**Задание 1. Получение проекта**

Выполните следующие шаги:

**На GitHub:**

1. Создайте новый репозиторий с файлом readme.md.
2. Создайте ветку develop.

**Локально:**

1. Клонируйте репозиторий с GitHub на компьютер.
2. Откройте проект в PyCharm (3-й и 4-й пункт можно сразу сделать из PyCharm).
3. Сделайте checkout удаленной ветки develop (станет текущей).

Изображение выглядит как текст

Автоматически созданное описание

1. Создайте, заполните и закоммитьте файл .gitignore.

Подумайте, какие файлы надо игнорировать при работе из PyCharm. Важно, чтобы лишние файлы, которые создает PyCharm, не попадали в коммиты.

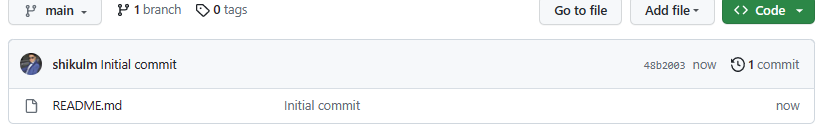
1. Отправьте изменения на GitHub.

**Выполнение**

1. Создайте новый репозиторий с файлом readme.md.

Изображение выглядит как текст

Автоматически созданное описание



1. Создайте ветку develop.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

1. Клонируйте репозиторий с GitHub на компьютер.
2. Откройте проект в PyCharm (3-й и 4-й пункт можно сразу сделать из PyCharm).

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст, Веб-сайт

Автоматически созданное описание

1. Сделайте checkout удаленной ветки develop (станет текущей).

Изображение выглядит как текст

Автоматически созданное описание

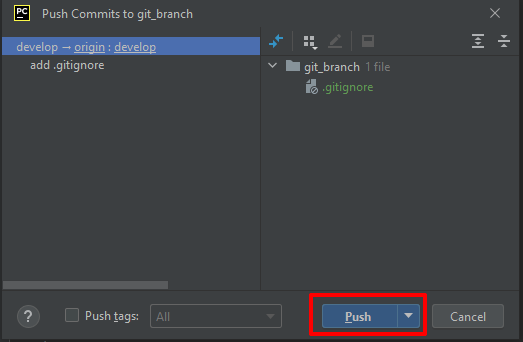
1. Создайте, заполните и закоммитьте файл .gitignore.

Подумайте, какие файлы надо игнорировать при работе из PyCharm. Важно, чтобы лишние файлы, которые создает PyCharm, не попадали в коммиты.

Изображение выглядит как текст

Автоматически созданное описание

1. Отправьте изменения на GitHub.



**Задание 2. Первый пул-реквест**

Выполните следующие шаги:

1. Создайте и переключитесь на ветку feature/Task2.
2. Создайте файл str\_func.py  и в нем функцию, которая принимает на вход строку и возвращает ее со всеми заглавными буквами.
3. Закоммитьте проделанную работу и отправьте изменения на GitHub.

**На GitHub:**

1. Сделайте pull request из ветки feature/Task2 в develop.

После pull request вашему тимлиду приходит уведомление. Он смотрит, что было сделано, вносит комментарии, отклоняет или принимает пул-реквест.

В учебных целях действия тимлида вы делаете самостоятельно.

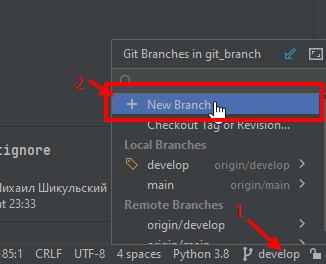
1. Сделайте за тимлида мердж с удалением ветки feature/Task2.

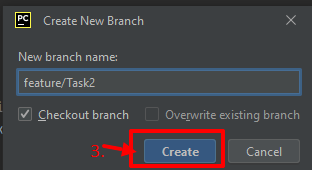
**Локально:**

1. Удалите локальную ветку feature/Task2.
2. Обновите локальную ветку develop.

**Выполнение**

1. Создайте и переключитесь на ветку feature/Task2.





1. Создайте файл str\_func.py  и в нем функцию, которая принимает на вход строку и возвращает ее со всеми заглавными буквами.

Изображение выглядит как текст

Автоматически созданное описание

1. Закоммитьте проделанную работу и отправьте изменения на GitHub.



