_ (0)
{
    printf (__PRETTY_FUNCTION__ "\n");
}

//-----
bool Stack::~~Stack()
{
    size_ = POISON_INT;
    ...
}

//-----
bool Stack::push (value_type value)
{
    if (size >= capacity)
        return false;
    data [Stack->size++];
    ...
}

//-----
//! Silent verifier
//-----
bool Stack::ok() const
{
    return size_ <= capacity_;
}

//-----
//! Dumper
//-----
bool Stack::dump() const
{
    ...
}
```

```
//-----
//! Macro to test object integrity
//-----
#define ASSERT_OK() \
    if (!ok()) \
    { \
        dump(); \
        assert (!"Object is OK"); \
    }

//-----
bool Stack::push (value_type value)
{
    ASSERT_OK(); // Entry verification

    if (size >= capacity)
        return false;
    data [Stack->size++];
    ...

    ASSERT_OK(); // Exit verification
}

//-----
Stack::value_type Stack::pop()
{
    ASSERT_OK(); // Entry verification

    value_type result = ...
    ...

    ASSERT_OK(); // Exit verification

    return result;
}
```

[X:\CPP-Course] - Far (x86)

X:\CPP-Course

Name

..

Stack.cpp

Quick view

Folder "X:\CPP-Course"

Contains:

Folders	0
Files	1
Files size	2
Allocated size	8 (400%)
Cluster size	4 096
Files slack	0 (0%)
MFT overhead	6 (75%)

..

Up

09.02.17 04:52

CPP-Course

T:\1\1 - one file>sx

X: => T:\1\1 - one file

[X:\CPP-Course] - Far (x86)

X:\CPP-Course

Name

..

Stack.bak

Stack.cpp

Stack.h

Stack (2).cpp

Stack Копия.cpp

Stack Копия Копия.cpp

Stack.cpp.bak

Stack.h.bak

StackMEOWMEOWMEOW ^_^_^.cpp

StackNew.bak

StackNew.cpp

StackNew.cpp.save

StackNEWNEWNEW.cpp

StackNNN.cpp

StackNNN.cpp.bak

StackOld.cpp

StackOld.cpp.bak

Quick view

#include <stdio.h>

class CStack

{

};

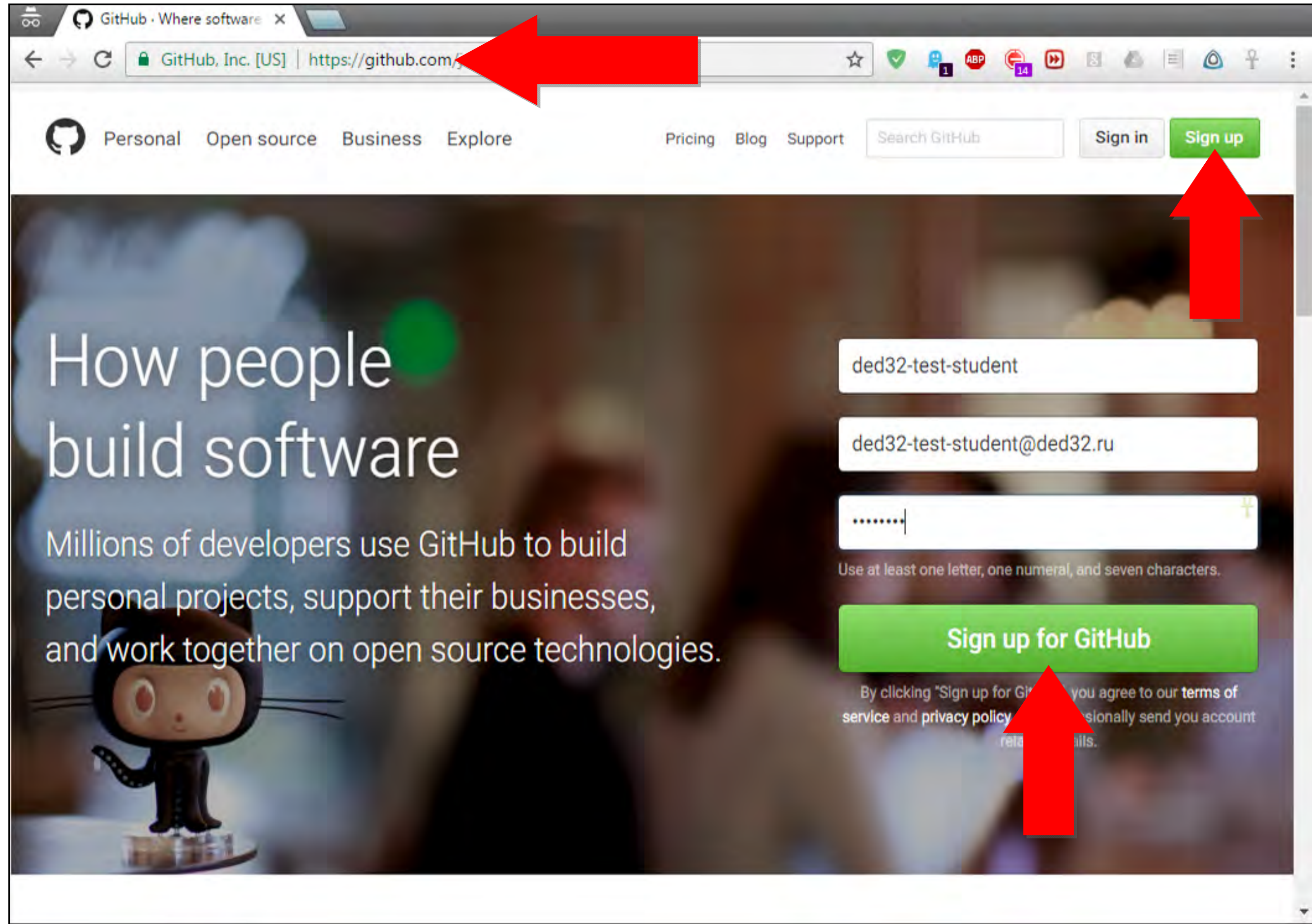
Stack.cpp


55 08.02.17 11:46

Stack.cpp

C++ source f

+



GitHub · Where software  Personal Open source Business Explore Pricing Blog Support Search GitHub Sign in **Sign up**

How people build software

Millions of developers use GitHub to build personal projects, support their businesses, and work together on open source technologies.

ded32-test-student

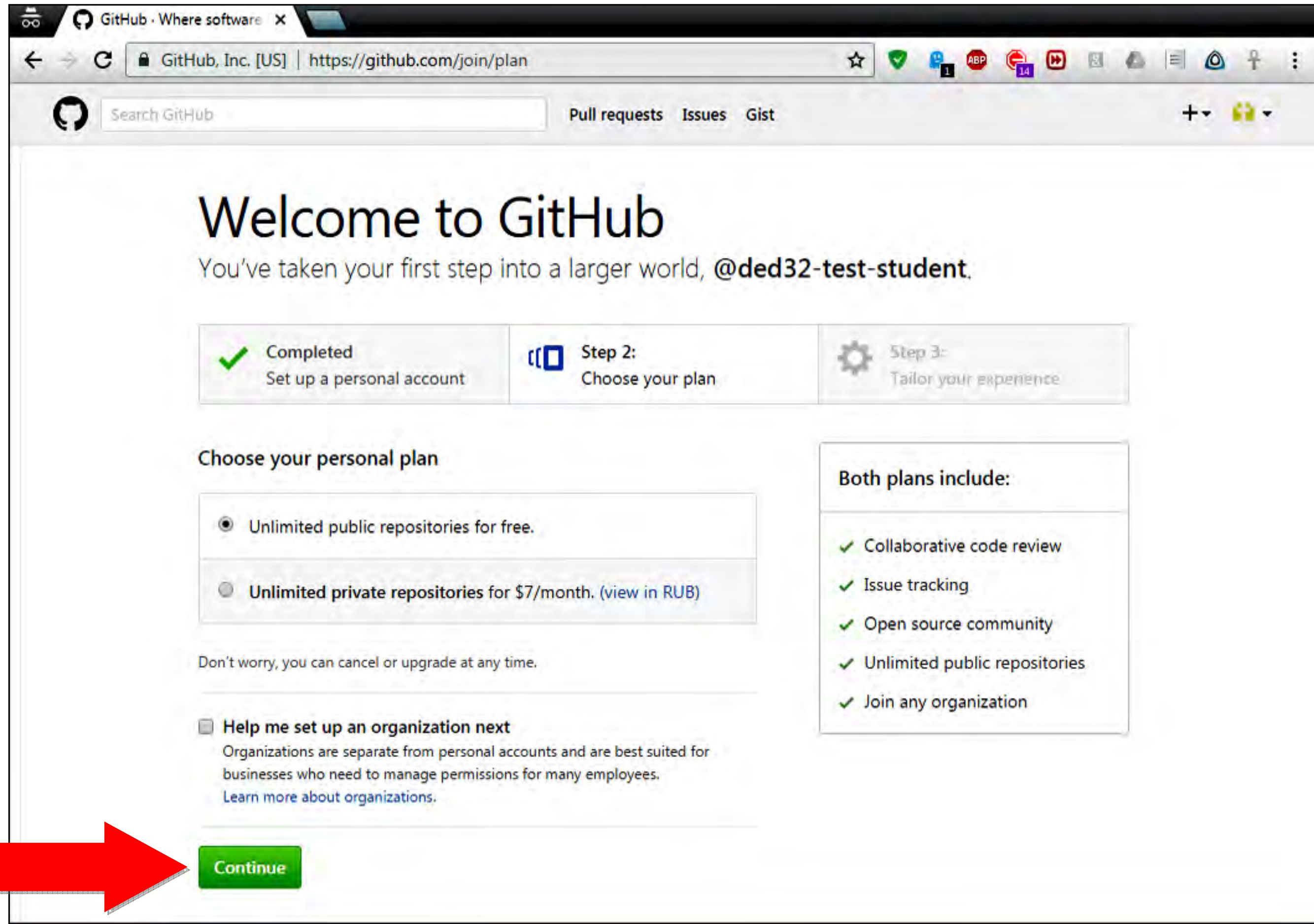
ded32-test-student@ded32.ru

.....


Use at least one letter, one numeral, and seven characters.

Sign up for GitHub

By clicking "Sign up for GitHub" you agree to our [terms of service](#) and [privacy policy](#). We occasionally send you account related emails.



The screenshot shows the GitHub registration page in a web browser. The browser's address bar displays the URL `https://github.com/join/plan`. The page header includes the GitHub logo, a search bar, and navigation links for 'Pull requests', 'Issues', and 'Gist'. The main heading is 'Welcome to GitHub', followed by a personalized message: 'You've taken your first step into a larger world, @ded32-test-student.' Below this, a progress bar shows three steps: 'Step 1: Completed Set up a personal account' (marked with a green check), 'Step 2: Choose your plan' (marked with a blue square icon), and 'Step 3: Tailor your experience' (marked with a gear icon). The 'Choose your personal plan' section offers two options: 'Unlimited public repositories for free.' (selected with a radio button) and 'Unlimited private repositories for \$7/month. (view in RUB)' (unselected). A reassuring note states, 'Don't worry, you can cancel or upgrade at any time.' Below the plan selection, there is a checkbox for 'Help me set up an organization next', with a brief explanation of organizations and a link to 'Learn more about organizations.' At the bottom, a large green 'Continue' button is highlighted by a large red arrow pointing towards it from the left.


GitHub · Where software  x


GitHub, Inc. [US] | `https://github.com/join/plan`


Search GitHub Pull requests Issues Gist

Welcome to GitHub

You've taken your first step into a larger world, @ded32-test-student.