Изображение выглядит как текст

Автоматически созданное описание

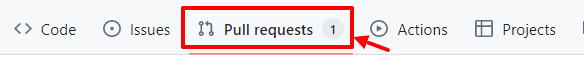
Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание

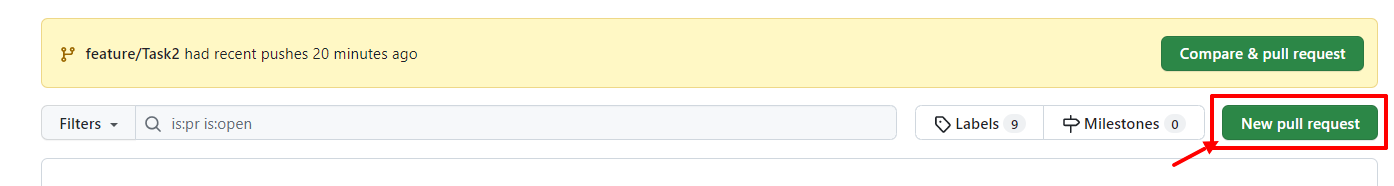
**На GitHub:**

1. Сделайте pull request из ветки feature/Task2 в develop.

Выбираем в меню:



Далее пошагово на открывшейся вкладке делаем pull request



Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

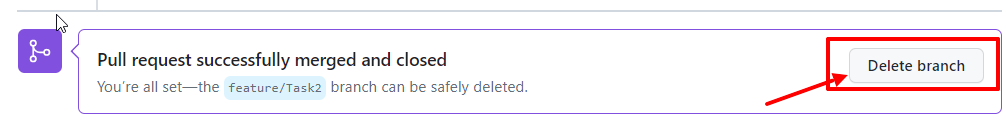
Автоматически созданное описание

1. Сделайте за тимлида мердж с удалением ветки feature/Task2.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

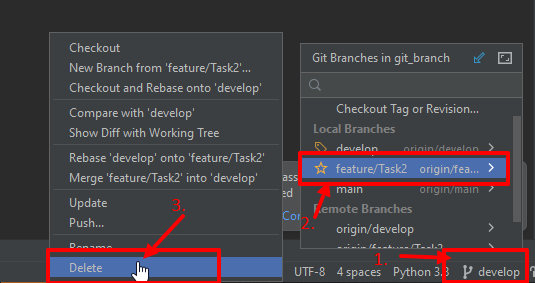
Автоматически созданное описание  


Ветвь feature/task2 удалилась:



**Локально:**

1. Удалите локальную ветку feature/Task2.



Изображение выглядит как текст

Автоматически созданное описание

1. Обновите локальную ветку develop.



Изображение выглядит как текст, табло, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, устройство, снимок экрана

Автоматически созданное описание

Появился коммит с pull request для ветки develop:

Изображение выглядит как текст, Веб-сайт

Автоматически созданное описание

**Задание 3. Наращивание функционала**

Тимлид дал задачу написать еще одну функцию и попросил не забывать добавлять описание к функциям (docstring).

Выполните следующие шаги:

**На GitHub:**

1. Добавьте docstring к функции из предыдущего задания (прямо в ветке develop). Этим вы сымитируете проделанную работу другого разработчика над этим же модулем.

**Локально:**

1. Напишите в файле str\_func.py вторую функцию, которая делает заглавными первые буквы каждого слова в строке, поступившей на вход функции. Добавьте к функции docstring.

Помните, что вы работаете по GitFlow. Разработка в ветке main и develop не ведется. Изменения туда попадают только через pull request. Что это значит для выполнения этого задания?

1. Добавьте docstring к функции из Задания 2.
2. Отправьте проделанную работу коллегам на код-ревью (т. е. pull-request, который должен показать, что есть конфликт: **This branch has conflicts that must be resolved**).

Изображение выглядит как текст

Автоматически созданное описание

**Выполнение**

**На GitHub:**

1. Добавьте docstring к функции из предыдущего задания (прямо в ветке develop). Этим вы сымитируете проделанную работу другого разработчика над этим же модулем.