 **Completed**
Set up a personal account

 **Step 2:**
Choose your plan

 **Step 3:**
Tailor your experience

Choose your personal plan

☒ Unlimited public repositories for free.


☐ Unlimited private repositories for \$7/month. [\(view in RUB\)](#)

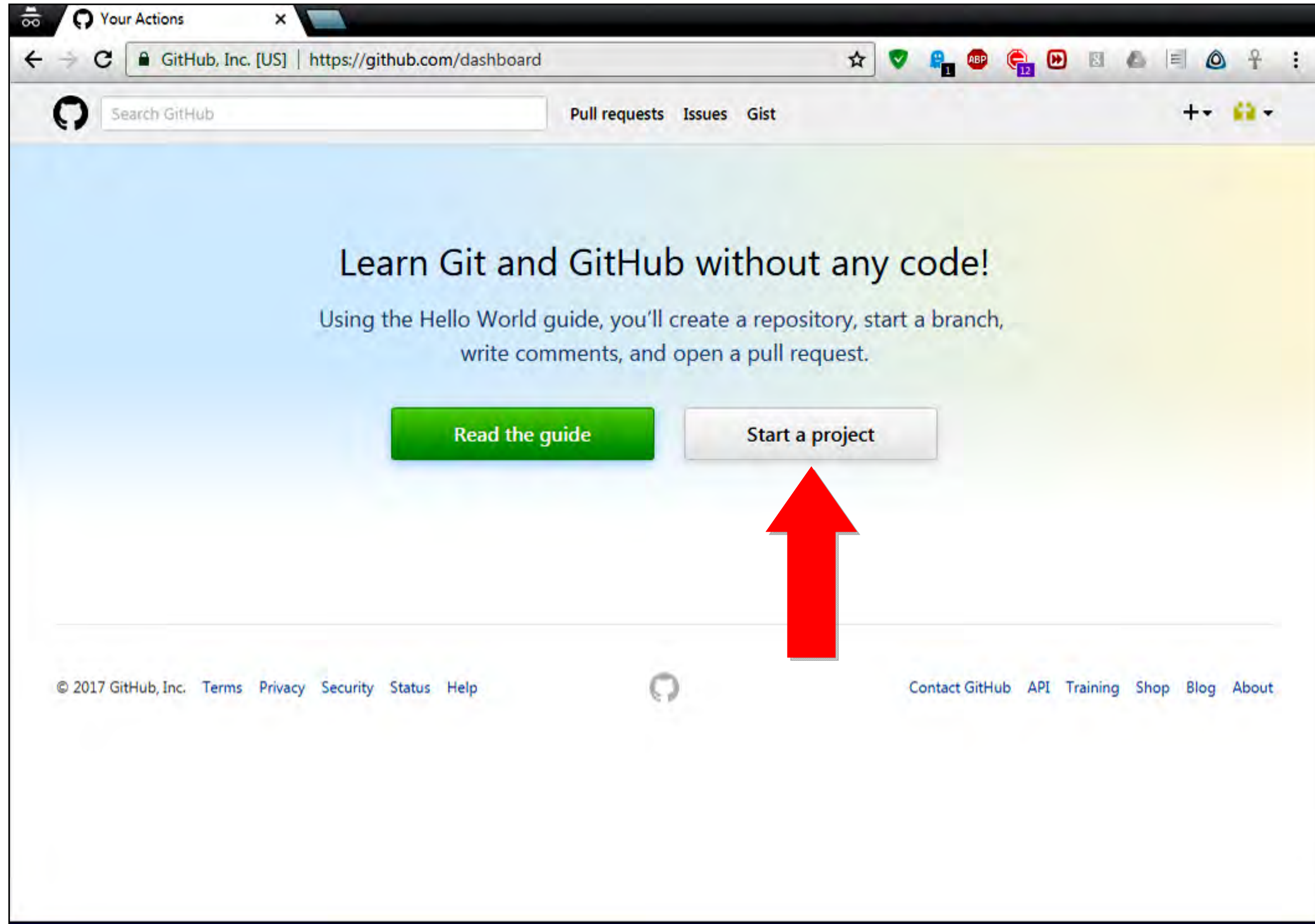
Don't worry, you can cancel or upgrade at any time.

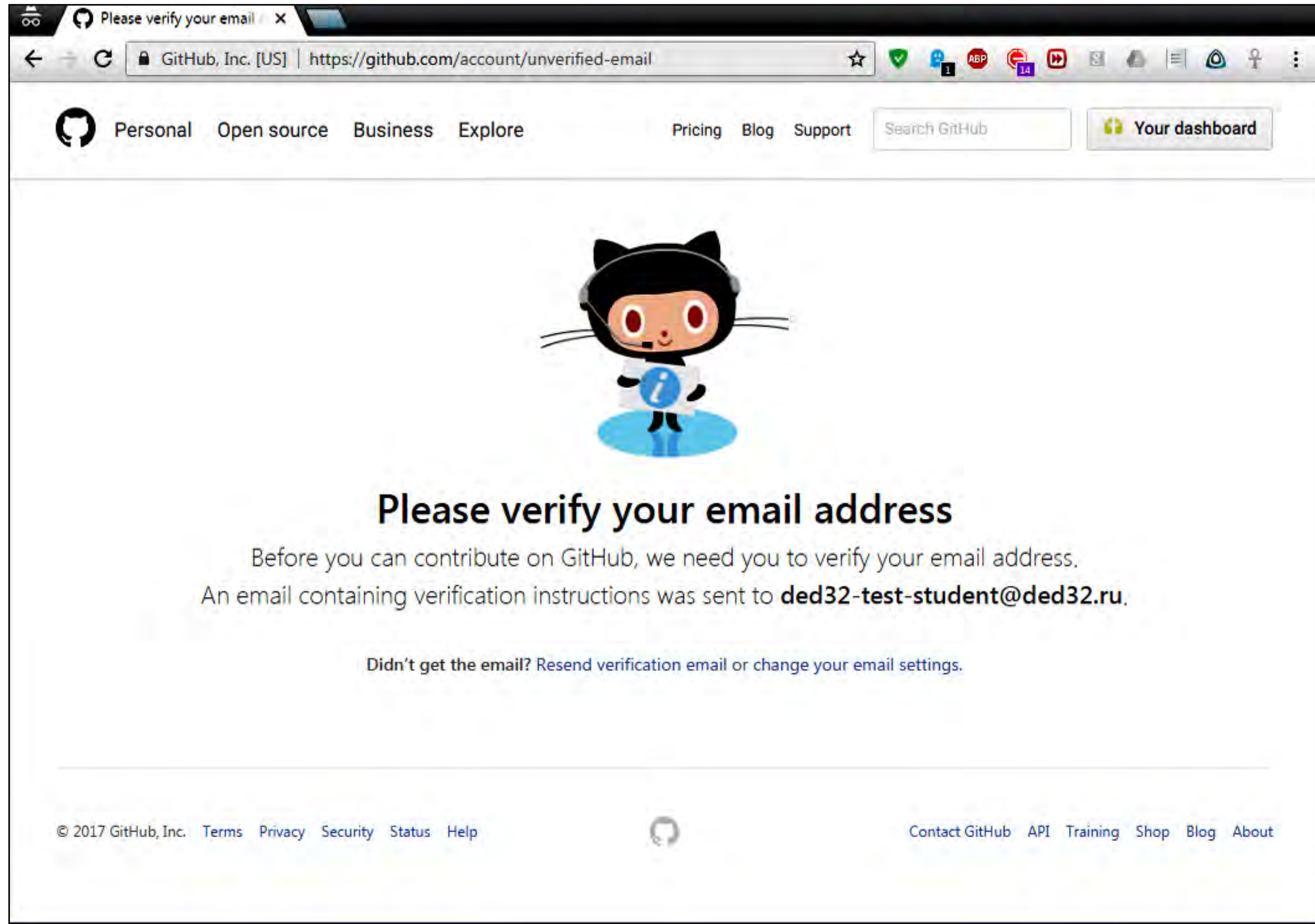
☐ **Help me set up an organization next**
Organizations are separate from personal accounts and are best suited for businesses who need to manage permissions for many employees.
[Learn more about organizations.](#)

Both plans include:

- ✓ Collaborative code review
- ✓ Issue tracking
- ✓ Open source community
- ✓ Unlimited public repositories
- ✓ Join any organization

 **Continue**






Please verify your email

GitHub, Inc. [US] | <https://github.com/account/unverified-email>


Personal Open source Business Explore Pricing Blog Support Search GitHub Your dashboard



Please verify your email address

Before you can contribute on GitHub, we need you to verify your email address.
An email containing verification instructions was sent to **ded32-test-student@ded32.ru**.

Didn't get the email? [Resend verification email or change your email settings.](#)

© 2017 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [API](#) [Training](#) [Shop](#) [Blog](#) [About](#)

Email settings

GitHub, Inc. [US] | https://github.com/settings/emails

Search GitHub

Pull requests Issues Gist

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Saved replies

Authorized applications

Installed integrations

Developer settings

Email

Your primary GitHub email address will be used for account receipts) as well as any web-based GitHub operations (e.g. edits and merges).

ded32-test-student@ded32.ru Primary Private

Add email address

Add

☒ Keep my email address private

We will use ded32-test-student@users.noreply.github.com when performing web-based Git operations and sending email on your behalf. If you want command line Git operations to use your private email you must set your email in Git.

Email preferences

☐ Receive all emails, except those I unsubscribe from.

We'll occasionally contact you with the latest news and happenings from the GitHub Universe. [Learn more.](#)

☒ Only receive account related emails, and those I subscribe to.

We'll only send you legal or administrative emails, and any emails you've specifically subscribed to.

Save email preferences

Signed in as ded32-test-student

Your profile

Your stars

Explore

Integrations


Help

Settings


Sign out

https://github.com/ded32-test-student

or activity notification controls? Check the [Notification center.](#)



[Pull requests](#) [Issues](#) [Gist](#)



ded32-test-student

[Add a bio](#)

Edit profile

⌚

Joined 5 minutes ago

ProTip! Updating your profile with your name, location, and a profile picture helps other GitHub users get to know you.

Edit profile

Overview

Repositories 0

Stars 0

Followers 0

Following 0

Popular repositories

You don't have any public repositories yet.

1 contribution in the last year

Contribution settings

Feb

Mar

Apr

May

Jun

Jul

Aug

Sep

Oct

Nov

Dec

Jan

Mon

Wed

Fri

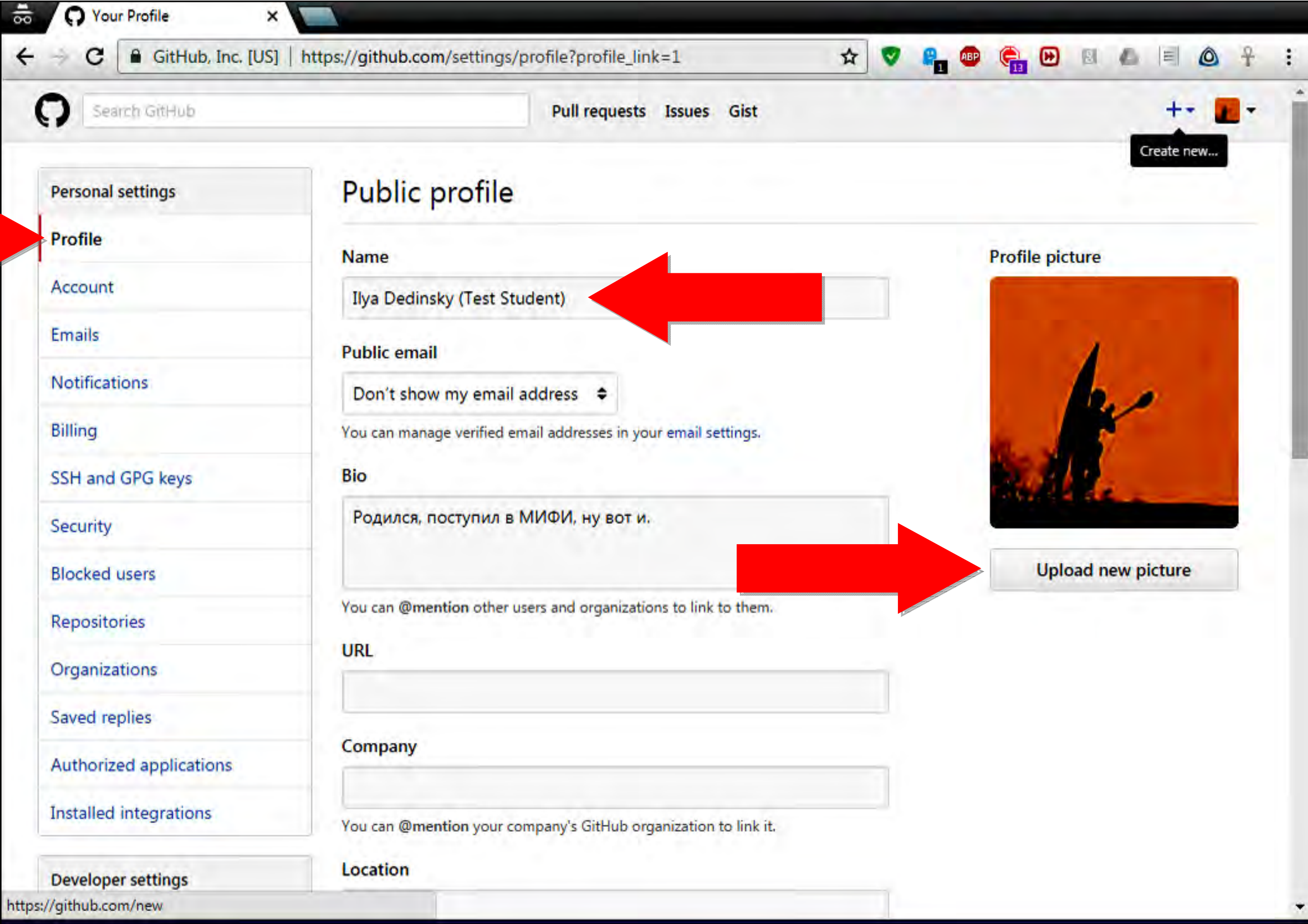
Learn how we count contributions.

Less More

This is your contribution graph. Your first is for joining GitHub and you'll earn more as you make

an contributions. More contributions means a darker green square for that day. Over time, your chart

40



The screenshot shows the GitHub 'Your Profile' settings page. The left sidebar contains a list of settings categories: Personal settings, Profile, Account, Emails, Notifications, Billing, SSH and GPG keys, Security, Blocked users, Repositories, Organizations, Saved replies, Authorized applications, and Installed integrations. A red arrow points to the 'Profile' tab. The main content area is titled 'Public profile' and contains several form fields: Name (Ilya Dedinsky (Test Student)), Public email (Don't show my email address), Bio (Родился, поступил в МИФИ, ну вот и.), URL, Company, and Location. A red arrow points to the 'Name' field. To the right of the bio field is a 'Profile picture' section with a placeholder image and an 'Upload new picture' button. A red arrow points to this button. The browser's address bar shows the URL 'https://github.com/settings/profile?profile_link=1'.

Personal settings

Profile

Account

Emails

Notifications

Billing

SSH and GPG keys

Security

Blocked users

Repositories

Organizations

Saved replies

Authorized applications

Installed integrations

Developer settings

Public profile

Name

Ilya Dedinsky (Test Student)

Public email

Don't show my email address

You can manage verified email addresses in your [email settings](#).

Bio

Родился, поступил в МИФИ, ну вот и.

You can @mention other users and organizations to link to them.


URL

Company

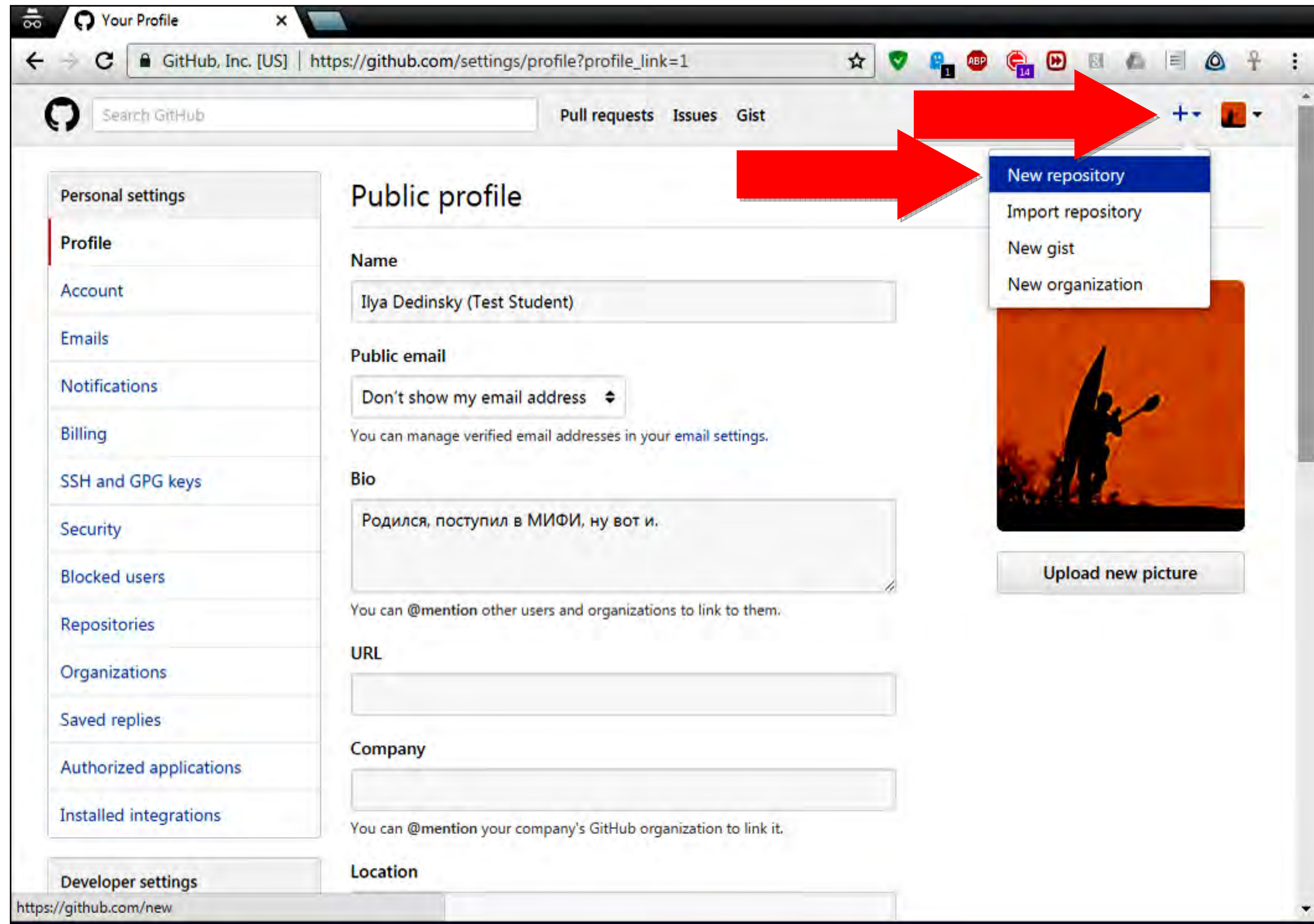
You can @mention your company's GitHub organization to link it.

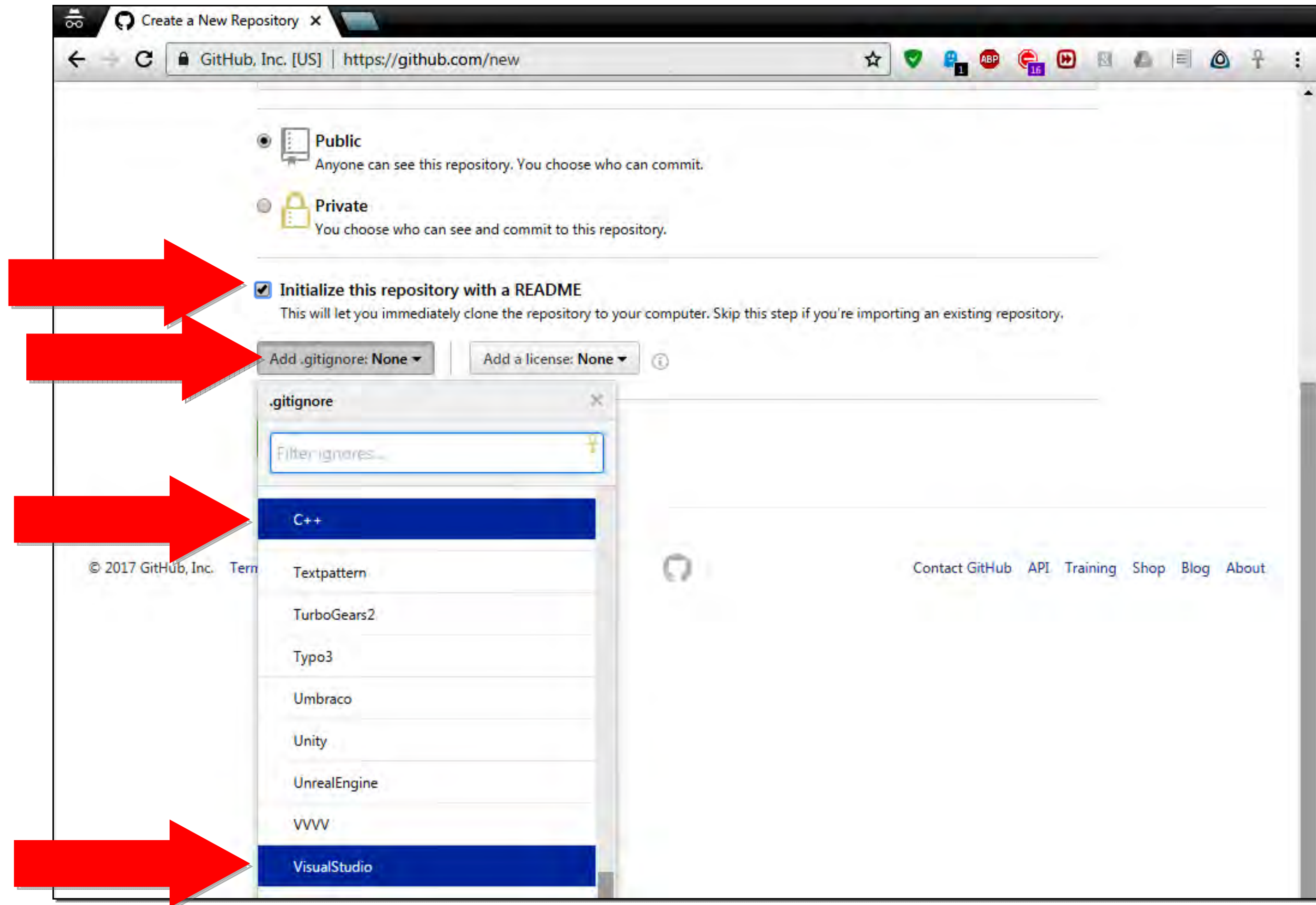
Location

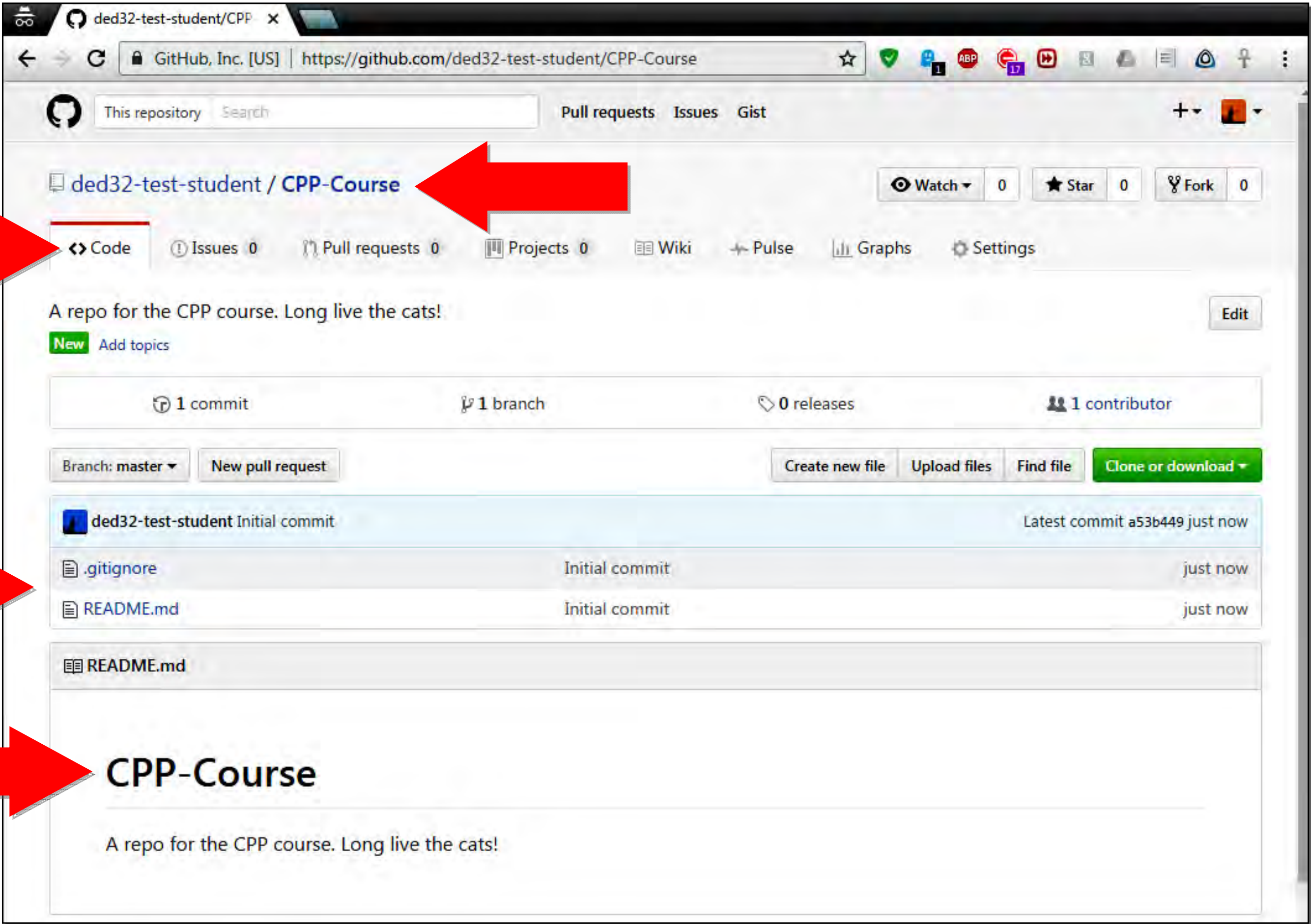
Profile picture



[Upload new picture](#)







The screenshot shows a web browser window displaying a GitHub repository page. The browser's address bar shows the URL `https://github.com/ded32-test-student/_CPP-Course`. The repository page header includes the repository name `ded32-test-student / CPP-Course`, which is pointed to by a red arrow. Below the header, there are tabs for `Code`, `Issues 0`, `Pull requests 0`, `Projects 0`, `Wiki`, `Pulse`, `Graphs`, and `Settings`. The `Code` tab is selected and pointed to by a red arrow. The repository description is `A repo for the CPP course. Long live the cats!`. Below this, there are statistics: `1 commit`, `1 branch`, `0 releases`, and `1 contributor`. A row of buttons includes `Branch: master`, `New pull request`, `Create new file`, `Upload files`, `Find file`, and `Clone or download`. The commit history shows `ded32-test-student Initial commit` with the latest commit `a53b449` just now. Below the commit history, there is a list of files: `.gitignore` and `README.md`, both with `Initial commit` status. At the bottom, the repository title `CPP-Course` is displayed, pointed to by a red arrow, followed by the same description: `A repo for the CPP course. Long live the cats!`.

Репозиторий: Добавляем соавтора и ~~большого брата~~ ментора

The screenshot shows the GitHub interface for the repository `ded32-test-student / CPP-Course`. The browser address bar shows the URL `https://github.com/ded32-test-student/ CPP-Course/settings/collabor`. The repository page includes tabs for `Code`, `Issues 0`, `Pull requests 0`, `Projects 0`, `Wiki`, `Pulse`, `Graphs`, and `Settings`. The `Settings` tab is selected and highlighted with a red arrow. On the left sidebar, the `Collaborators` option is also highlighted with a red arrow. The main content area is titled `Collaborators` and contains the text: "This repository doesn't have any collaborators yet. Use the form below to add a collaborator." Below this, there is a search bar with the placeholder text "Search by username, full name or email address" and a note: "You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead." The search bar contains the text `ded32`, and a dropdown list shows search results: `ded32 Ilya Dedinsky` (highlighted), `ded32-test-student Ilya Dedinsky (Test Student)`, `ded32-tutor`, and `ded32-student`. To the right of the search bar is an `Add collaborator` button, which is also highlighted with a red arrow.

Collaborators

GitHub, Inc. [US] | https://github.com/ded32-test-student/PHP-Course/settings/collaborator

This repository Search

Pull requests Issues Gist

ded32-test-student / PHP-Course

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Options

Collaborators

Branches


Webhooks

Integrations & services

Deploy keys

Collaborators

Push access to the repository



Ilya Dedinsky

Awaiting ded32's response

Copy invite link Cancel invite

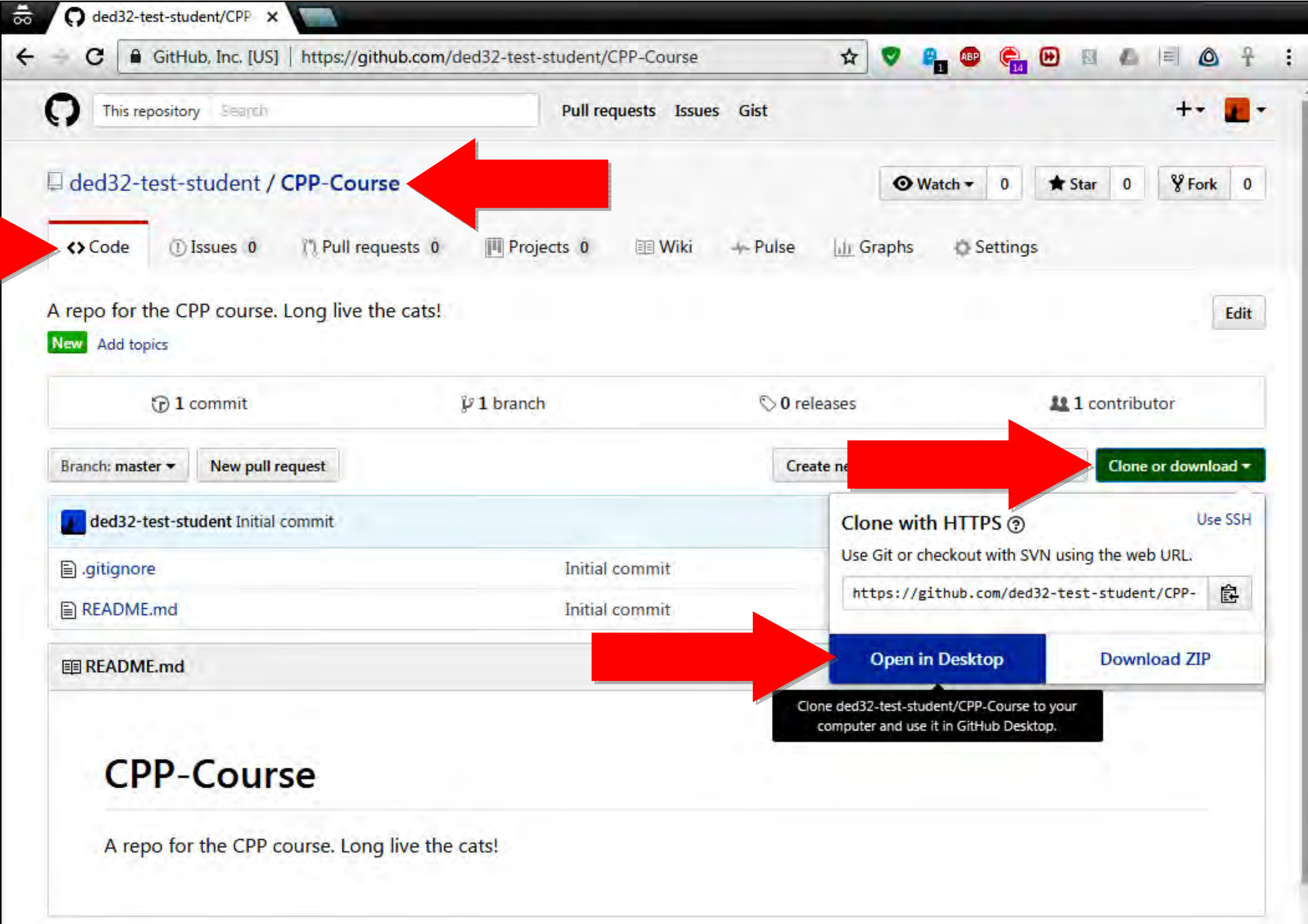
Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

Add collaborator

© 2017 GitHub, Inc. Terms Privacy Security Status Help

Contact GitHub API Training Shop Blog About



The screenshot shows the GitHub web interface for the repository `ded32-test-student / CPP-Course`. The browser address bar displays `https://github.com/ded32-test-student/_CPP-Course`. The repository name is highlighted with a red arrow. Below the repository name, the `Code` tab is selected, also indicated by a red arrow. The repository description is "A repo for the CPP course. Long live the cats!". The repository statistics show 1 commit, 1 branch, 0 releases, and 1 contributor. The `Clone or download` button is highlighted with a red arrow. A dropdown menu is open, showing the option to `Clone with HTTPS`. The `Open in Desktop` button is highlighted with a red arrow. A black tooltip box at the bottom right contains the text: "Clone ded32-test-student/ CPP-Course to your computer and use it in GitHub Desktop."

ded32-test-student / CPP-Course

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

A repo for the CPP course. Long live the cats!

New Add topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request Create new branch

ded32-test-student Initial commit

.gitignore Initial commit

README.md Initial commit

README.md

Clone with HTTPS ? Use SSH

Use Git or checkout with SVN using the web URL.

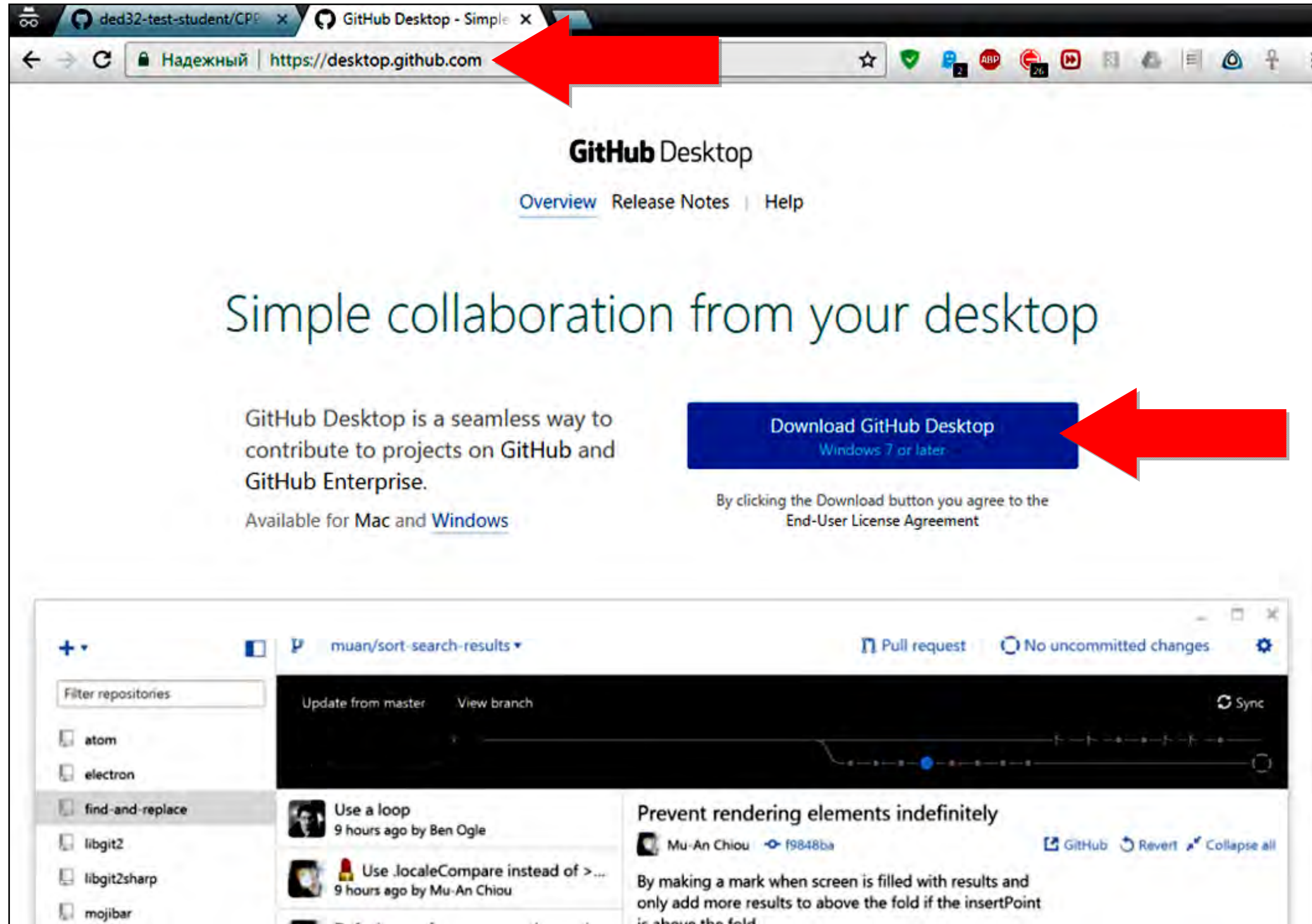
`https://github.com/ded32-test-student/ CPP-`

Open in Desktop Download ZIP

Clone ded32-test-student/ CPP-Course to your computer and use it in GitHub Desktop.

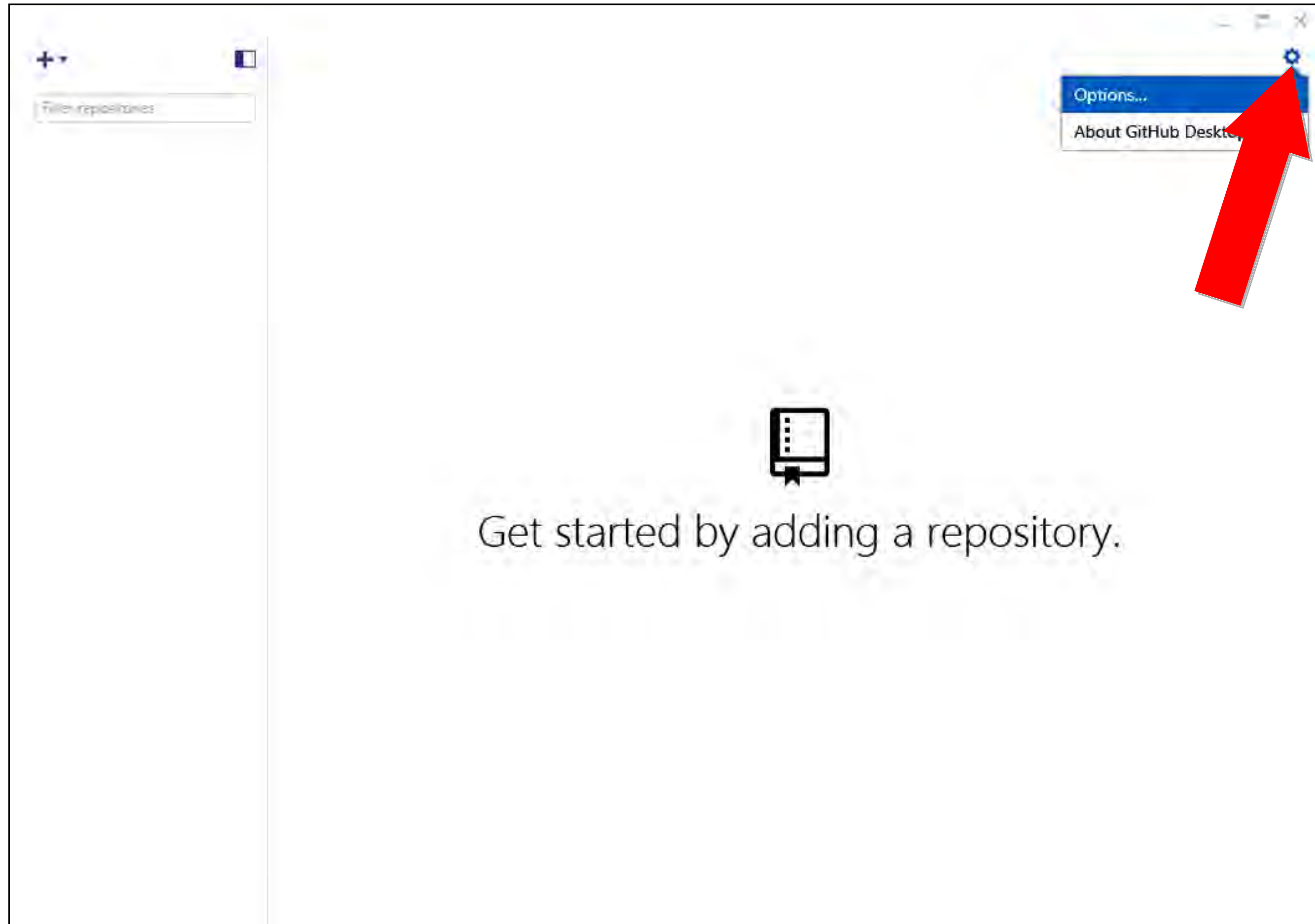
CPP-Course

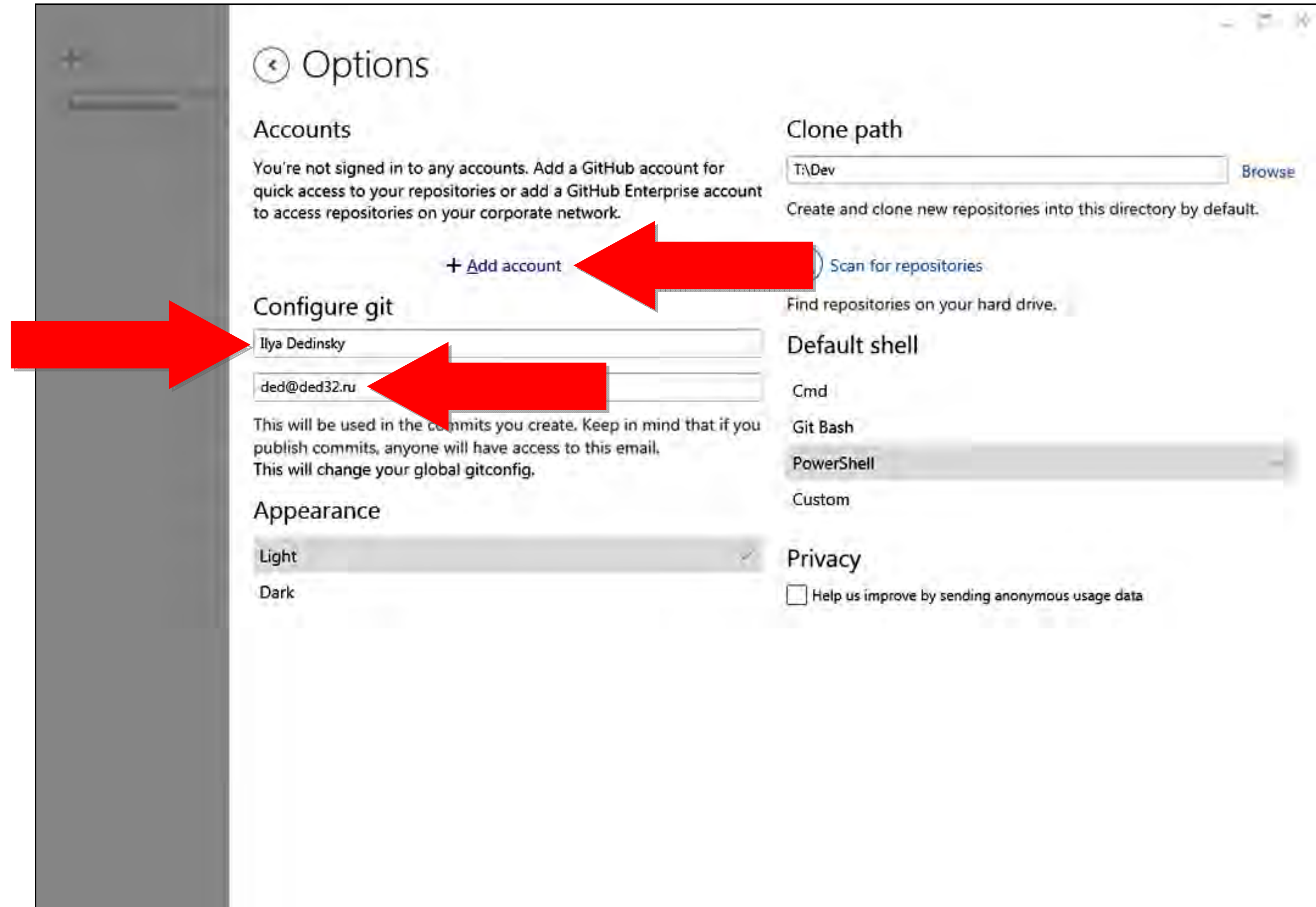
A repo for the CPP course. Long live the cats!

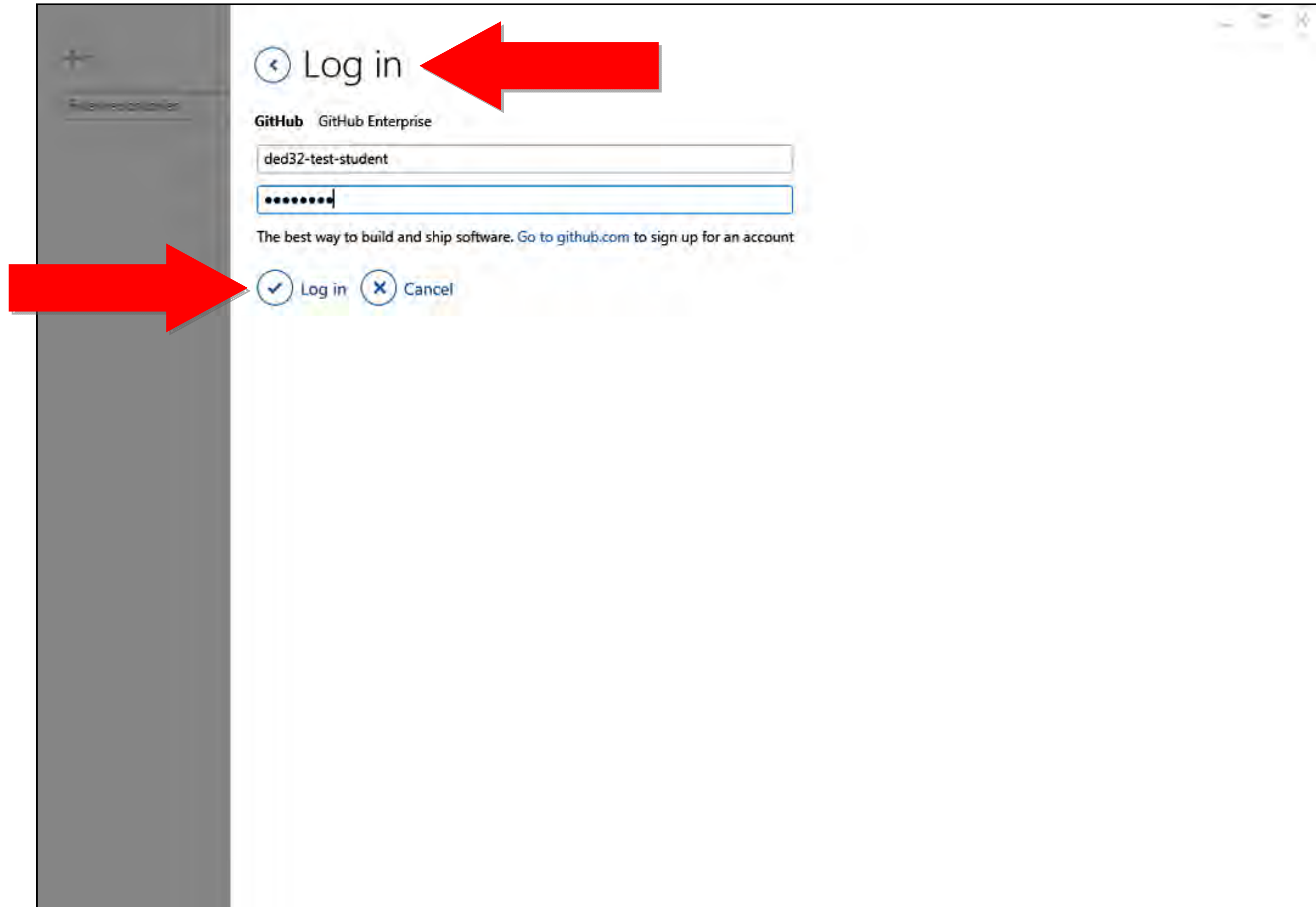


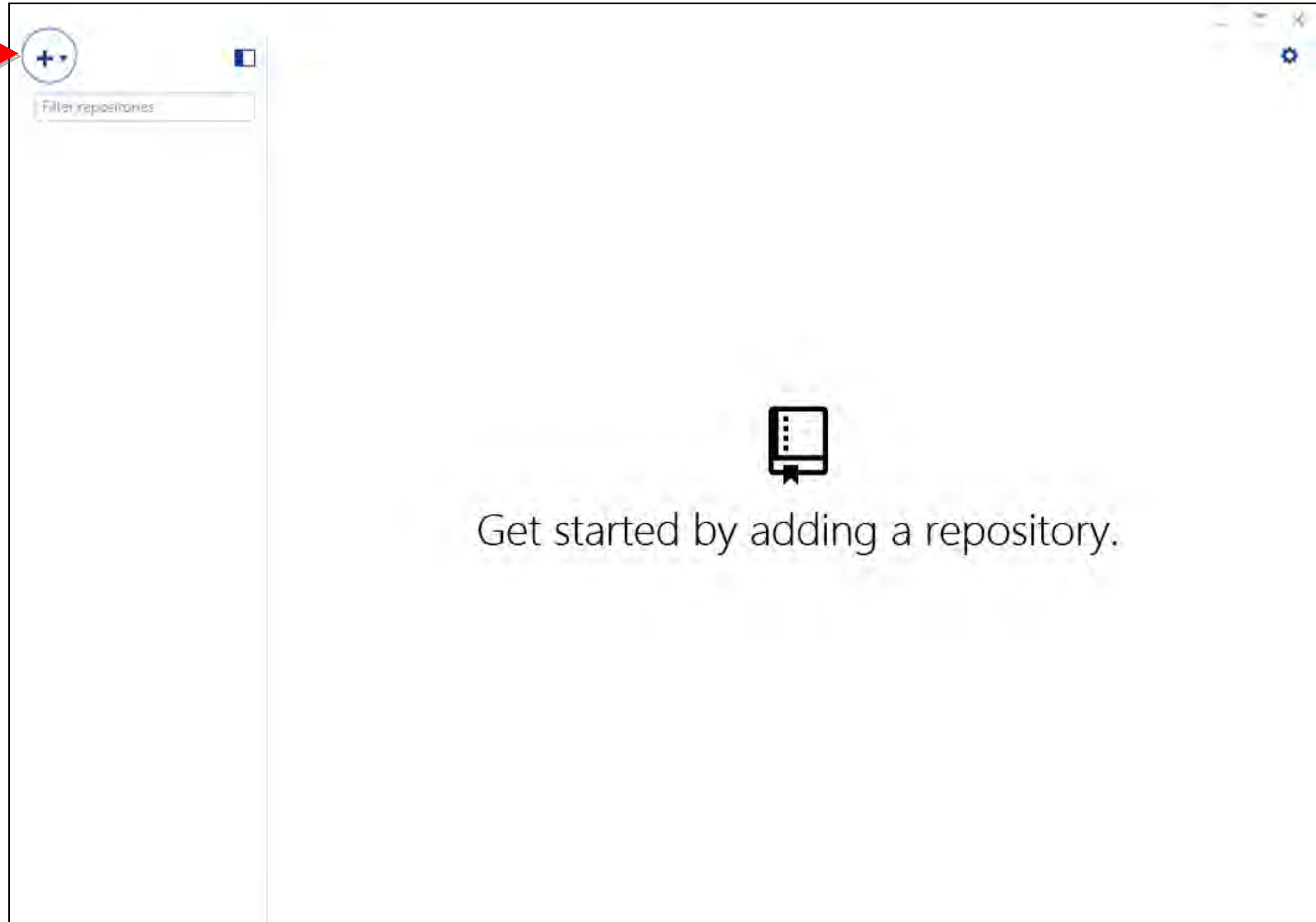
The image shows a browser window with the GitHub Desktop website. The address bar shows the URL <https://desktop.github.com>. The page title is "GitHub Desktop". The main heading is "Simple collaboration from your desktop". Below this, there is a description: "GitHub Desktop is a seamless way to contribute to projects on GitHub and GitHub Enterprise. Available for Mac and [Windows](#)". A large blue button labeled "Download GitHub Desktop" is prominently displayed, with a red arrow pointing to it. Below the button, it says "By clicking the Download button you agree to the End-User License Agreement".

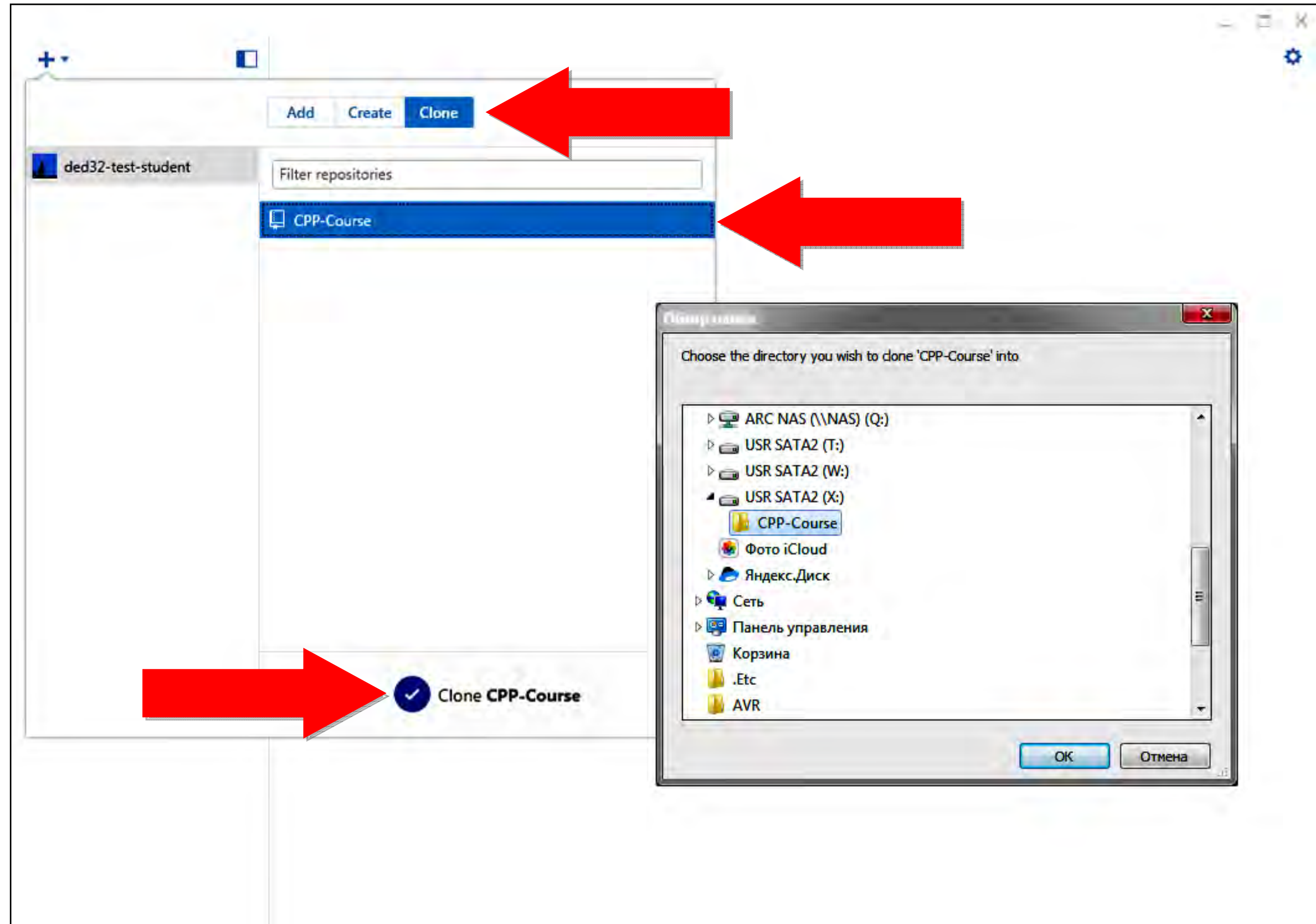
Below the website screenshot, there is a screenshot of the GitHub Desktop application interface. The interface shows a list of repositories on the left, including "atom", "electron", "find-and-replace", "libgit2", "libgit2sharp", and "mojibar". The main area displays a commit history for the "muan/sort-search-results" repository, showing commits by Ben Ogle and Mu-An Chiou. The interface also includes a "Pull request" button and a "No uncommitted changes" status.

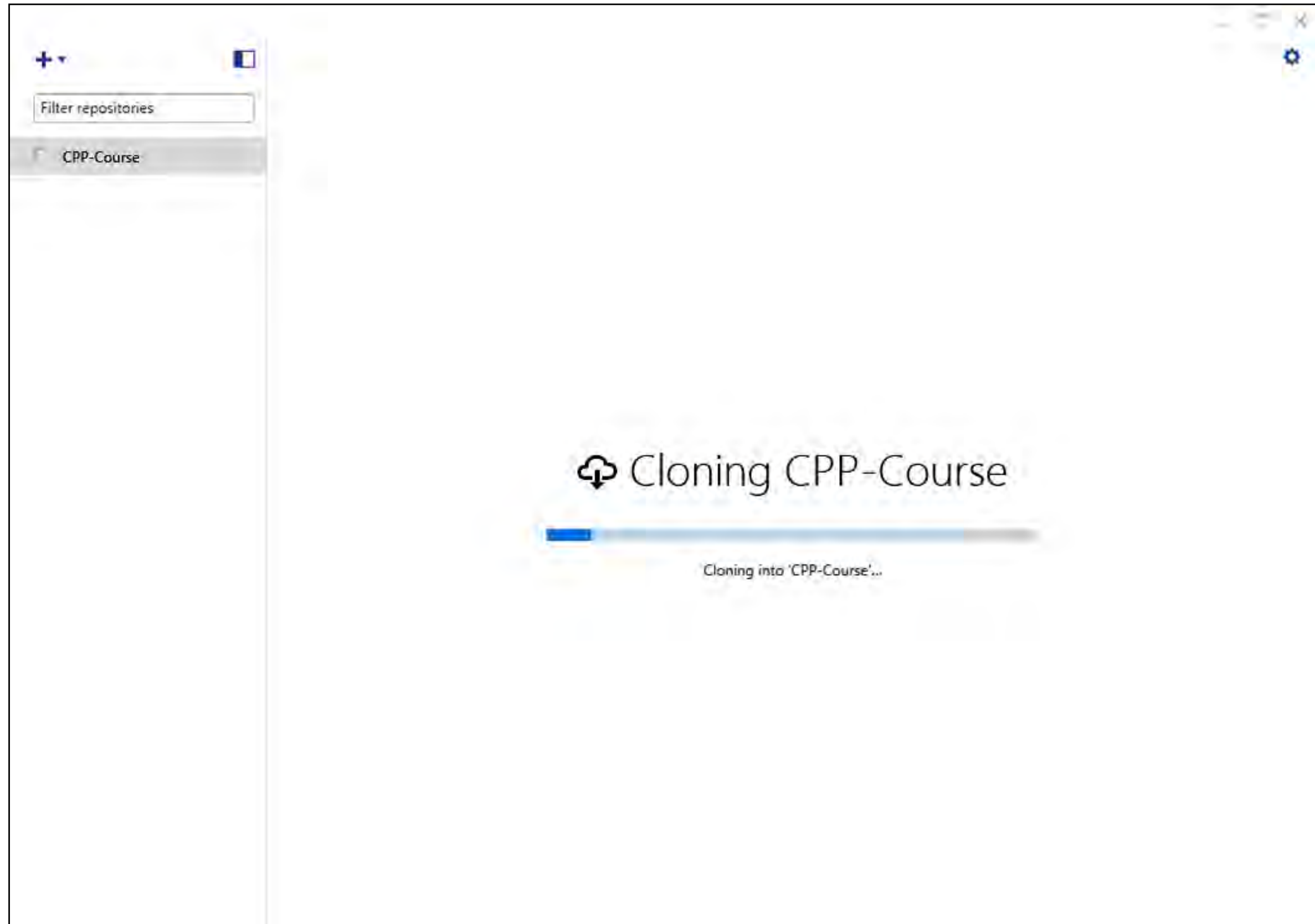


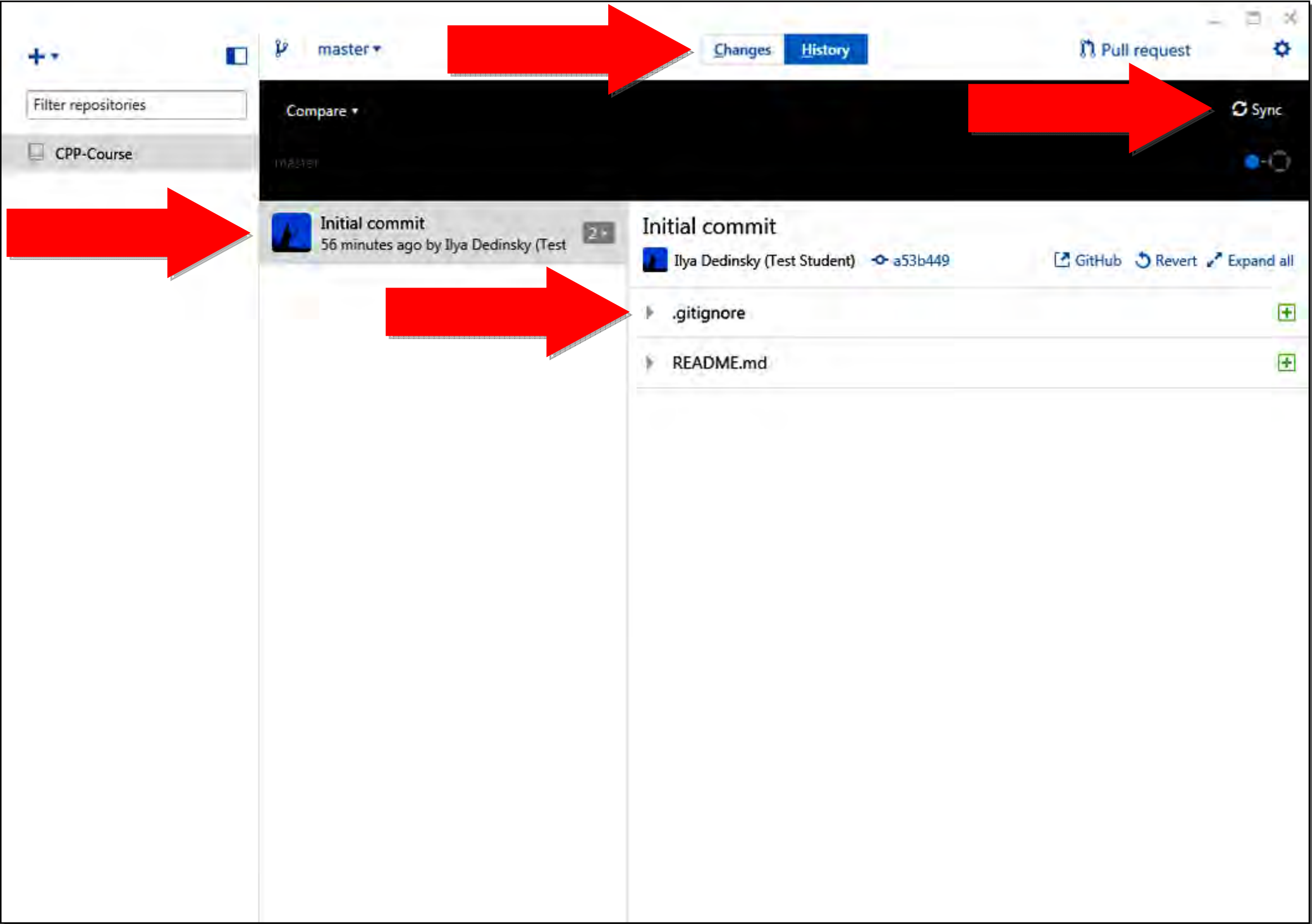














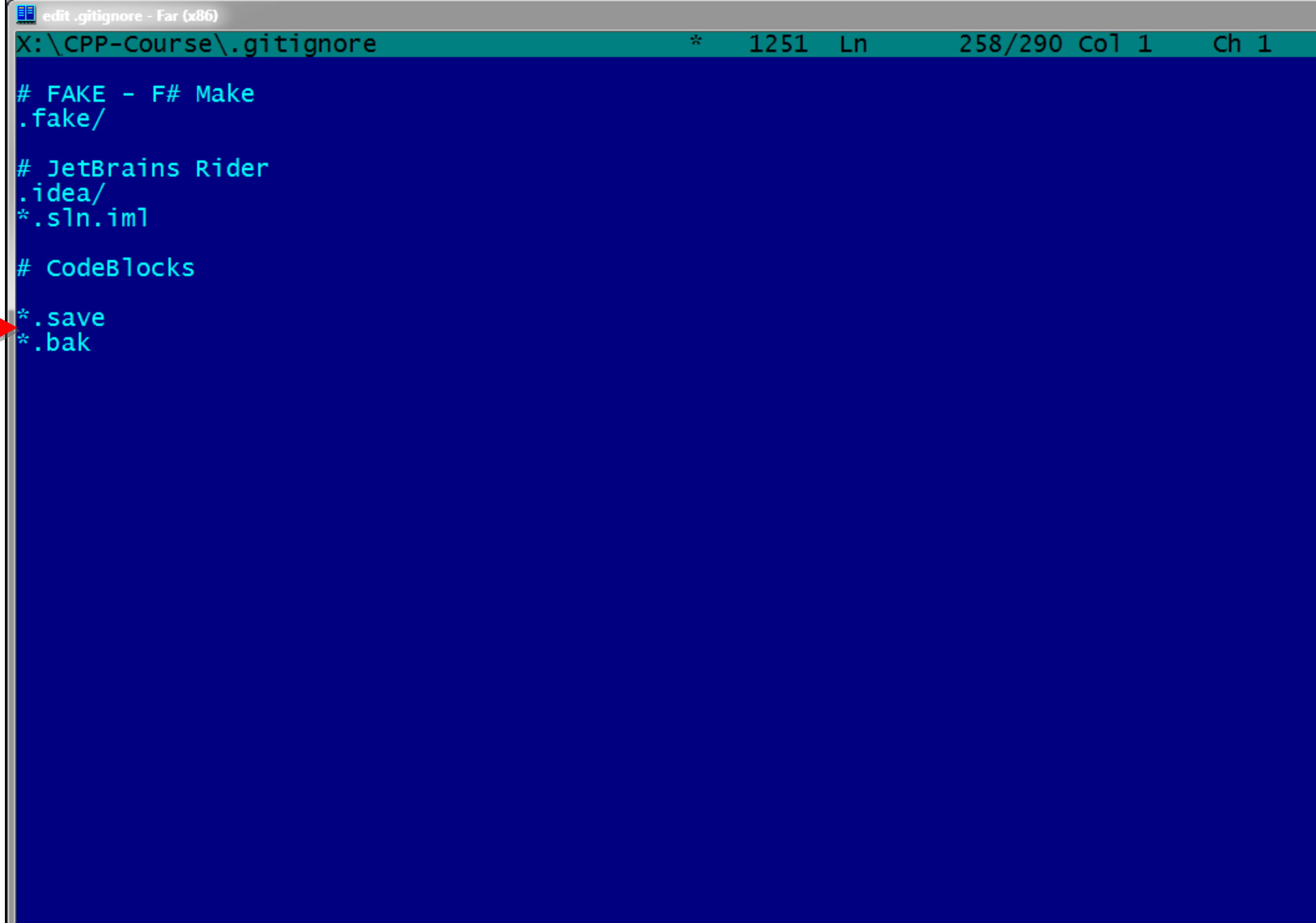
X:\CPP-Course

Name	Name
..	..
.git	\1 - one file
.gitignore	\2 - many fil
README.md	\3 - init rep
	\4 - start ta
	\5 - solving
	\6 - solving
	\7 - finishir
	\8 - merging
	\bin

..

Up 08.02.17 15:15

\bin

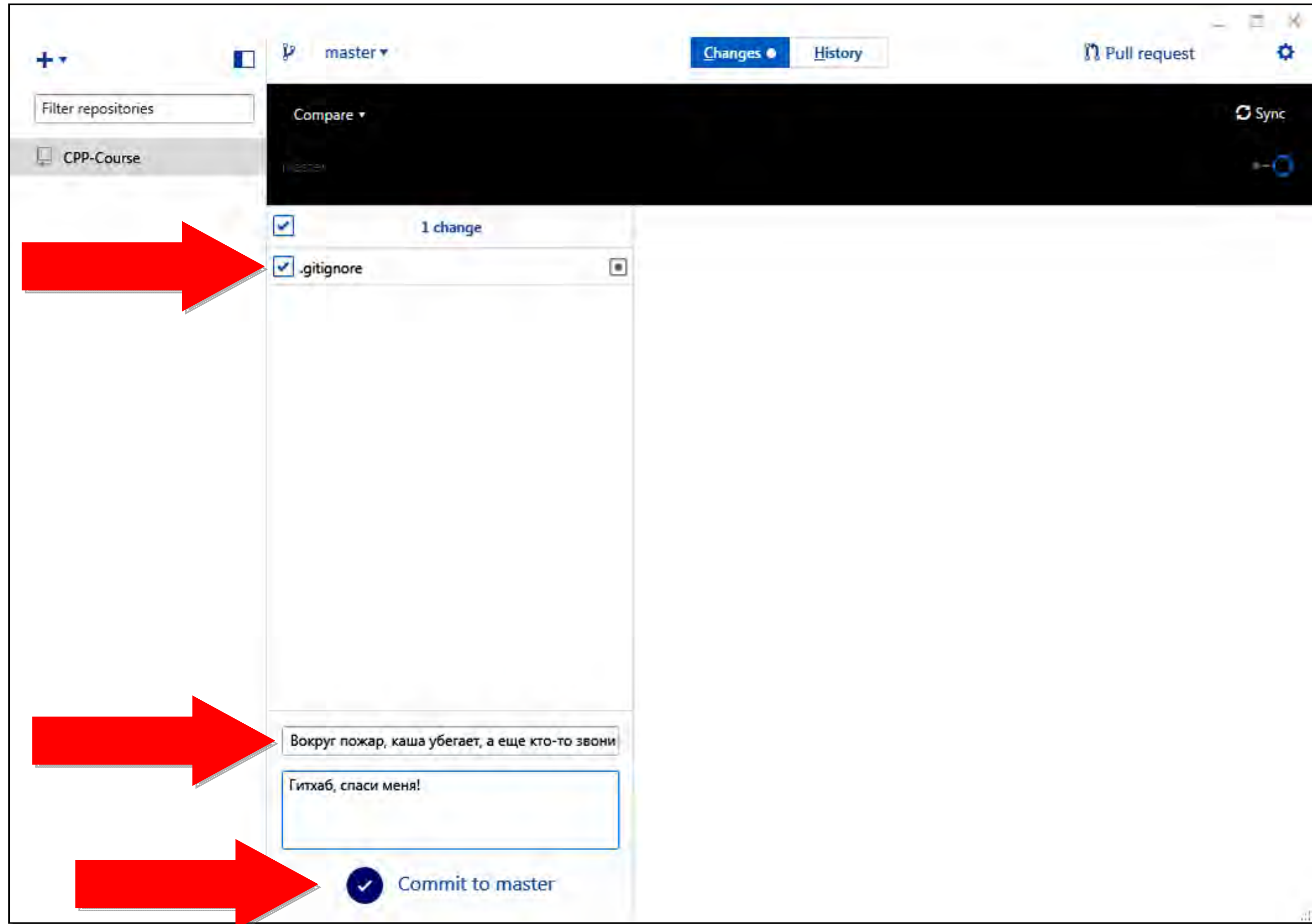


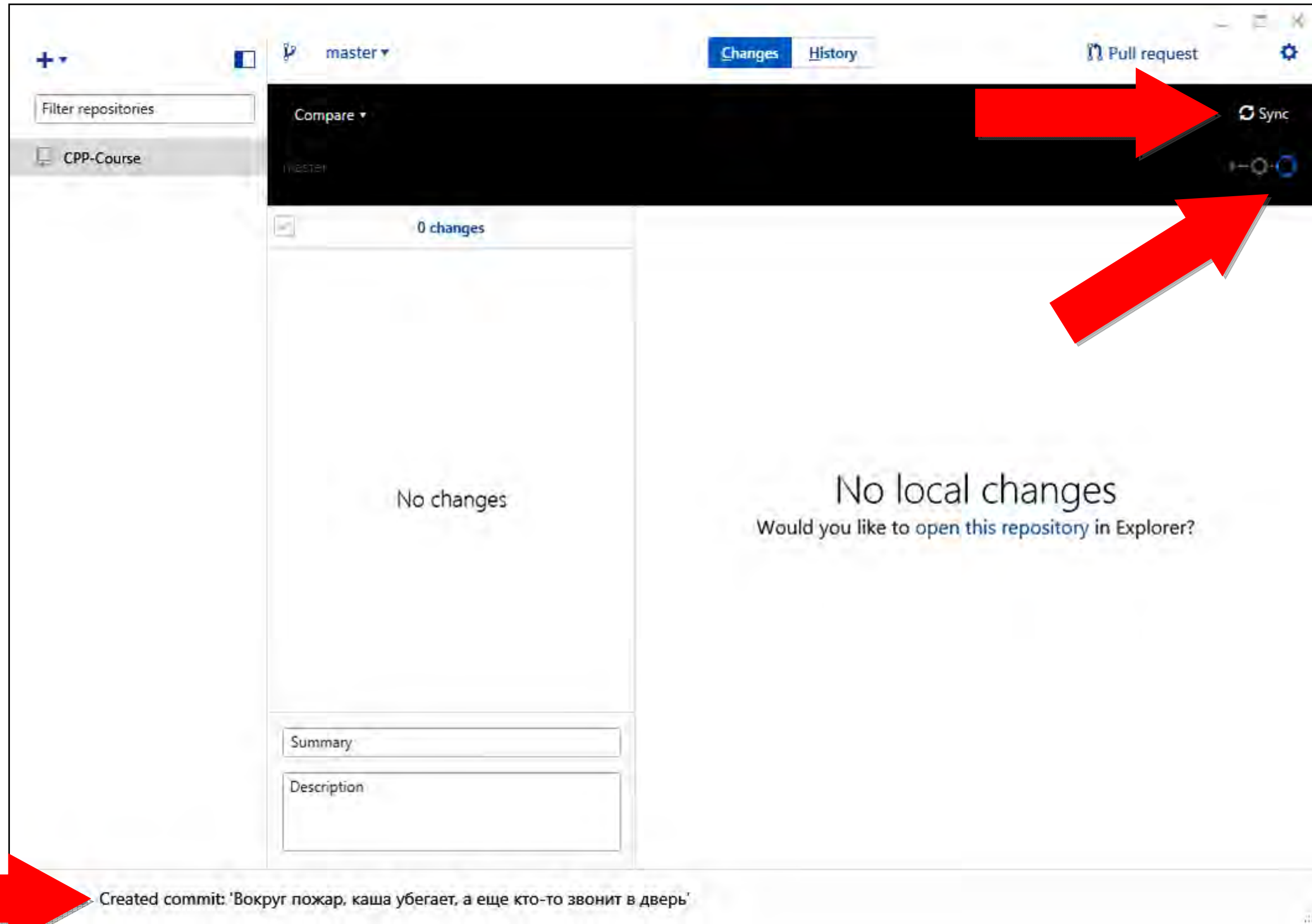
```
edit .gitignore - Far (x86)
X:\CPP-Course\.gitignore * 1251 Ln 258/290 Col 1 Ch 1

# FAKE - F# Make
.fake/

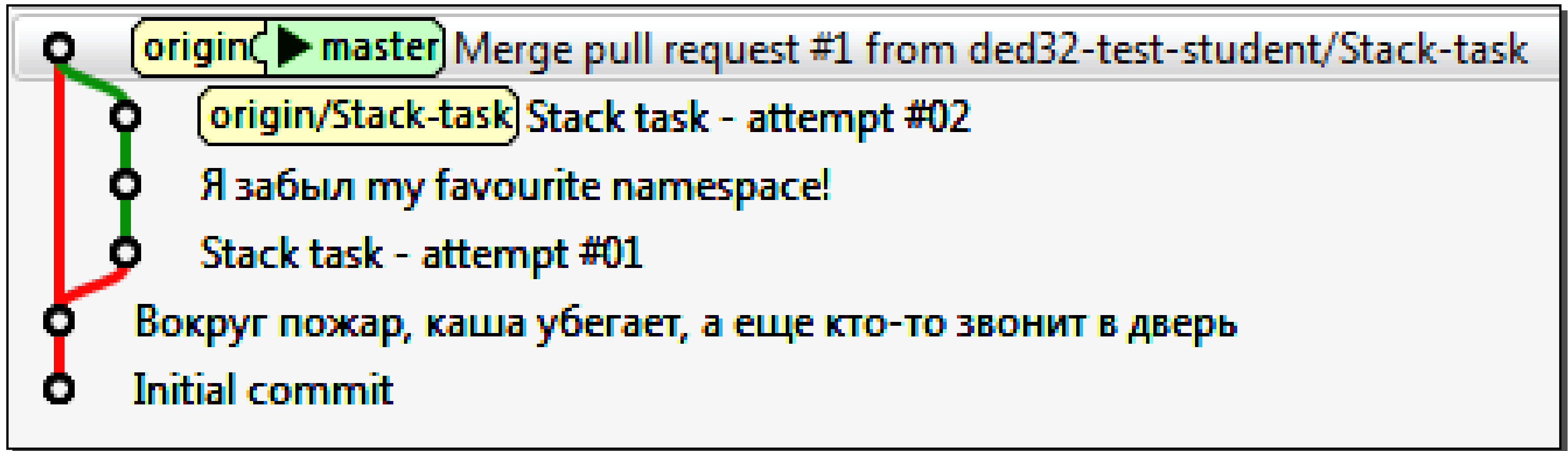
# JetBrains Rider
.idea/
*.sln.iml

# CodeBlocks
*.save
*.bak
```

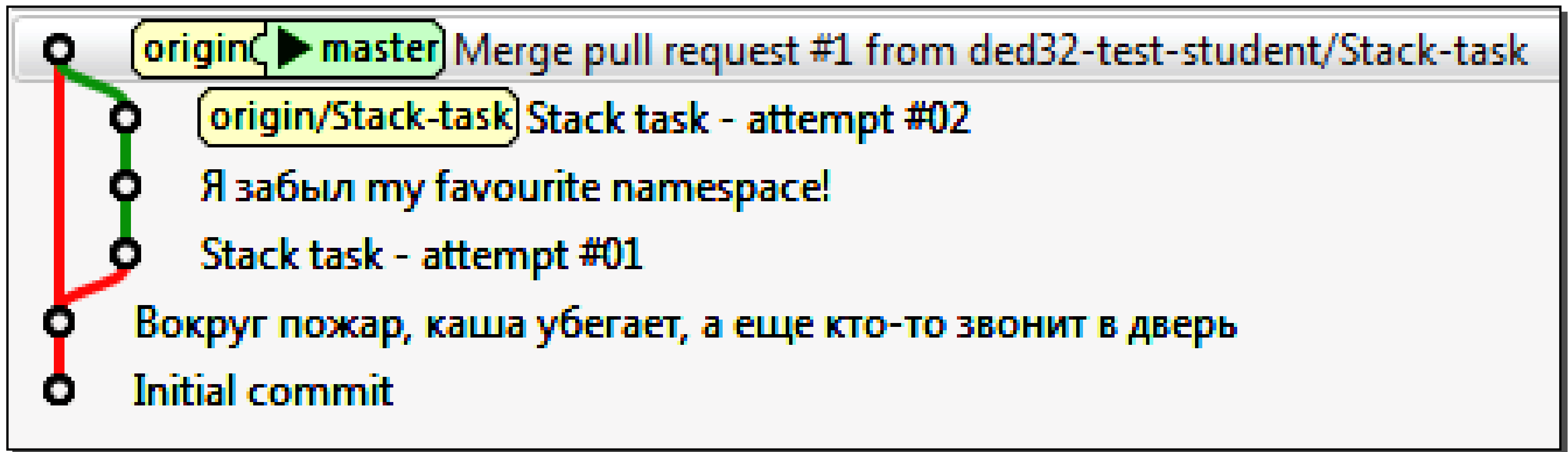


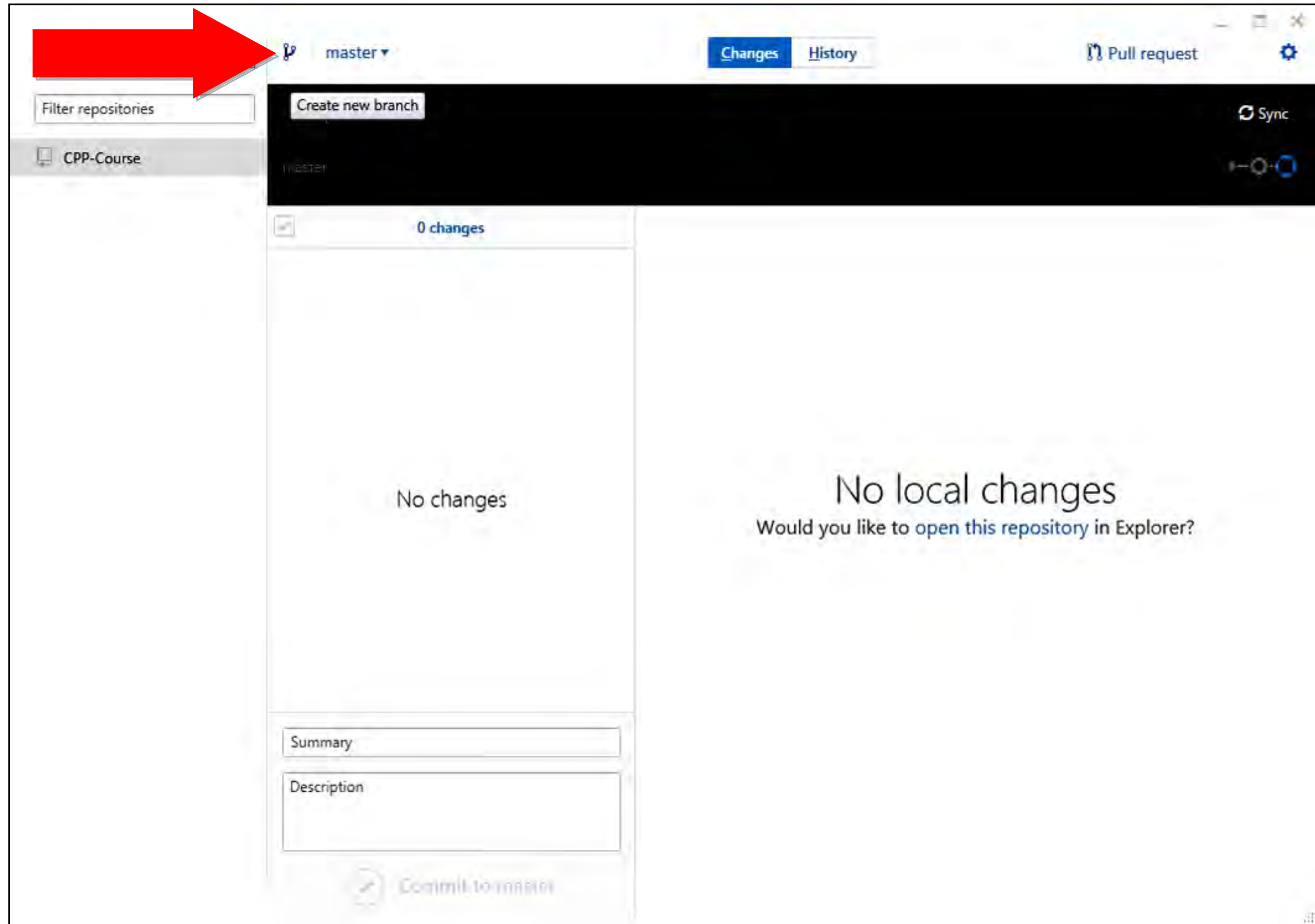


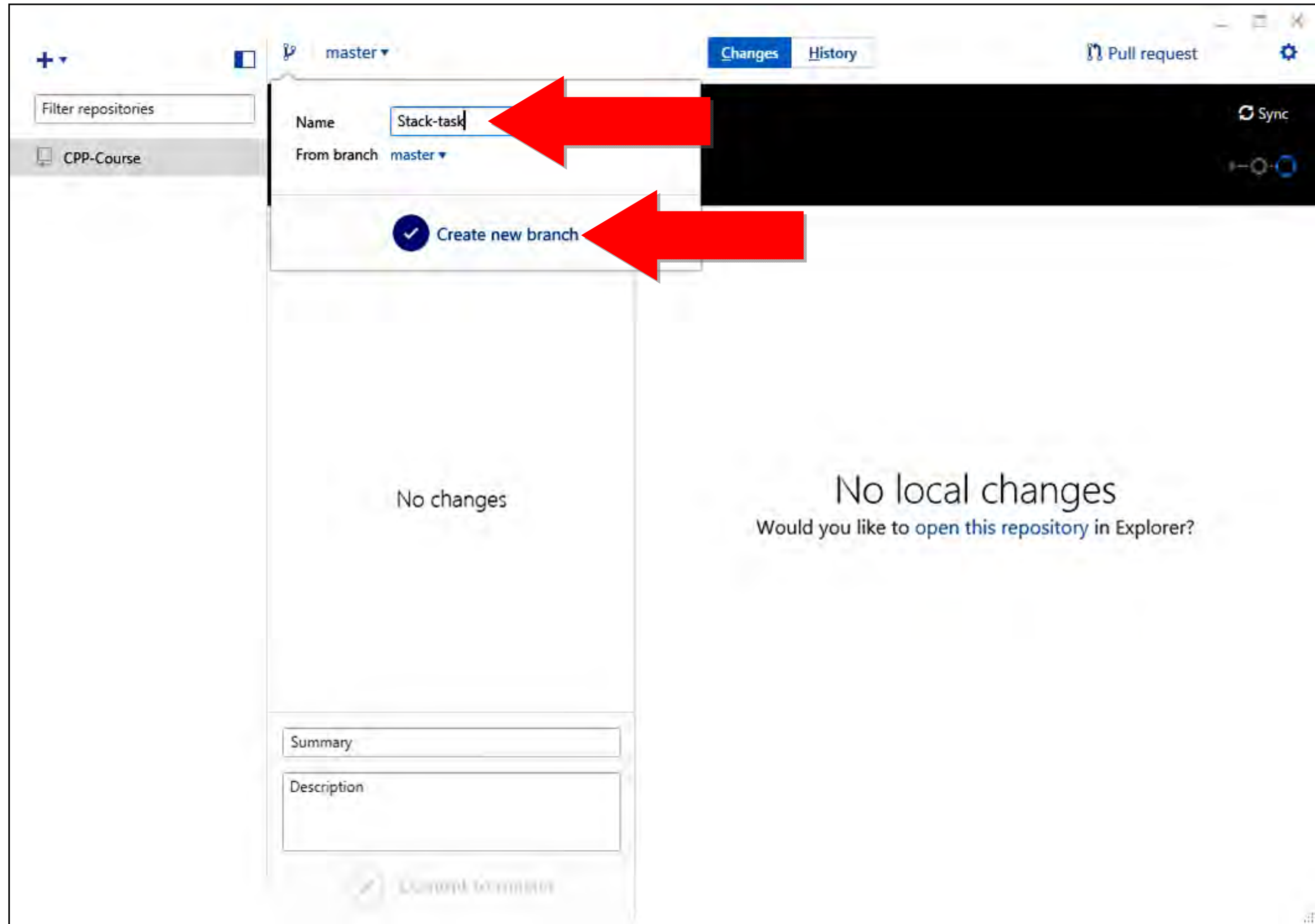
- Master – главная ветка, в ней все всегда отлажено и работает :)
- Stack-task – ветка текущего задания (чтобы не портить Master)
- Origin – обобщенное имя для репозитория на сервере

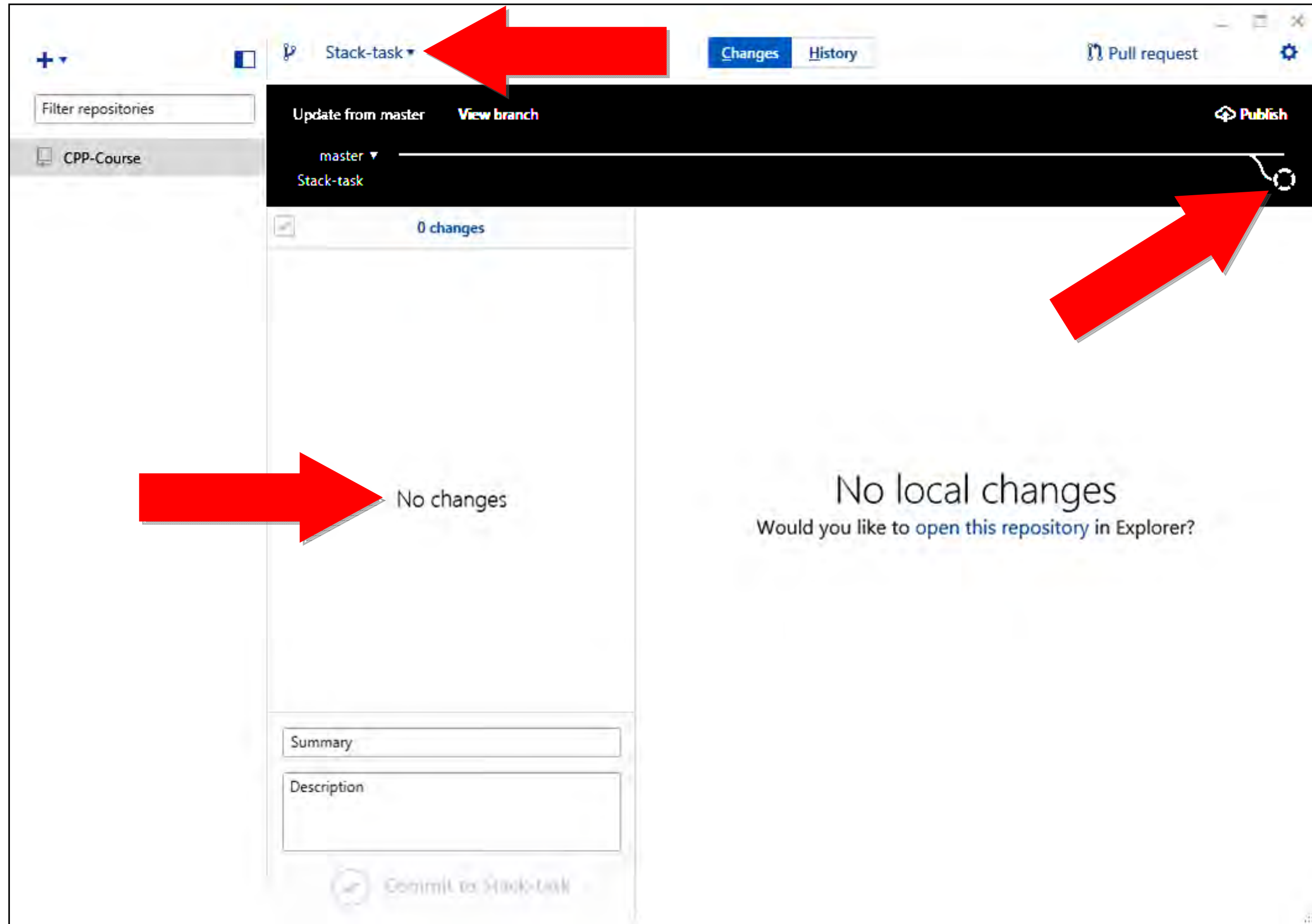


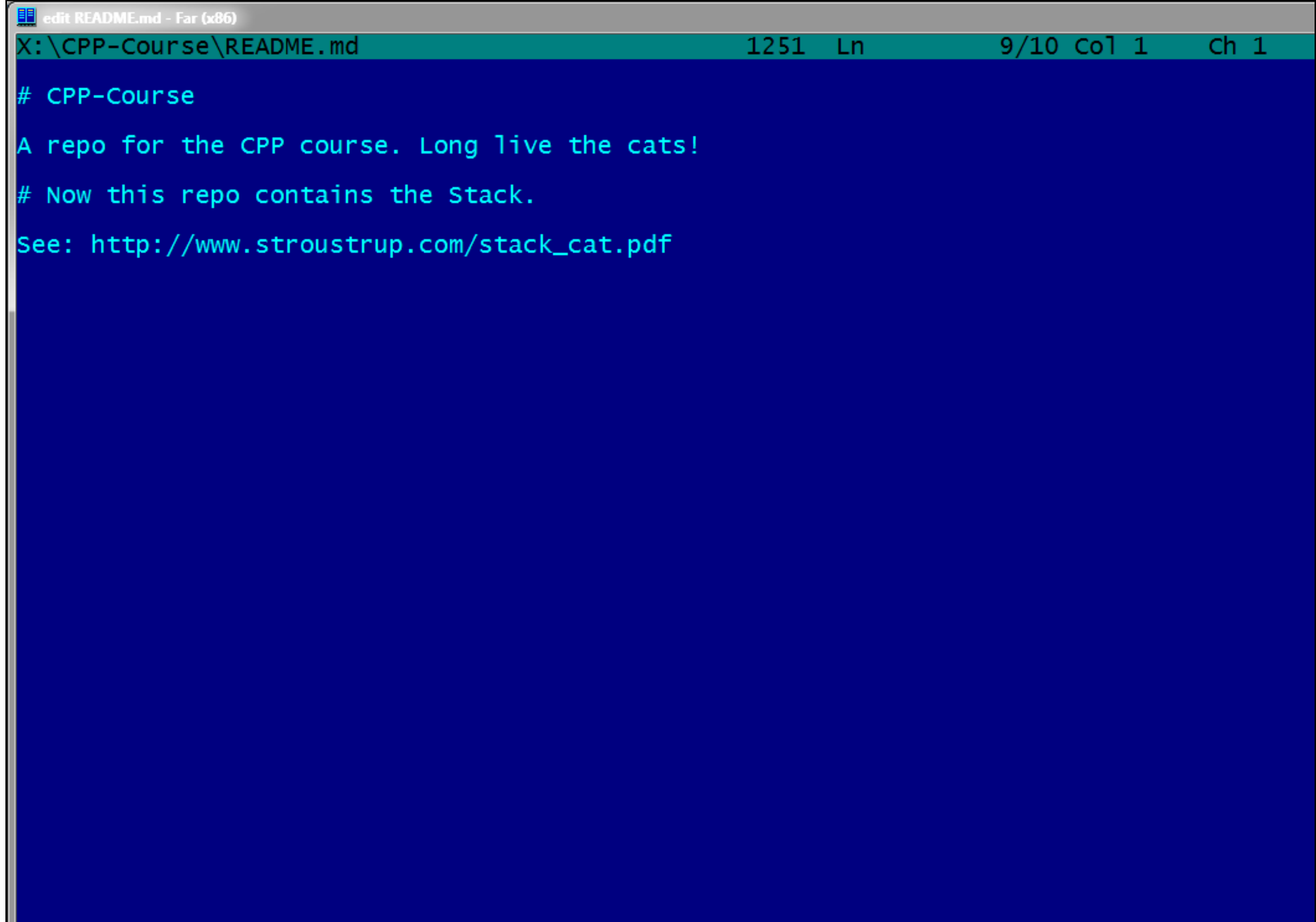
- ~~В любой непонятной ситуации~~ Получив задание (запланировав фичу, обнаружив баг) → делай новую ветку
- Пока не уверен → работай в созданной ветке
- Сделал → отправь pull request ментору
- Работа одобрена → слей (merge) ветку с Master











```
edit README.md - Far (x86)
X:\CPP-Course\README.md 1251 Ln 9/10 Col 1 Ch 1

# CPP-Course

A repo for the CPP course. Long live the cats!

# Now this repo contains the Stack.

See: http://www.stroustrup.com/stack\_cat.pdf
```

```
#include <stdio.h>

//=====

#define ASSERT_OK()  if (!ok()) { dump(); assert (!"Stack Error"); }

//=====

class CStack
{
public:

    bool ok()    const;
    bool dump() const;

    bool push (int value);
    int  pop();

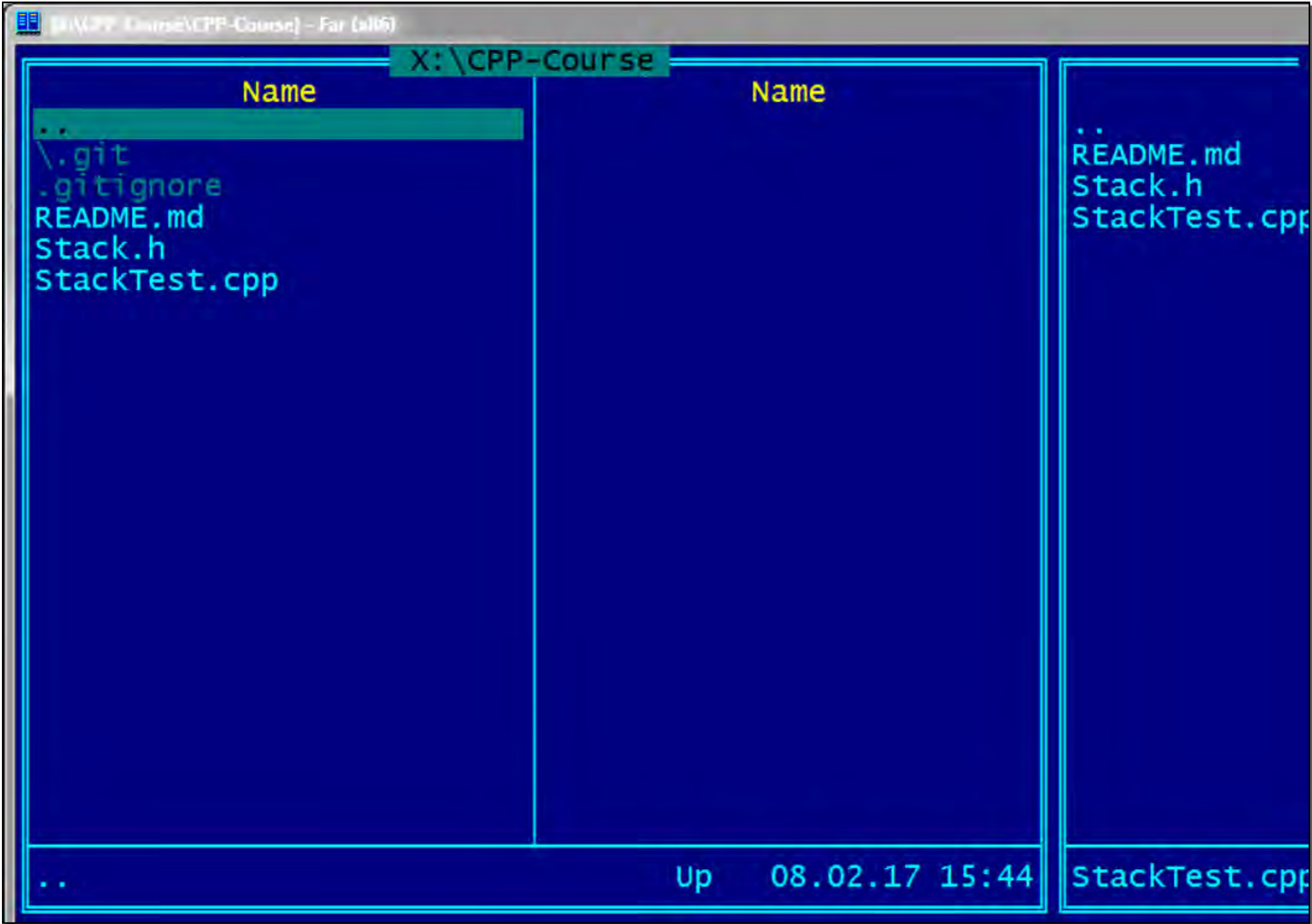
    int  clear();
    int  size() const;

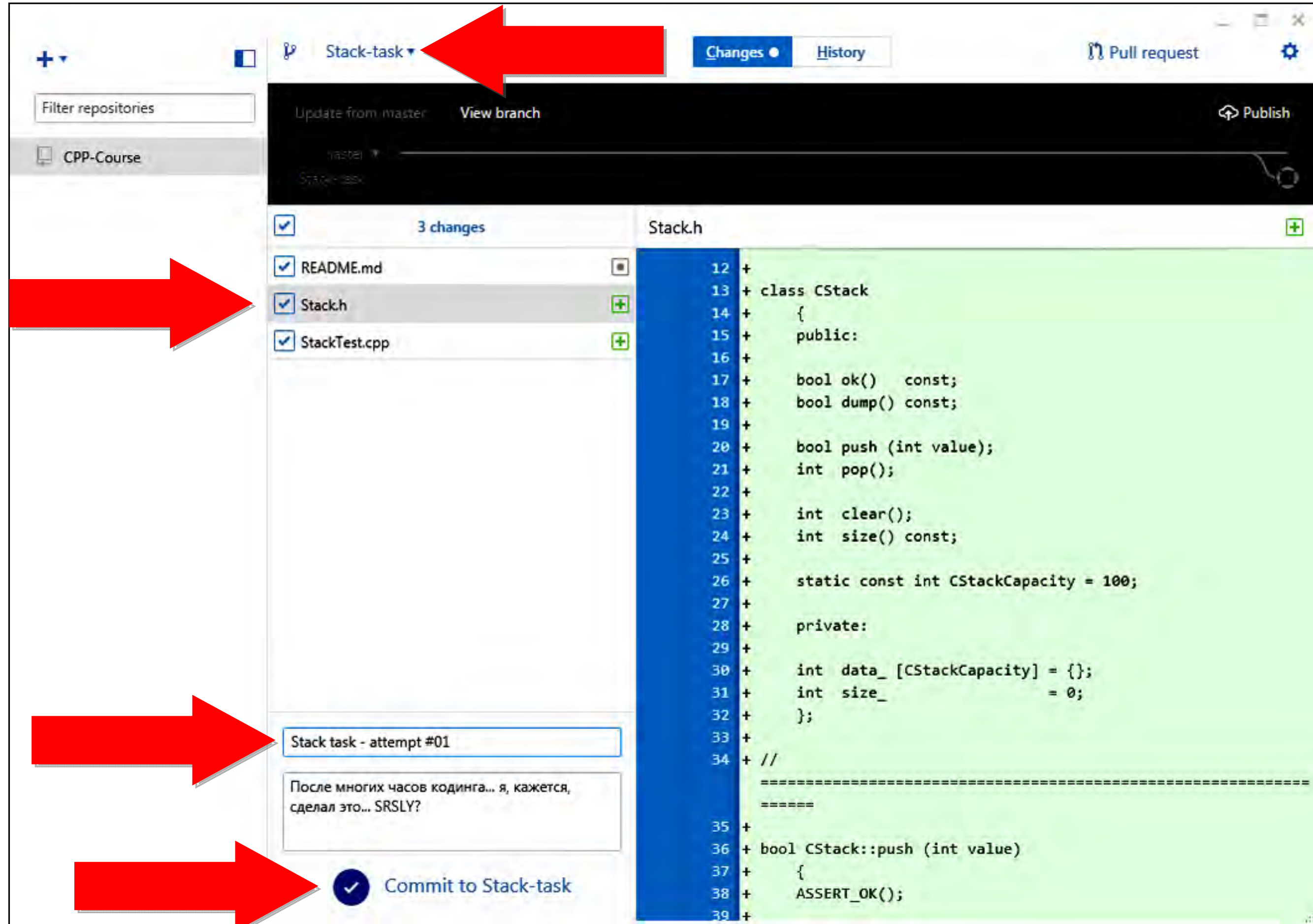
    static const int CStackCapacity = 100;

private:

    int  data_ [CStackCapacity] = {};
    int  size_           = 0;
};

//=====
```





The screenshot shows the GitHub Desktop interface for a repository named "CPP-Course". The current branch is "Stack-task". The interface displays three changes: README.md, Stack.h, and StackTest.cpp. The commit message field contains the text "Stack task - attempt #01" and "После многих часов кодинга... я, кажется, сделал это... SRSLY?". The "Commit to Stack-task" button is visible at the bottom.

Stack-task

Update from master View branch

3 changes

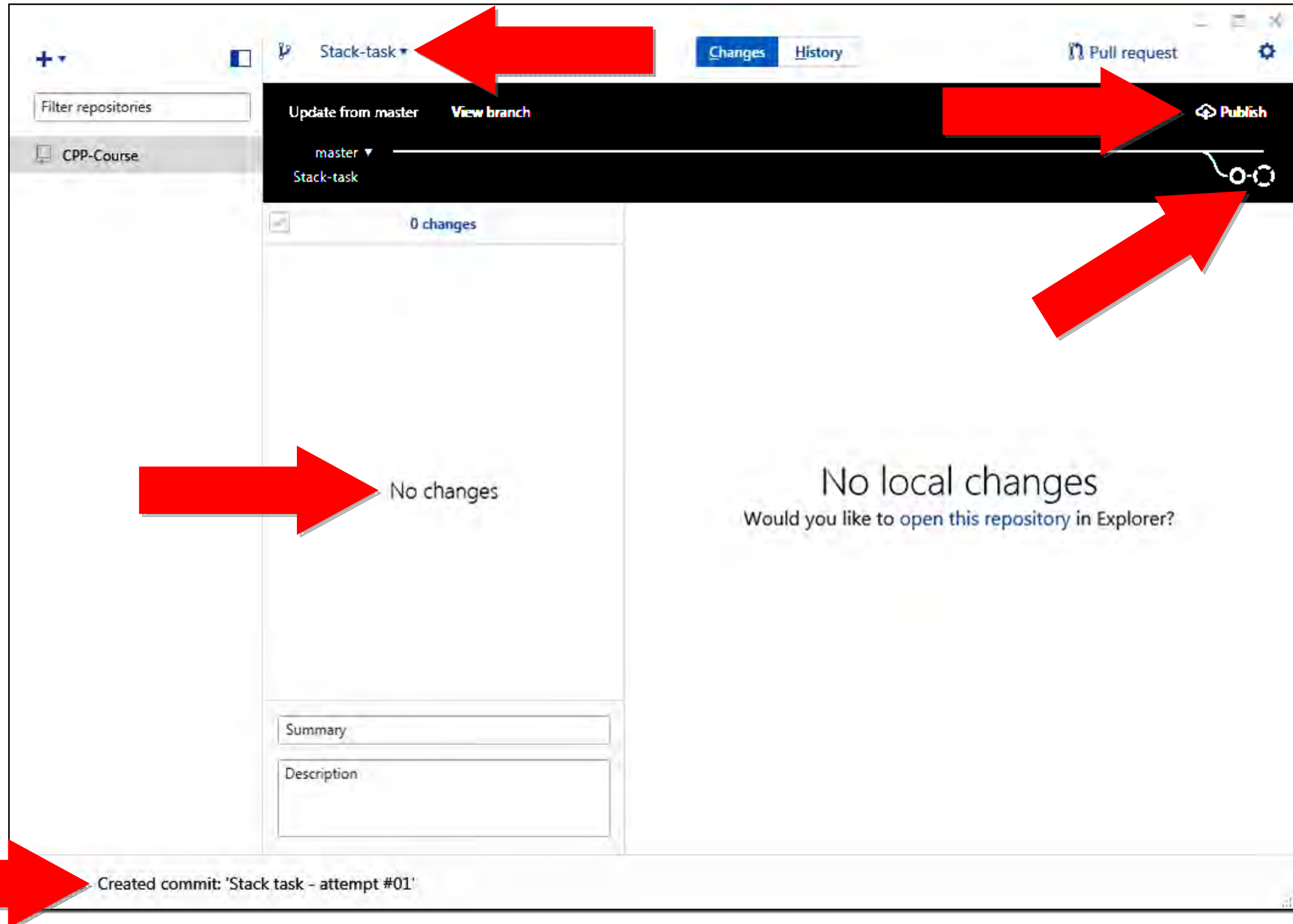
- ✓ README.md
- ✓ Stack.h
- ✓ StackTest.cpp

Stack task - attempt #01

После многих часов кодинга... я, кажется, сделал это... SRSLY?

Commit to Stack-task

```
Stack.h
12 +
13 + class CStack
14 + {
15 +     public:
16 +
17 +     bool ok()    const;
18 +     bool dump() const;
19 +
20 +     bool push (int value);
21 +     int  pop();
22 +
23 +     int  clear();
24 +     int  size() const;
25 +
26 +     static const int CStackCapacity = 100;
27 +
28 +     private:
29 +
30 +     int  data_ [CStackCapacity] = {};
31 +     int  size_                = 0;
32 + };
33 +
34 + //
35 +
36 + bool CStack::push (int value)
37 + {
38 +     ASSERT_OK();
39 + }
```



```
#include <stdio.h>

//=====

#define ASSERT_OK()  if (!ok()) { dump(); assert (!"Stack Error"); }

//=====

namespace Meow
{
    //=====

    class CStack
    {
    public:

        bool ok() const;
        bool dump() const;

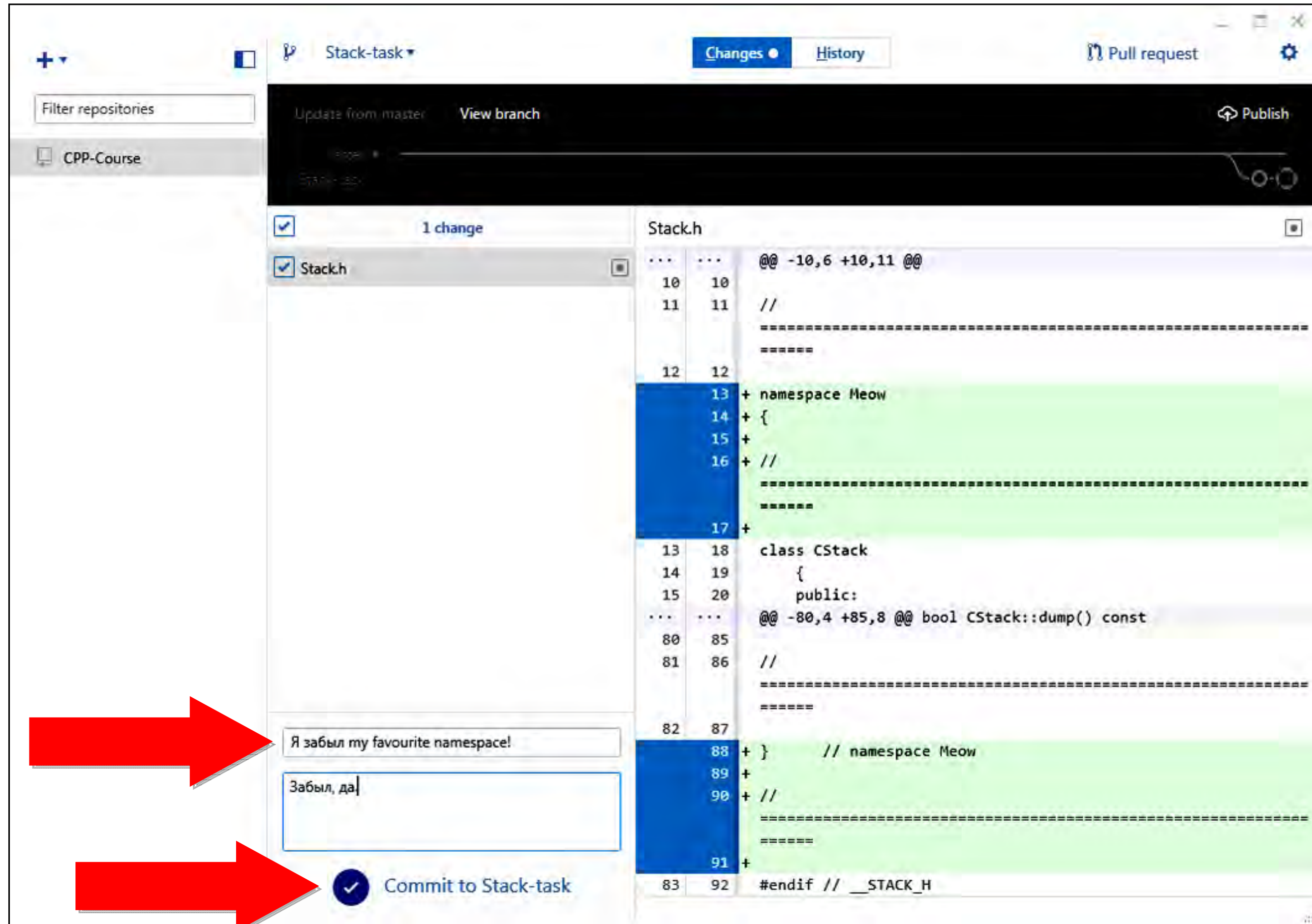
        bool push (int value);
        int pop();

        int clear();
        int size() const;

        static const int CStackCapacity = 100;

    private:

        int data_ [CStackCapacity] = {};
        int size_      = 0;
    };
}
```

The screenshot shows the GitHub Desktop interface for a repository named "Stack-task". The left sidebar shows a list of repositories with "CPP-Course" selected. The main area displays a list of changes, with "Stack.h" selected. The right pane shows the diff for "Stack.h", highlighting the addition of a namespace and a class. The commit message input field contains the text "Я забыл my favourite namespace!". The "Commit to Stack-task" button is visible at the bottom.

Stack-task

Changes History

Pull request

Update from master View branch

Filter repositories

CPP-Course

1 change

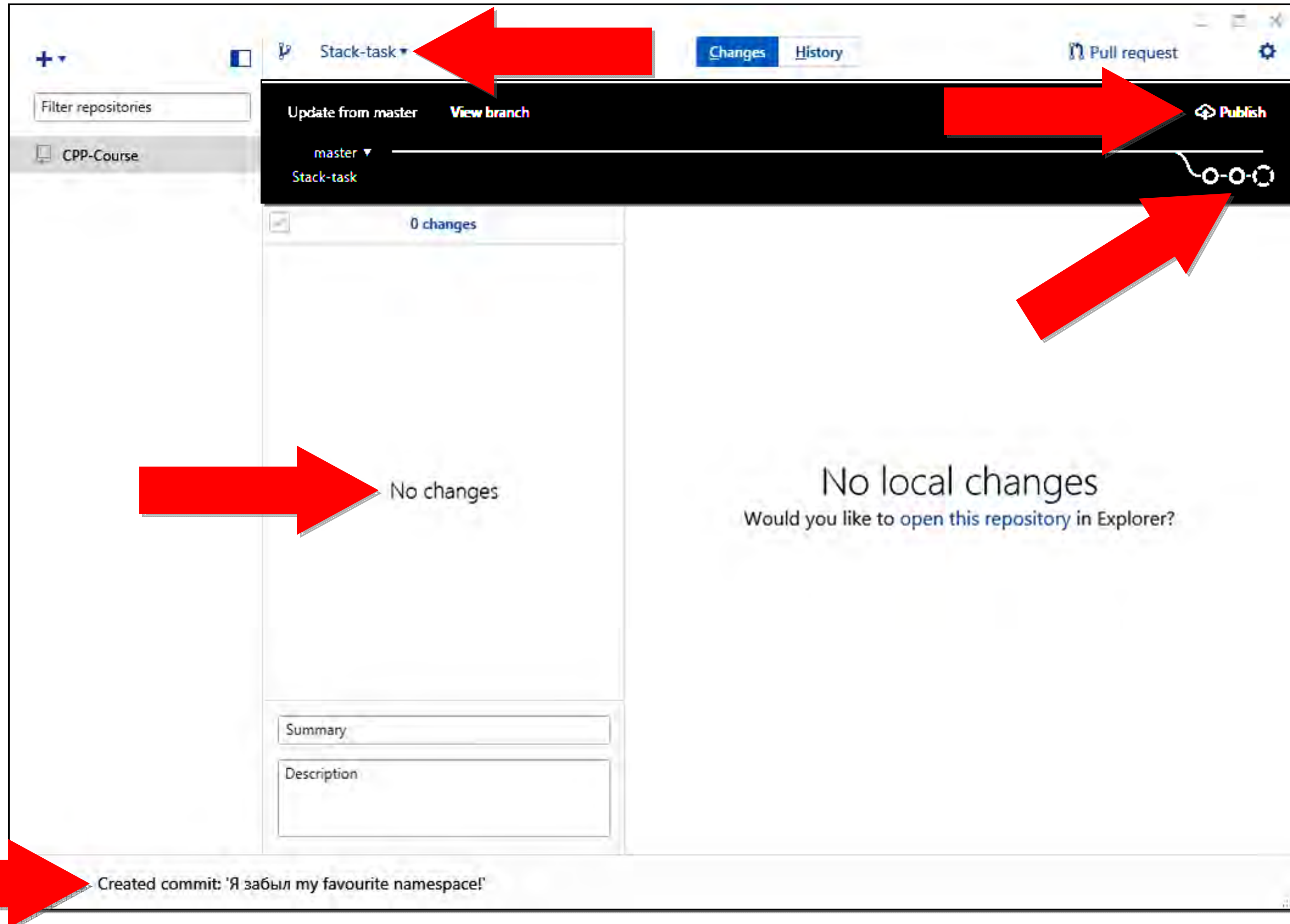
Stack.h

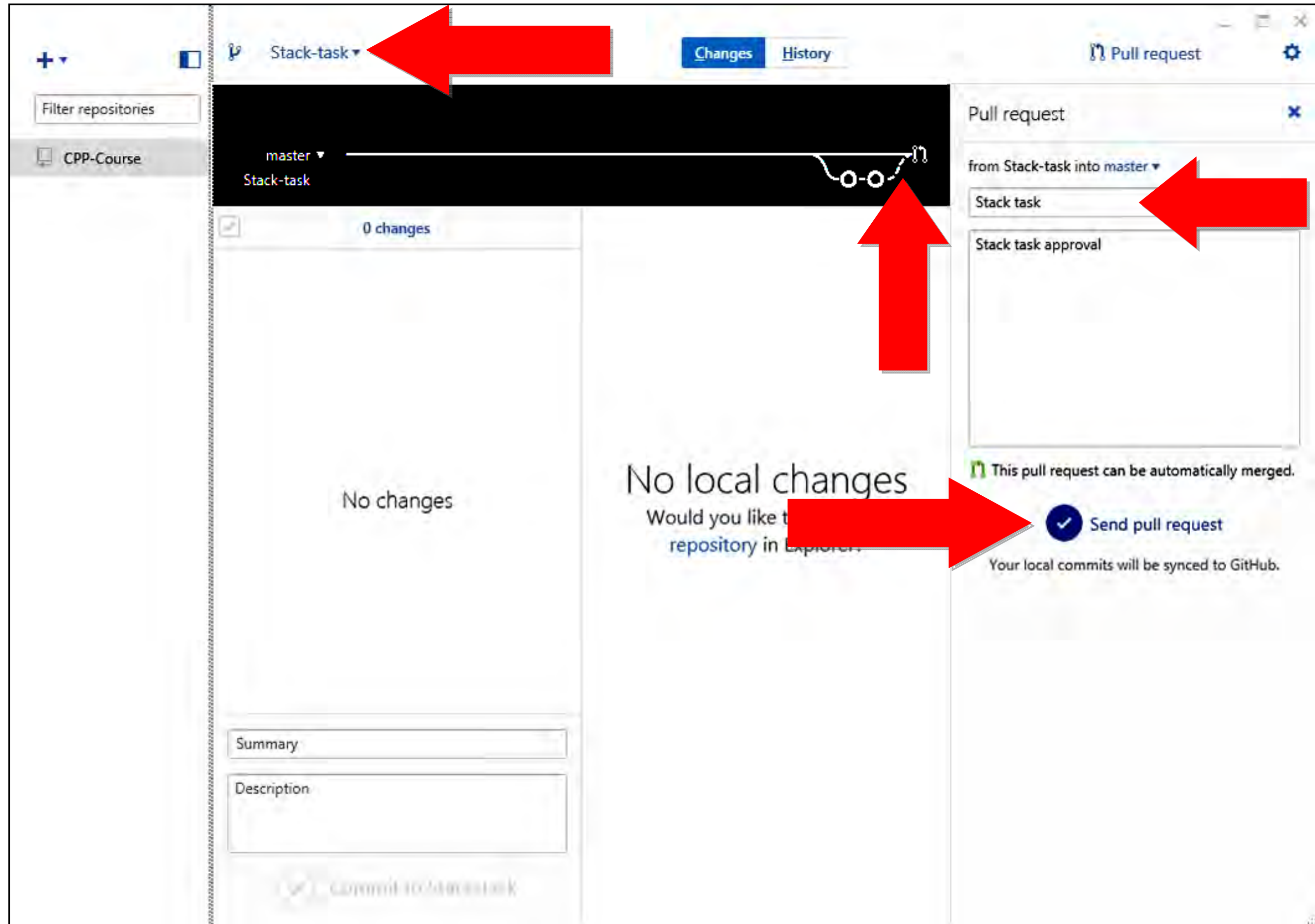
```
... 10 10 @@ -10,6 +10,11 @@
11 11 //
12 12
13 13 + namespace Meow
14 14 + {
15 15 +
16 16 + //
17 17 +
18 18 class CStack
19 19 {
20 20 public:
21 21 @@ -80,4 +85,8 @@ bool CStack::dump() const
22 22 //
23 23
24 24
25 25
26 26
27 27
28 28
29 29
30 30
31 31
32 32
33 33
34 34
35 35
36 36
37 37
38 38
39 39
40 40
41 41
42 42
43 43
44 44
45 45
46 46
47 47
48 48
49 49
50 50
51 51
52 52
53 53
54 54
55 55
56 56
57 57
58 58
59 59
60 60
61 61
62 62
63 63
64 64
65 65
66 66
67 67
68 68
69 69
70 70
71 71
72 72
73 73
74 74
75 75
76 76
77 77
78 78
79 79
80 80
81 81
82 82
83 83
84 84
85 85
86 86
87 87
88 88 + } // namespace Meow
89 89 +
90 90 + //
91 91 +
92 92 #endif // __STACK_H
```

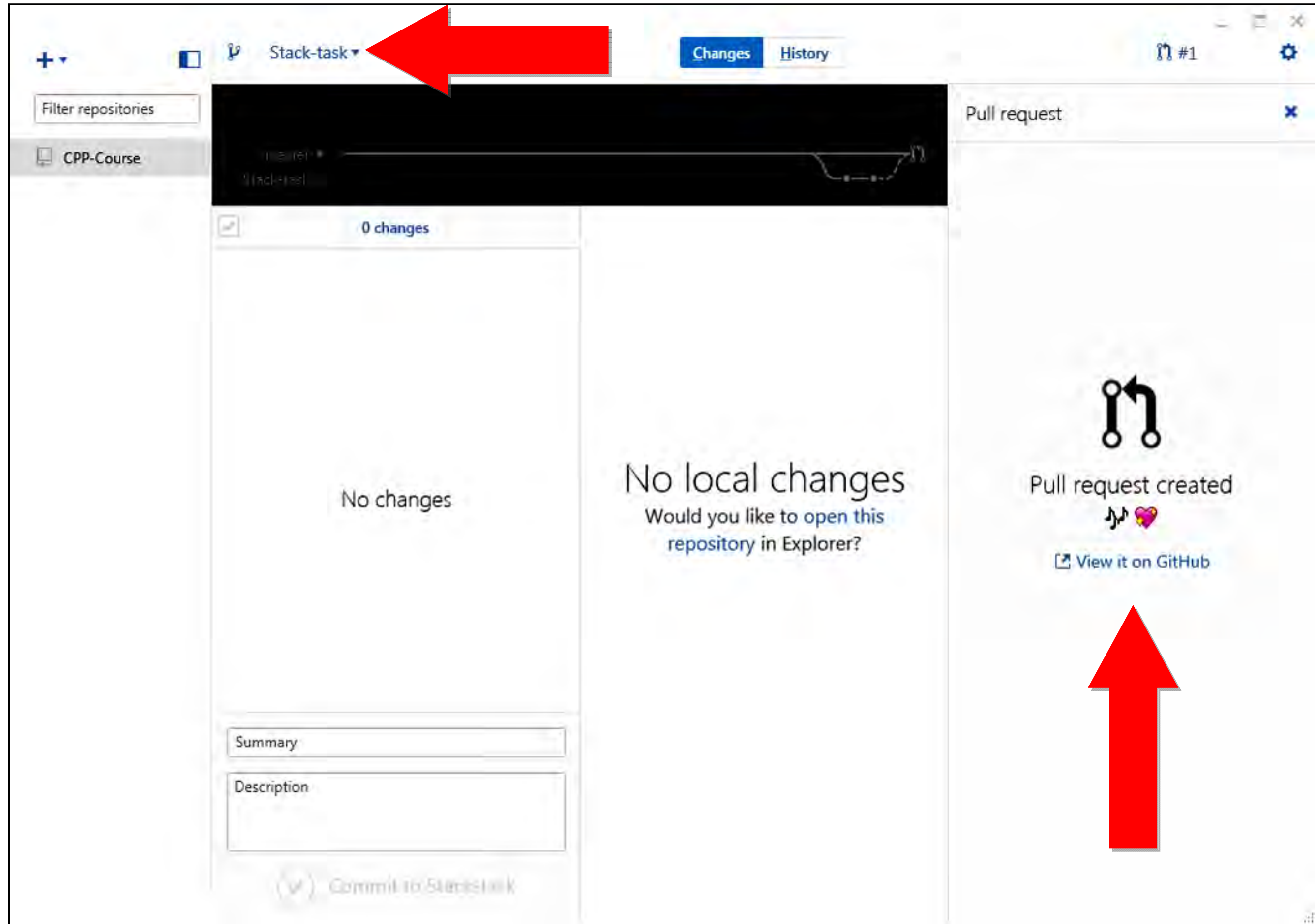
Я забыл my favourite namespace!

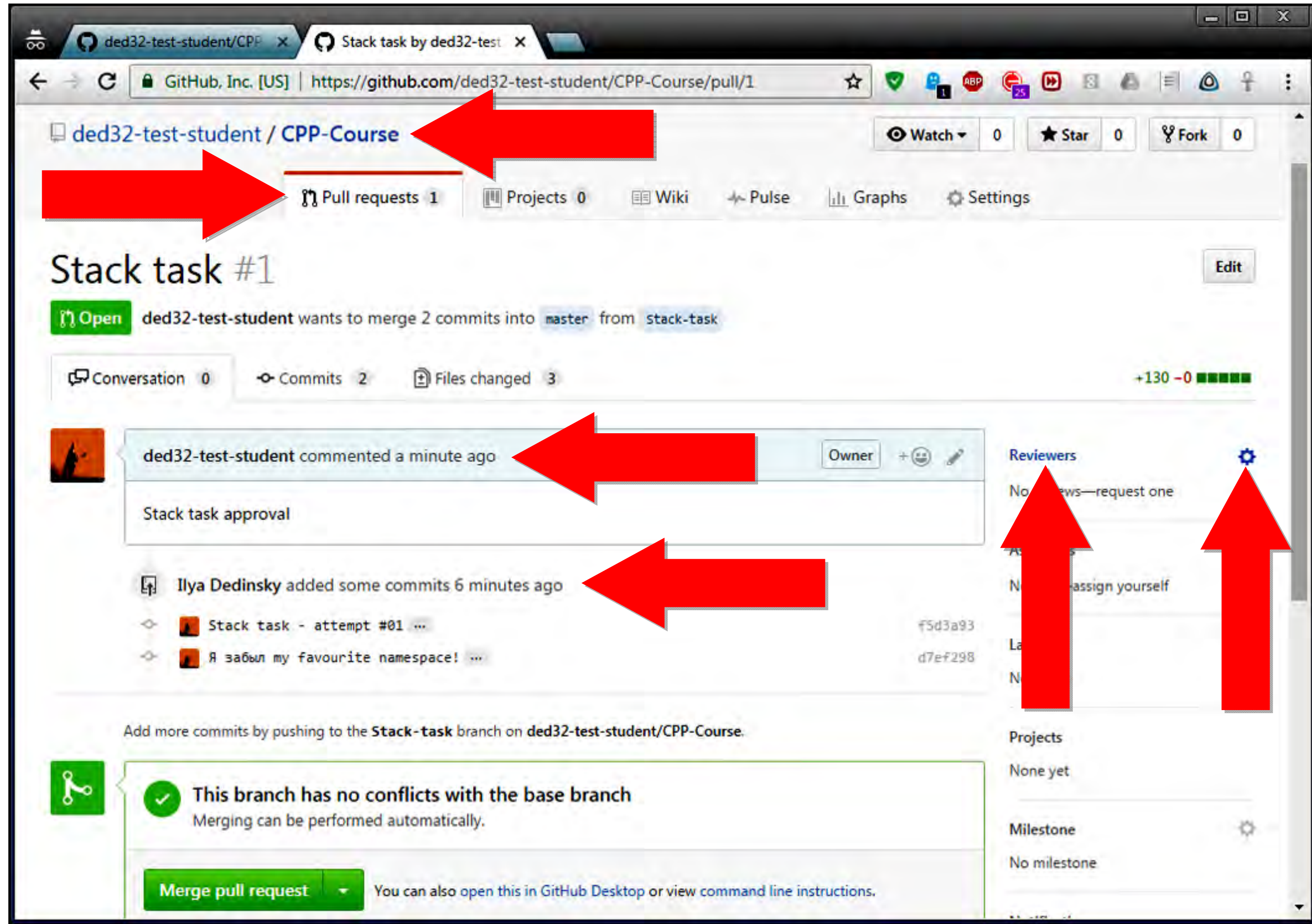
Забыл, да

Commit to Stack-task









The screenshot shows a GitHub pull request page for the repository 'ded32-test-student / CPP-Course'. The pull request is titled 'Stack task #1' and is created by 'ded32-test-student'. It shows 2 commits and 3 files changed. The pull request is open and ready for review. The page includes a sidebar with navigation links (Pull requests, Projects, Wiki, Pulse, Graphs, Settings) and a right sidebar with sections for Reviewers, Projects, and Milestone. A green box at the bottom indicates that the branch has no conflicts with the base branch and provides a 'Merge pull request' button.

ded32-test-student / CPP-Course

Stack task #1

ded32-test-student wants to merge 2 commits into master from Stack-task

Conversation 0 Commits 2 Files changed 3 +130 -0

ded32-test-student commented a minute ago

Stack task approval

Ilya Dedinsky added some commits 6 minutes ago

- Stack task - attempt #01 ...
- Я забыл my favourite namespace! ...

Add more commits by pushing to the Stack-task branch on ded32-test-student/ CPP-Course.

This branch has no conflicts with the base branch
Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Reviewers

No reviews—request one

Assign yourself

Projects

None yet

Milestone

No milestone

ded32-test-student / CPP-Course

Code

Issues 0

Pull requests 1

Projects 0

Wiki

Pulse

Graphs

Settings

Stack task #1

Edit

Open

ded32-test-student wants to merge 2 commits into master from Stack-task

Conversation 0

Commits 2

Files changed 3

+130 -0

ded32-test-student commented a minute ago

Owner

Reviewers

Stack task approval

Ilya Dedinsky added some commits 6 minutes ago

Stack task - attempt #01 ...

Я забыл my favourite namespace! ...

Add more commits by pushing to the Stack-task branch on ded32-test-student/ CPP-Course.

✓ This branch has no conflicts with the base branch

Merging can be performed automatically.

Merge pull request

You can also open this in GitHub Desktop or view command line instructions.

Request a review

Filter people

ded32 Ilya Dedinsky

Labels

None yet

Projects

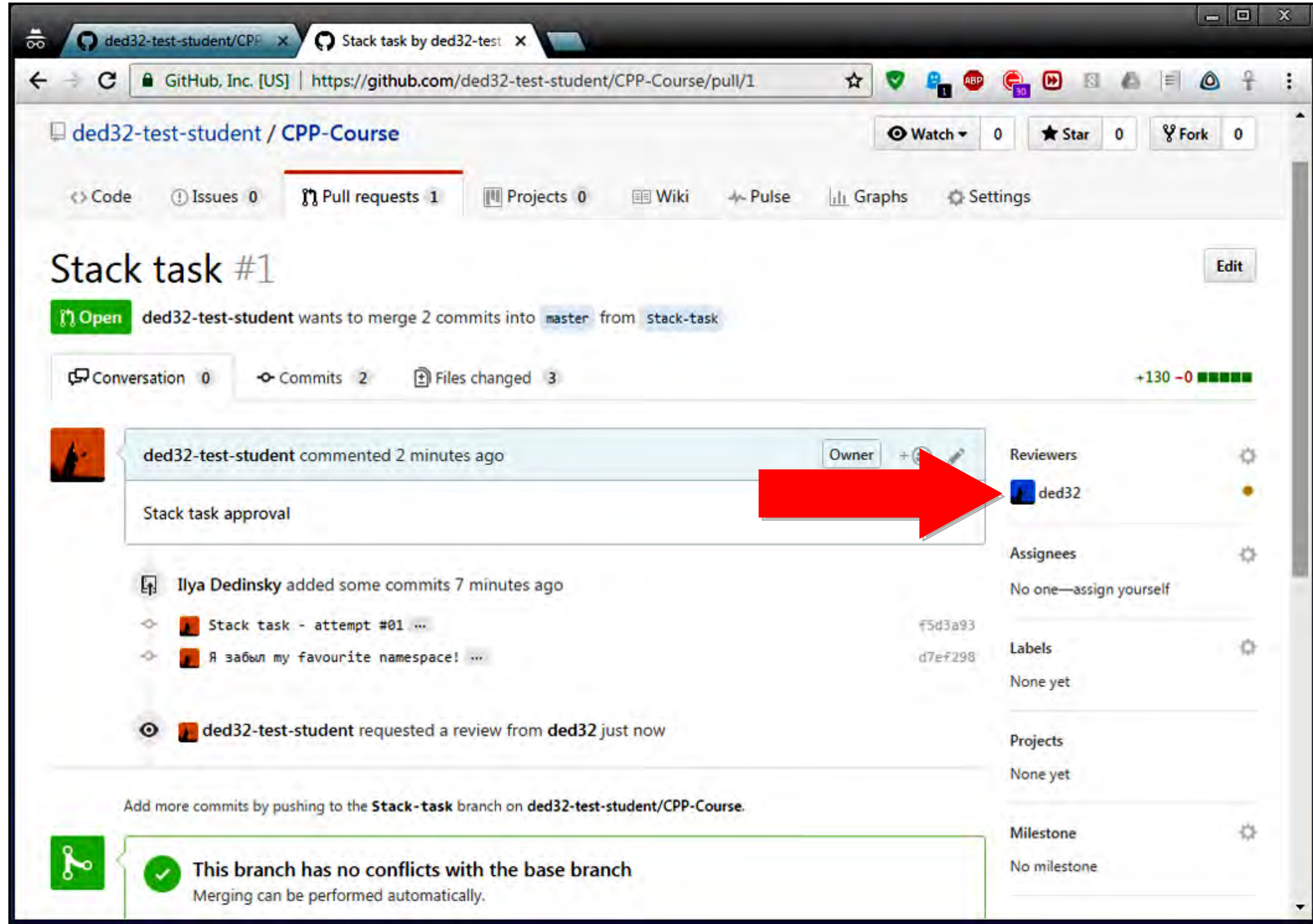
None yet

Milestone

No milestone

76

Pull request на сайте, выбираем ревьюера



The screenshot shows a GitHub pull request page for the repository 'ded32-test-student / CPP-Course'. The pull request is titled 'Stack task #1' and is created by 'ded32-test-student'. It shows 2 commits and 3 files changed. The pull request is currently open and has no reviews yet. A red arrow points to the 'Reviewers' section on the right, which lists 'ded32' as a reviewer. The 'Assignees' section shows 'No one—assign yourself'. The 'Labels' section shows 'None yet'. The 'Projects' section shows 'None yet'. The 'Milestone' section shows 'No milestone'. At the bottom, a green box indicates 'This branch has no conflicts with the base branch' and 'Merging can be performed automatically.'

ded32-test-student / CPP-Course

Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 1 Projects 0 Wiki Pulse Graphs Settings

Stack task #1

Edit

Open ded32-test-student wants to merge 2 commits into master from Stack-task

Conversation 0 Commits 2 Files changed 3 +130 -0

ded32-test-student commented 2 minutes ago

Stack task approval

Owner +

Reviewers

ded32

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Ilya Dedinsky added some commits 7 minutes ago

- Stack task - attempt #01 ... f5d3a93
- Я забыл my favourite namespace! ... d7ef298

ded32-test-student requested a review from ded32 just now

Add more commits by pushing to the Stack-task branch on ded32-test-student/ CPP-Course.

This branch has no conflicts with the base branch

Merging can be performed automatically.

ded32-test-student / CPP-Course

Watch 0

Star 0

Fork 0

Code

Issues 0

Pull requests 1

Projects 0

Wiki

Pulse

Graphs

Stack task #1

Edit

Open

ded32-test-student wants to merge 2 commits into master from Stack-task

Conversation 0

Commits 2

Files changed 3

Changes from all commits 3 files +130 -0

Unified Split Review changes

7 README.md

@@ -1,2 +1,9 @@

1 +

1 2 # CPP-Course

3 +

2 4 A repo for the CPP course. Long live the cats!

5 +

6 +# Now this repo contains the Stack.

7 +

8 +See: http://www.stroustrup.com/stack_cat.pdf

9 +

92 Stack.h

@@ -0,0 +1,92 @@

1 +

2 +#ifndef STACK_H

Submit your review

Review summary

Hi, great try. But you completely forgot about the *docs.* Please add doxygen formatting and run doxygen to generate the docs.

Comment

Submit general feedback without explicit approval.

Approve

Submit feedback and approve merging these changes.

☒ Request changes

Submit feedback that must be addressed before merging.

Submit review

Stack task #1


Open

ded32-test-student wants to merge 2 commits into master from stack-task

Conversation 1

Commits 2


Files changed 3




ded32-test-student commented 6 minutes ago


Owner + 😊 ✎


Stack task approval




Ilya Dedinsky added some commits 11 minutes ago


 Stack task - attempt #01 ... f5d3a93

 Я забыл my favourite namespace! ... d7ef298



ded32-test-student requested a review from ded32 4 minutes ago






ded32 requested changes just now

View changes

Hi, great try. But you completely forgot about the docs. Please add doxygen formatting and run doxygen to generate the docs.

Reviewers

 ded32

Assignees

No one—assignees

Labels

None yet


Projects

None yet

Milestone

No milestone

Notifications



79

Pull request, дадим ревьюверу что он хочет (иначе не отстанет)

```
//=====
//! @file stack.h
//{=====
//! @mainpage
//!
//! The MEPHI Nuclear Stack Library
//!
//! See the files:
//!   - stack.h
//!   - StackTest.cpp
//!
//! @note Please return (the_books) in the library before end of semester!
//!       Otherwise, rector's reactor may explode.
//!
//! @version 1.0
//!
//! @author (C) Ded, 2017
//}=====

#ifndef __STACK_H
#define __STACK_H

#include <stdio.h>

//=====
//{ General staff
//=====

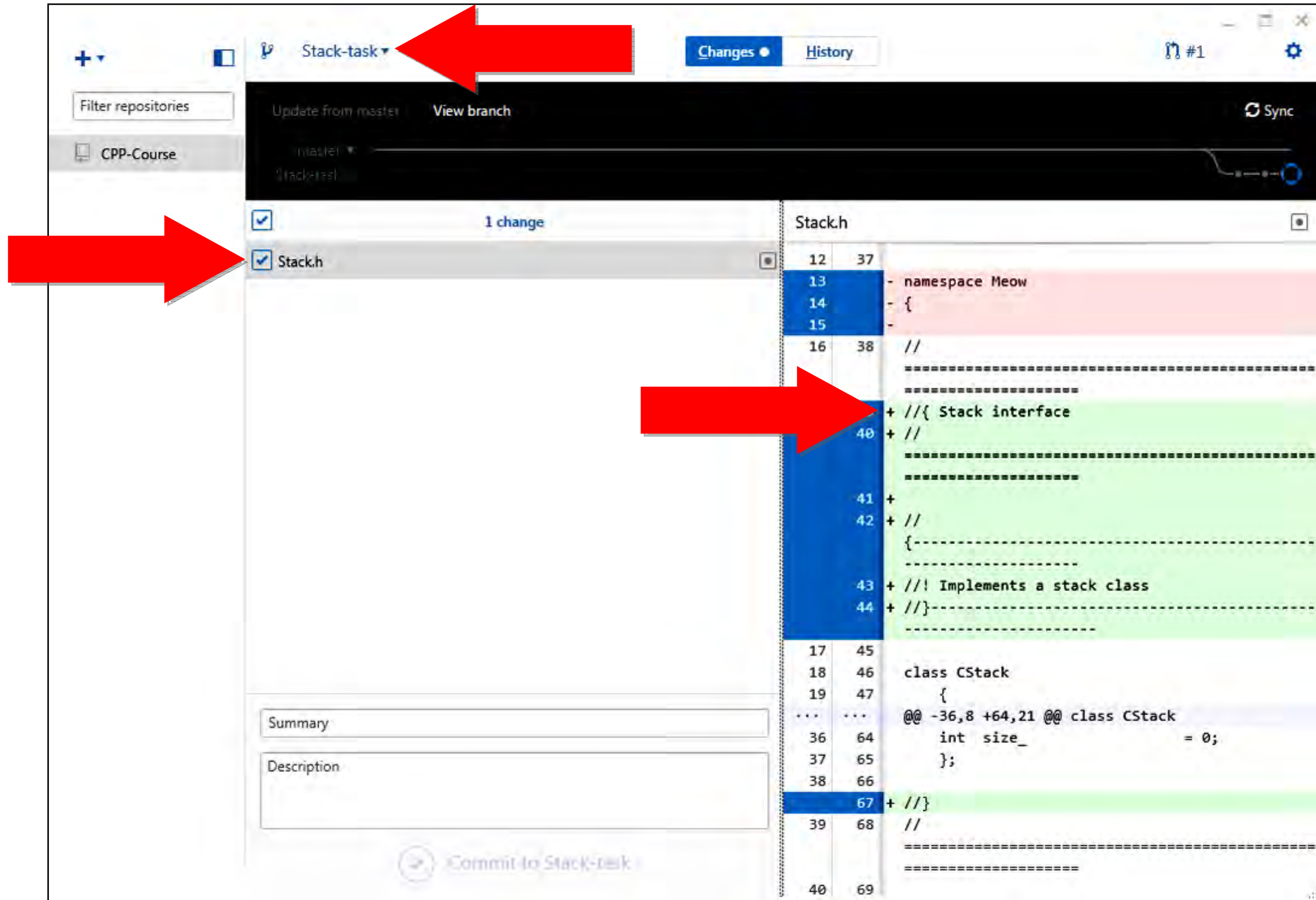
//=====
//{ Stack interface
//=====

//=====
//{ Stack implementation
//=====
```


Pull request, дадим ревьюеру что он хочет (он настырный)

```
//{-----  
//! Adds an element to the stack.  
//!  
//! @param value A value to be added.  
//!  
//! @return Operation status.  
//}-----  
  
bool CStack::push (int value)  
{  
    ASSERT_OK();  
  
    // TODO Feed the cats  
  
    ASSERT_OK();  
    return true;  
}  
  
//{-----  
//! Removes an element from the stack.  
//!  
//! @return Value of the element removed.  
//}-----  
  
int CStack::pop()  
{  
    ASSERT_OK();  
  
    // TODO Remove cats from the box (beware!)  
  
    ASSERT_OK();  
    return 0;  
}
```

Pull request, Github Desktop, измененные файлы на компе



Doxygen

Home

Downloads

Documentation

Extensions

Support

Doxygen

About

Downloads

Changelog

Documentation

Get Involved

Wish list

Examples

Links

Extensions

Support

Report bugs

Donate

GIT binaries

Windows binaries for the latest tagged releases are sometimes available as well.

Sources and Binaries

Latest release

The latest version of doxygen is 1.8.13 (release date December 29th 2016).

The source distribution

doxygen-1.8.13.src.tar.gz (4.7MB) (ftp | http)

A binary distribution for Linux x86-64

doxygen-1.8.13.linux.bin.tar.gz (16.0MB) (ftp | http)

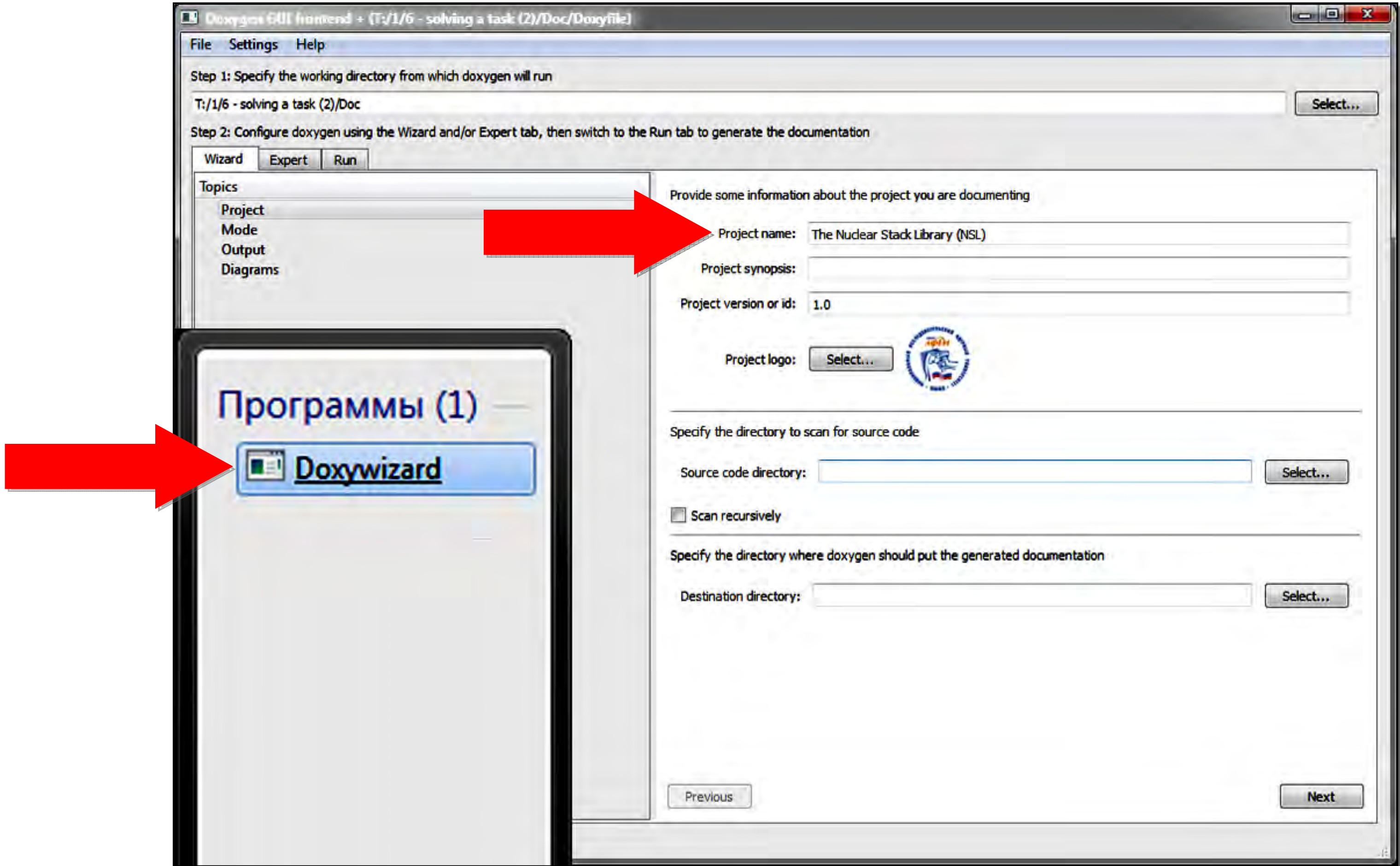
Compiled using Ubuntu 14.04 with kernel 3.13.0 and gcc 4.8.2. This archive includes the manual, but not the GUI frontend.

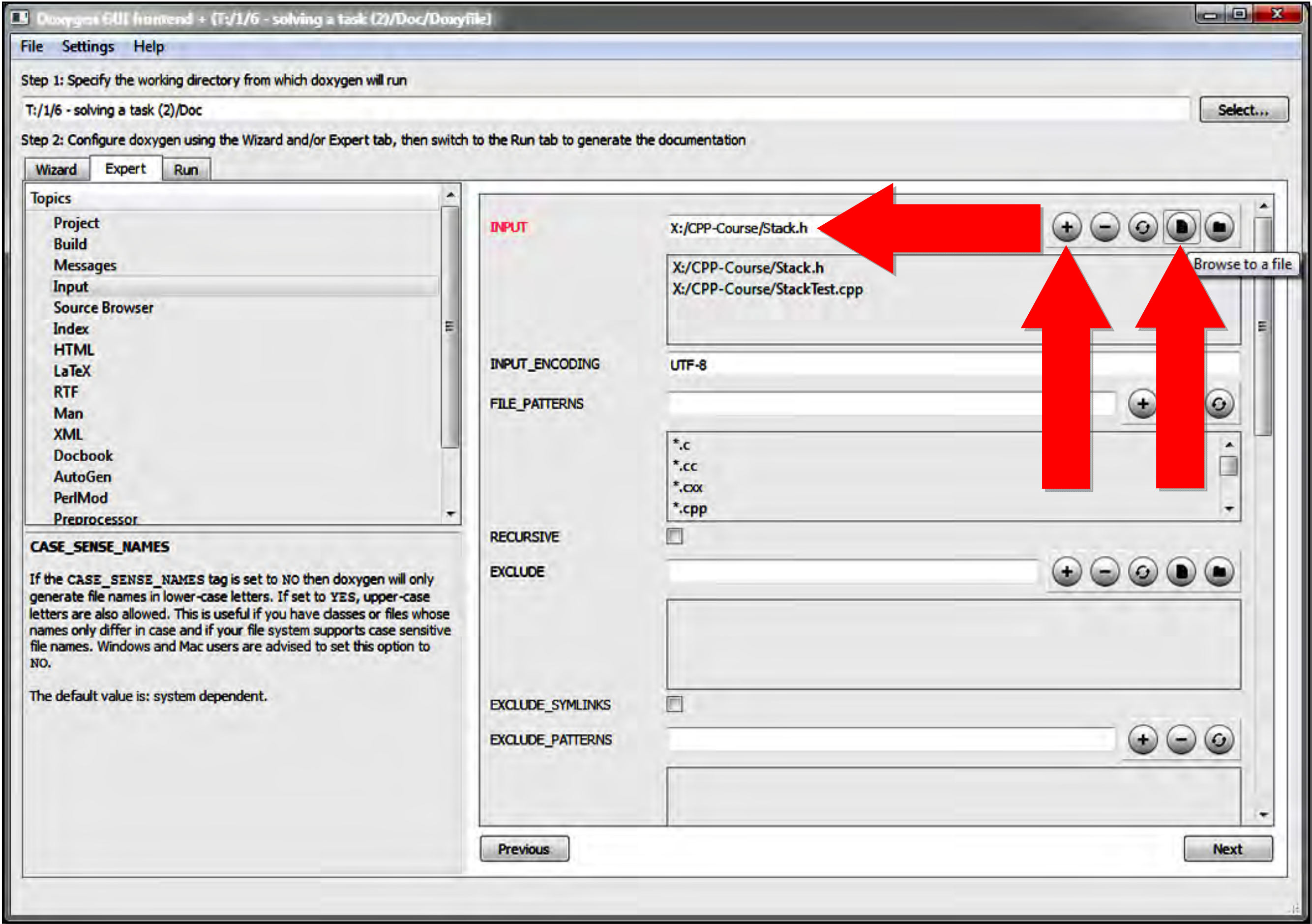
A binary distribution for Windows. All versions of Windows since XP are supported.

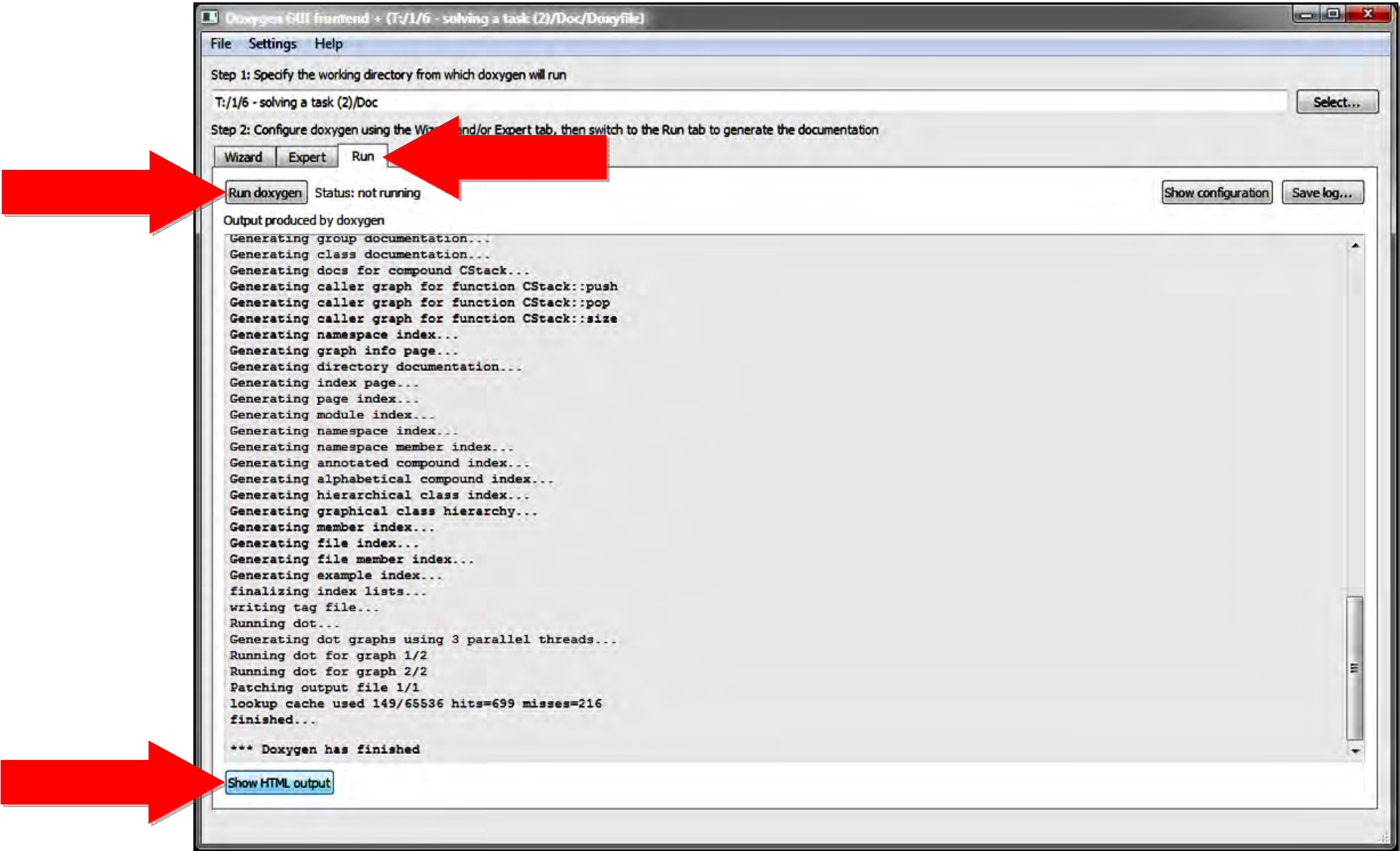
doxygen-1.8.13-setup.exe (24.4MB) (ftp | http)


This is a self-installing archive that includes the HTML and compressed HTML versions of the GUI frontend. It bundles 32-bit and 64-bit versions of doxygen.exe, and will install them on the OS.

If you are allergic to installers and GUIs, haven't sufficient bandwidth, or don't have admin









The Nuclear Stack Library (NSL) 1.0

Main PageClassesFiles

The Nuclear Stack Library (NSL)

- Classes
- Files

The Nuclear Stack Library (NSL) Documentation

The MEPHI Nuclear Stack Library

See the files:

- Stack.h
- StackTest.cpp

Note


Please return (the_books) in the library before end of semester! Otherwise, rector's reactor may explode.

Version

1.0

Author

(C) Ded, 2017



The Nuclear Stack Library (NSL) 1.0

Main Page

Classes ▾

Files ▾

▼ The Nuclear Stack Library (NSL)

▼ Classes

▼ Class List

► CStack

Class Index

► Class Members

► Files

Q Search

Public Member Functions | Static Public Attributes | Private Attributes | List of all members

CStack Class Reference

Implements a stack class. [More...](#)

#include <Stack.h>


Collaboration diagram for CStack:

CStack
+ CStackCapacity
- data_
- size_
+ ok()
+ dump()
+ push()
+ pop()
+ clear()
+ size()

Public Member Functions

bool	ok () const
bool	dump () const
bool	push (int value)
int	pop ()
int	clear ()
int	size () const

Static Public Attributes



The Nuclear Stack Library (NSL) 1.0

Main Page

Classes ▾

Files ▾

▼ The Nuclear Stack Library (NSL)

▼ Classes

▼ Class List

▼ CStack

ok

dump

push

pop

clear

size

CStackCapacity

data_

size_

Class Index

► Class Members

► Files

Q Search

◆ push()

bool CStack::push (int value)


Adds an element to the stack.

Parameters
value A value to be added.

Returns
Operation status.

Definition at line 81 of file Stack.h.

```
82 {  
83     ASSERT_OK();  
84     // TODO Feed the cats  
85     ASSERT_OK();  
86     return true;  
87 }
```

Here is the caller graph for this function:


```
graph RL  
    main --> ALMOST_GOOGLETEST_StackTest_Push42  
    ALMOST_GOOGLETEST_StackTest_Push42 --> CStack_push
```

◆ pop()

int CStack::pop ()

Github Desktop: Файлы с доками в локальном репо на компе

Filter repositories

CPP-Course

Stack-task

Changes

History

#1

Settings

Update from master

View branch

Sync

139 changes

Stack.h

Doc\html\Stack_8h_incl.map

Doc\html\Stack_8h_incl.md5

Doc\html\Stack_8h_incl.png

Doc\html\svgpan.js

Doc\html\sync_off.png

Doc\html\sync_on.png

Doc\html\tabs.css

Doc\html\tab_a.png

Doc\html\tab_b.png

Doc\html\tab_h.png

Doc\html\tab_s.png

Doc\Images\logo-m.png

Stack.h

Stack task - attempt #02

I did the docs! =^..^=

Commit to Stack-task

Stack.h

12 37

13 37 namespace Meow

14 37 {

15 37

16 38 //

17 45

18 46 class CStack

19 47 {

20 47

21 47 @@ -36,8 +64,21 @@ class CStack

22 47 int size_ = 0;

23 47

24 47

25 47

26 47

27 47

28 47

29 47

30 47

31 47

32 47

33 47

34 47

35 47

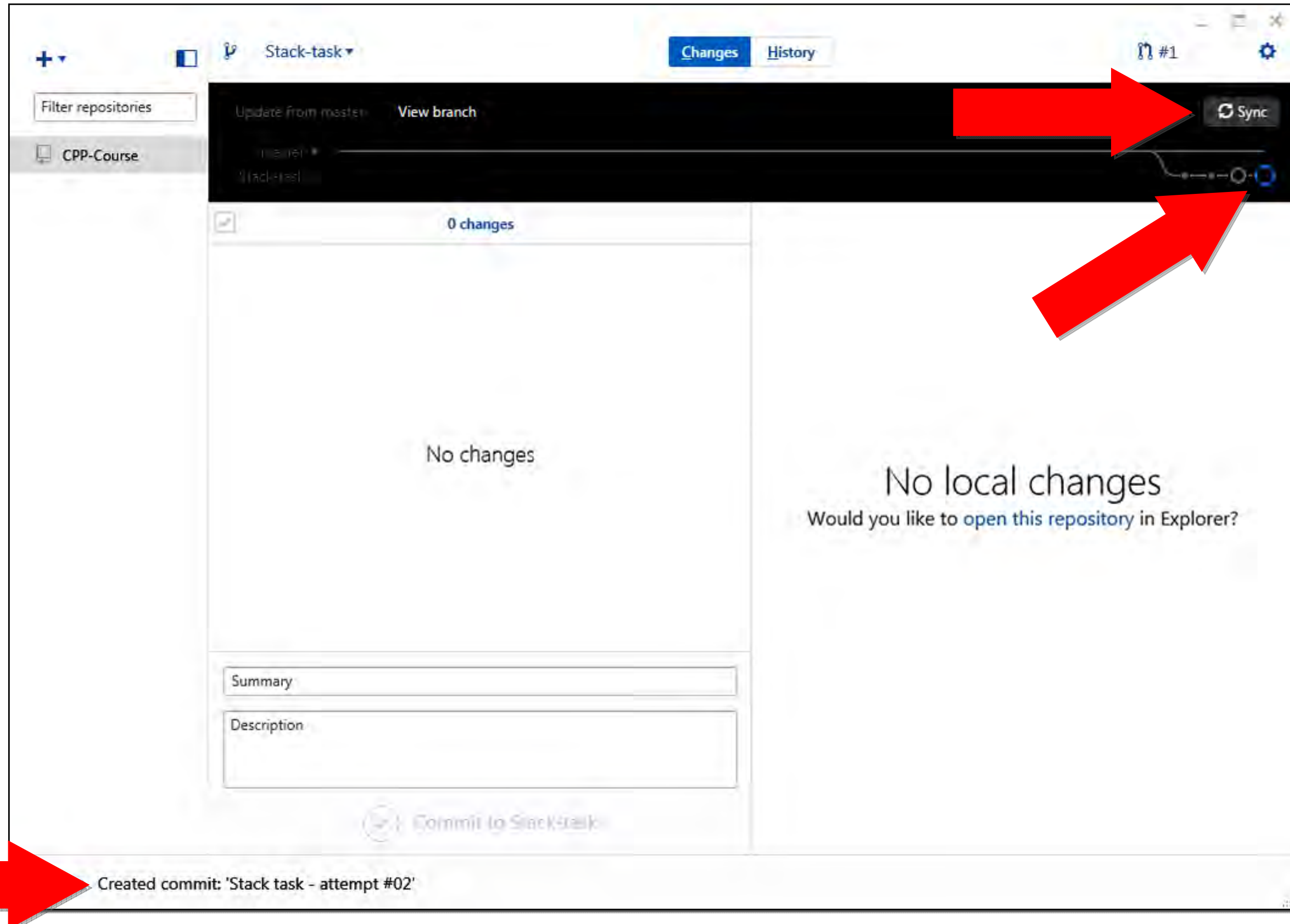
36 64



37 65

38 66


39 68


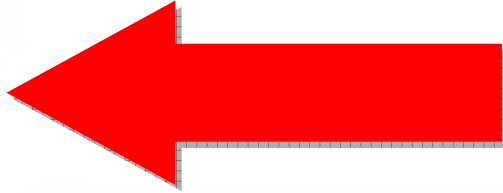
40 69




 ded32 requested changes an hour ago [View changes](#)


Hi, great try. But you completely forgot about the *docs*. Please add doxygen formatting and run doxygen to generate the docs.


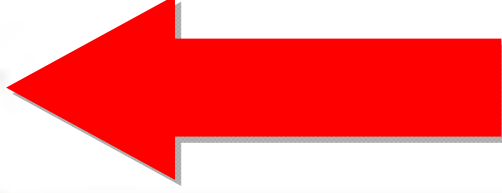
 New changes since you last viewed [View changes](#)


 Stack task - attempt #02 ...  d6d9ce9
I did the docs! =^..^=



ded32 commented 27 seconds ago [Collaborator](#) + 😊 ✎ ✕

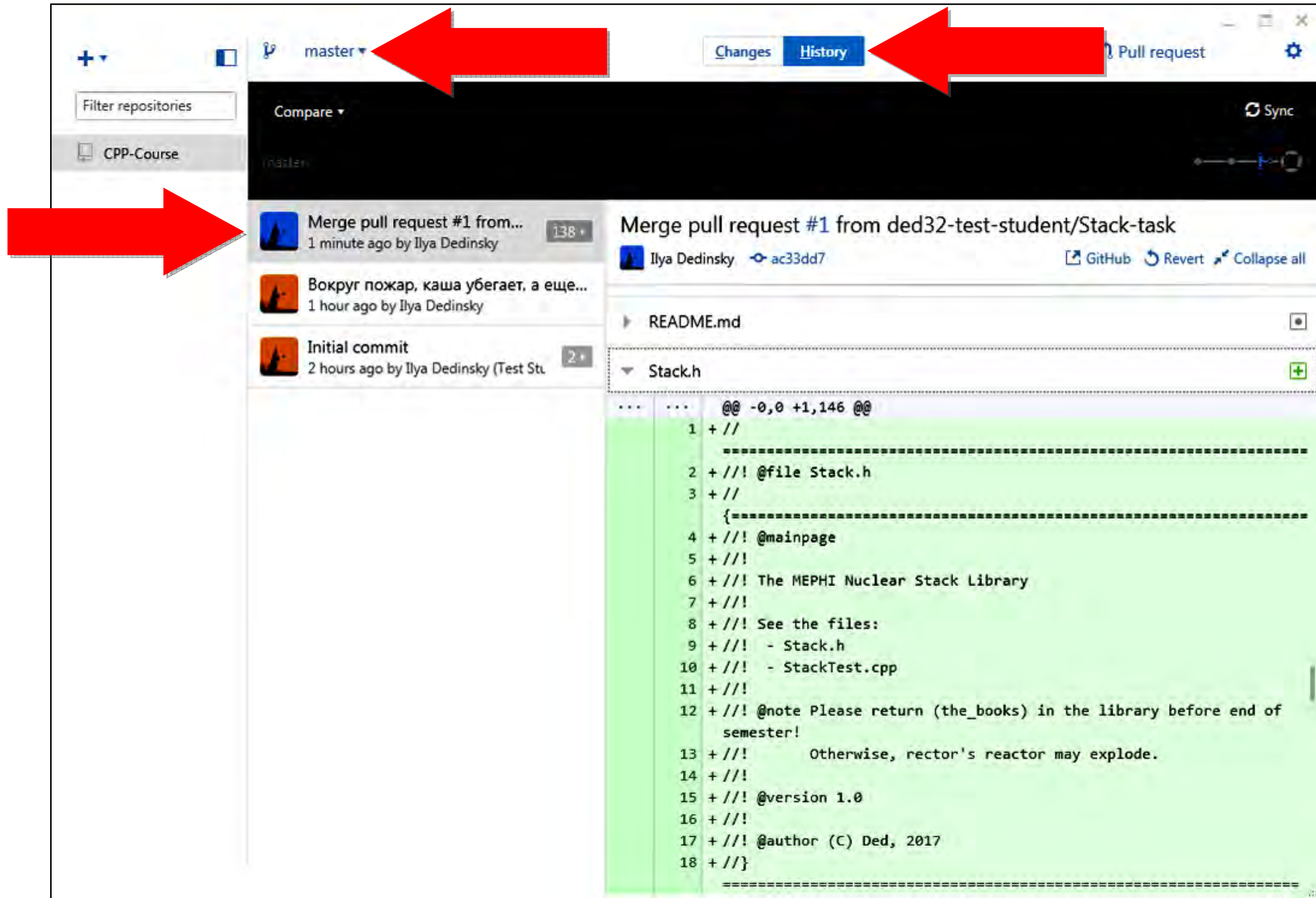
OK, this is fine! Approved now. 

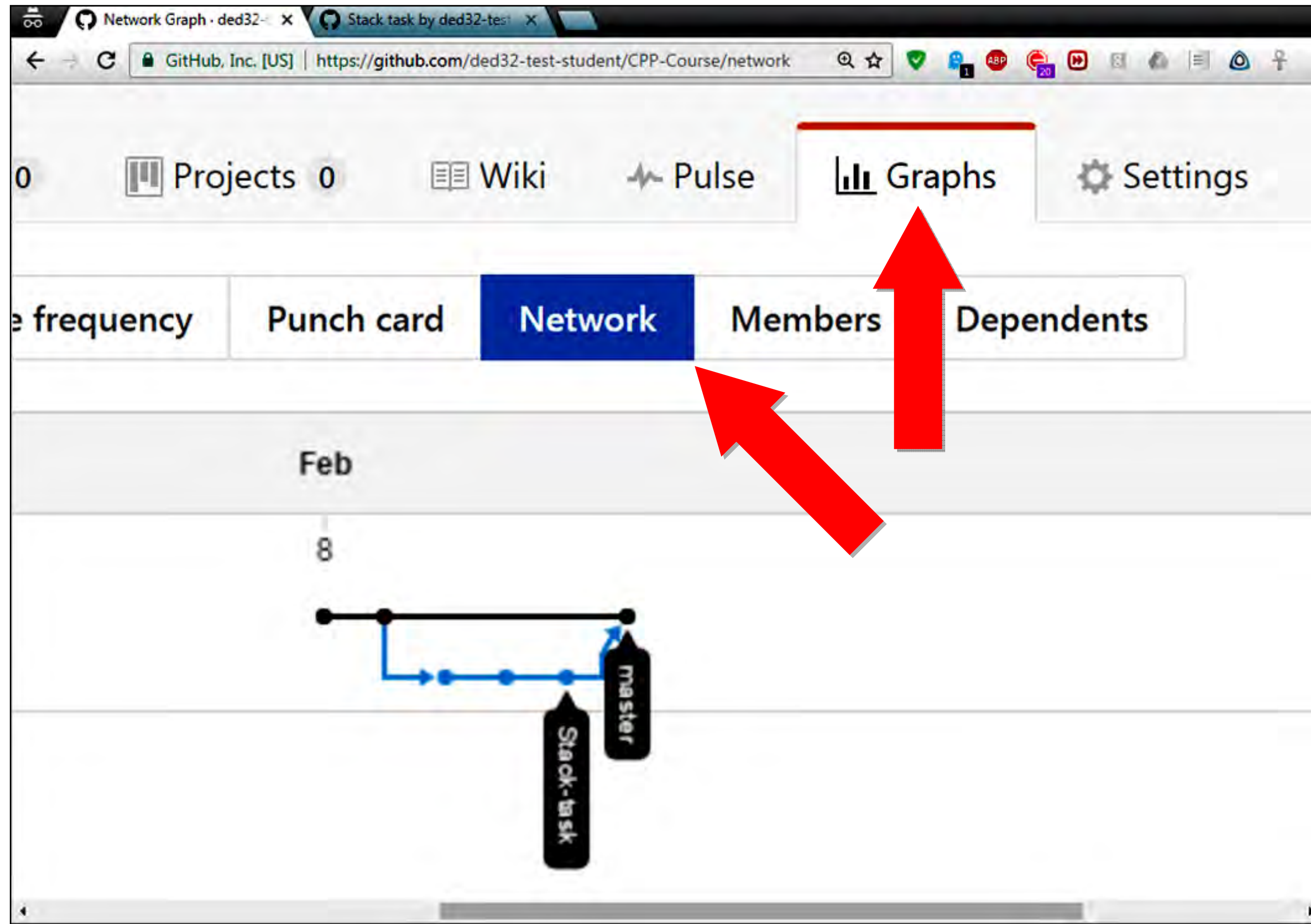
 ded32 merged commit **ac33dd7** into **master** just now  [Revert](#)



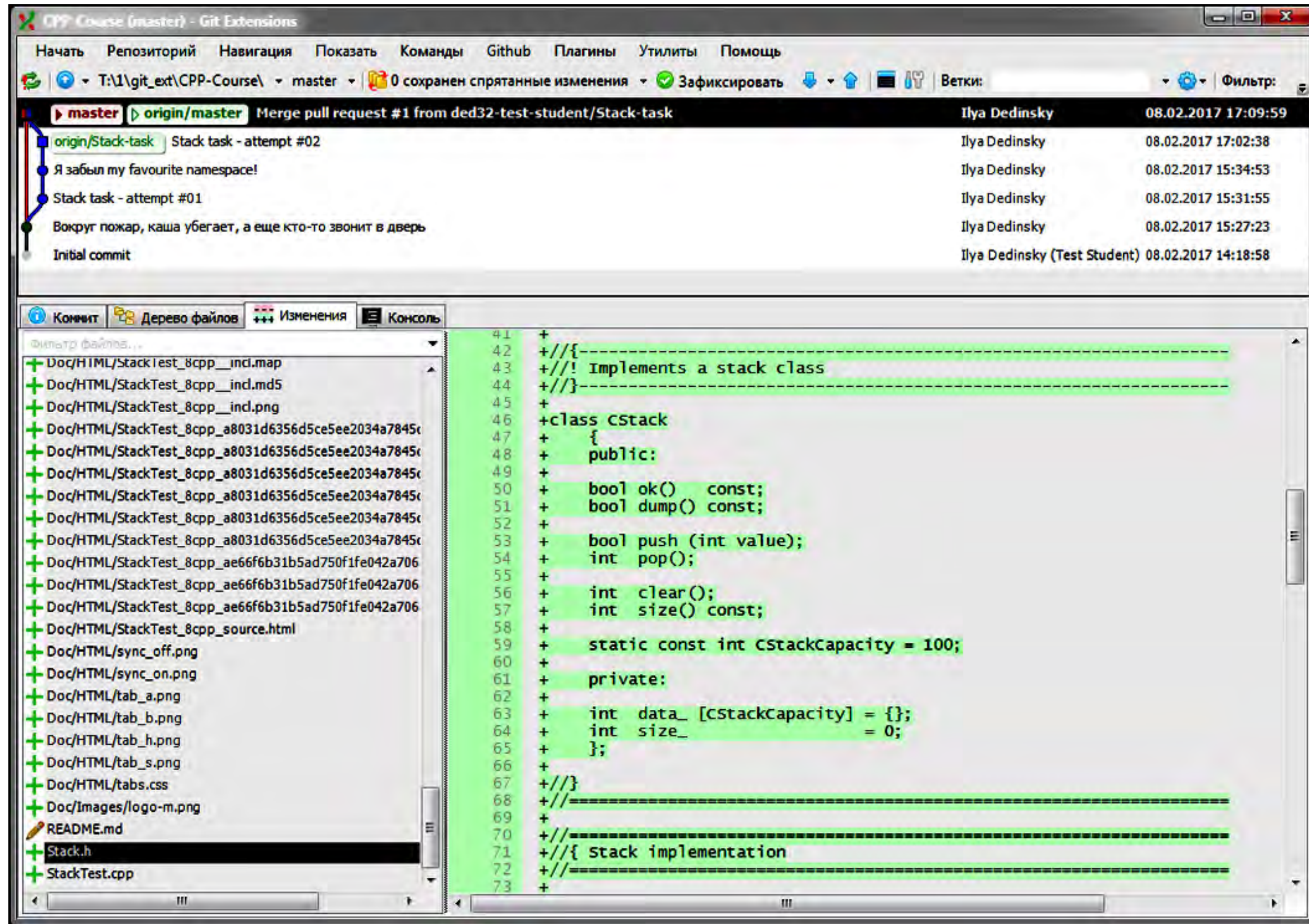
Pull request successfully merged and closed
You're all set—the **stack-task** branch can be safely deleted.

[Delete branch](#)









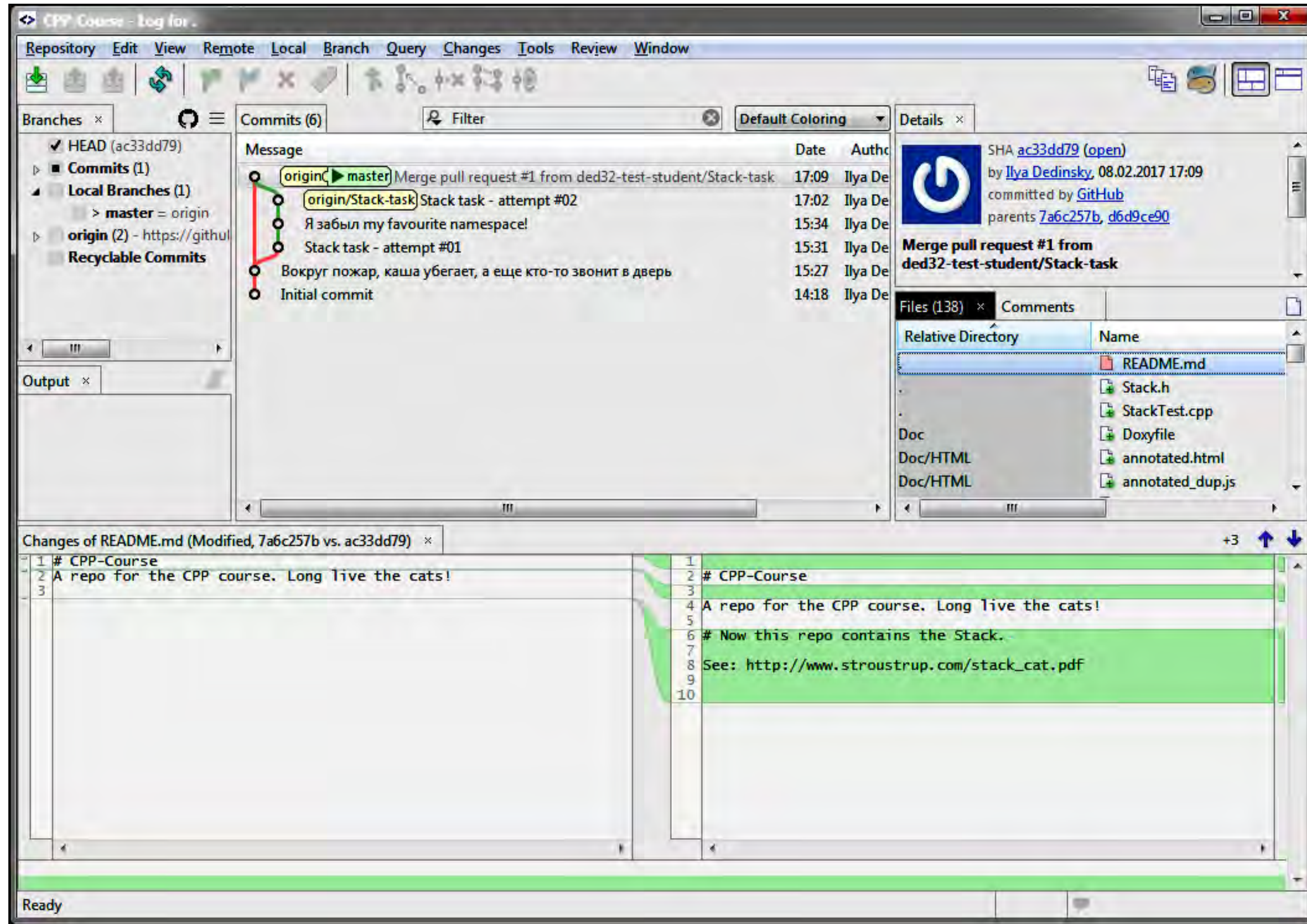
The screenshot shows the Git Extensions interface for a repository named "CPP Course (master)". The top menu bar includes options like "Начать", "Репозиторий", "Навигация", "Показать", "Команды", "Github", "Плагины", "Утилиты", and "Помощь". The status bar at the top indicates the current branch is "master" and shows a merge pull request from "ded32-test-student/Stack-task".

The left sidebar displays a commit history for the "origin/master" branch. The commits are listed in reverse chronological order:

- Initial commit
- Вокруг пожар, каша убегает, а еще кто-то звонит в дверь
- Stack task - attempt #01
- Я забыл my favourite namespace!
- origin/Stack-task Stack task - attempt #02

The main area shows a diff view for the "origin/Stack-task" branch. The diff highlights changes in the "Stack.h" file, showing the implementation of a stack class. The changes are as follows:

```
41 +
42 +//{-----
43 +//! Implements a stack class
44 +//}-----
45 +
46 +class Cstack
47 +{
48 +    public:
49 +
50 +    bool ok() const;
51 +    bool dump() const;
52 +
53 +    bool push (int value);
54 +    int pop();
55 +
56 +    int clear();
57 +    int size() const;
58 +
59 +    static const int CstackCapacity = 100;
60 +
61 +    private:
62 +
63 +    int data_ [CstackCapacity] = {};
64 +    int size_ = 0;
65 +};
66 +
67 +//}
68 +//=====
69 +
70 +//=====
71 +//{ stack implementation
72 +//=====
73 +
```

The screenshot shows the SmartGit application window for a repository named "CPP Course". The interface includes a menu bar (Repository, Edit, View, Remote, Local, Branch, Query, Changes, Tools, Review, Window), a toolbar, and several panels.

Left Panel: Shows the repository structure. The "HEAD (ac33dd79)" is checked. Under "Commits (1)", the "Local Branches (1)" section shows a branch named "master" pointing to the origin. The "origin (2) - https://github.com" is also listed.

Commits (6) Panel: Displays a list of commits. The "origin/master" branch is highlighted. The commits are:

- origin/master: Merge pull request #1 from ded32-test-student/Stack-task (17:09, Ilya De)
- origin/Stack-task: Stack task - attempt #02 (17:02, Ilya De)
- Я забыл my favourite namespace! (15:34, Ilya De)
- Stack task - attempt #01 (15:31, Ilya De)
- Вокруг пожар, каша убегает, а еще кто-то звонит в дверь (15:27, Ilya De)
- Initial commit (14:18, Ilya De)

Details Panel: Shows the details of the selected commit (SHA ac33dd79). It includes the commit message "Merge pull request #1 from ded32-test-student/Stack-task", the author "Ilya Dedinsky", the date "08.02.2017 17:09", and the parents "7a6c257b, d6d9ce90".

Files (138) Panel: Lists the files in the repository. The "README.md" file is selected.

Changes of README.md (Modified, 7a6c257b vs. ac33dd79) Panel: Shows the diff between the two versions of the file. The changes are highlighted in green.

```
1 # CPP-Course
2 A repo for the CPP course. Long live the cats!
3
4 # Now this repo contains the Stack.
5
6 See: http://www.stroustrup.com/stack_cat.pdf
7
8
9
10
```

The status bar at the bottom indicates "Ready".

Оставьте фидбек по лекции

На Техноатоме, не на Гитхабе :)

ТЕХНОАТОМ



Спасибо за внимание

&&

Задавайте вопросы!