Изображение выглядит как текст

Автоматически созданное описание

**Локально:**

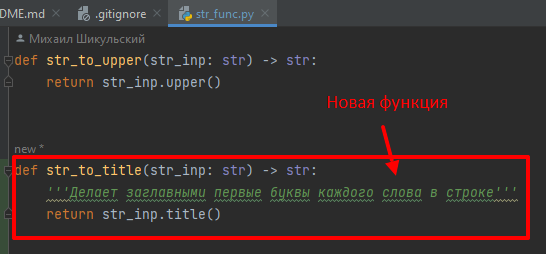
1. Напишите в файле str\_func.py вторую функцию, которая делает заглавными первые буквы каждого слова в строке, поступившей на вход функции. Добавьте к функции docstring.

Помните, что вы работаете по GitFlow. Разработка в ветке main и develop не ведется. Изменения туда попадают только через pull request. Что это значит для выполнения этого задания?

Изображение выглядит как Веб-сайт

Автоматически созданное описание

В новую ветвь добавил функцию



1. Добавьте docstring к функции из Задания 2.

Изображение выглядит как текст

Автоматически созданное описание

1. Отправьте проделанную работу коллегам на код-ревью (т. е. pull-request, который должен показать, что есть конфликт: **This branch has conflicts that must be resolved**).

Ltk

LtkEcnf

Делаем коммит изменений на клиенте и отправляем в GitHub



Изображение выглядит как текст, монитор, снимок экрана, черный

Автоматически созданное описание

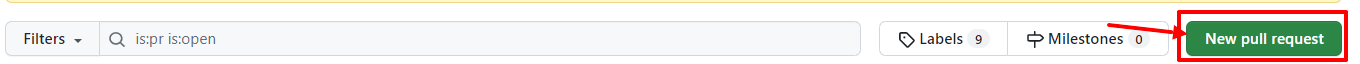
Изображение выглядит как текст

Автоматически созданное описание

Создаем Pull Requet в GitHub

Изображение выглядит как стрела

Автоматически созданное описание



Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

**Задание 4. Решение конфликтов**

Ваш пул-реквест не прошел, так как есть конфликт. Другой программист уже внес описание к вашей первой функции, пока вы работали над второй.

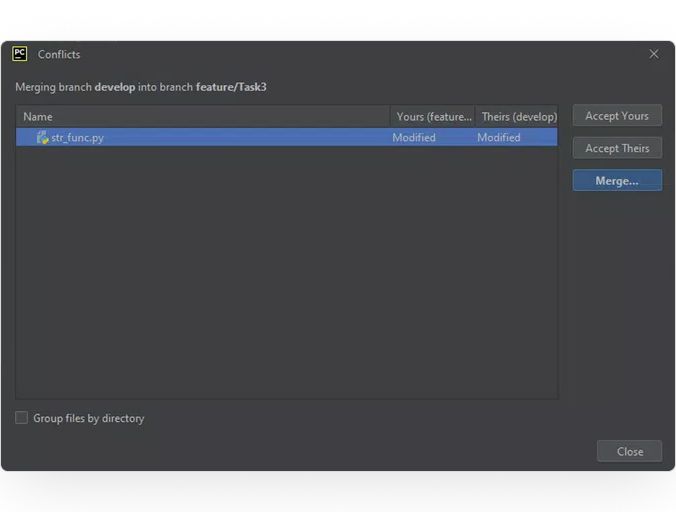
Выполните следующие шаги:

1. Подтяните изменения в ветку develop (у вас должны появиться правки другого программиста).

Изображение выглядит как текст

Автоматически созданное описание

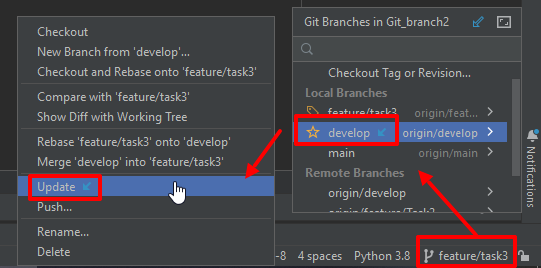
1. Сделайте мердж develop в текущую ветку, где вы ведете разработку (вам предложат порешать конфликт).



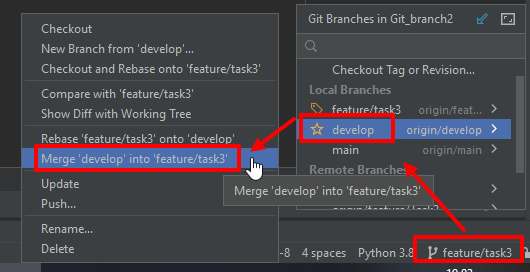
1. Решите конфликт, отклонив описание функции другого программиста.
2. Сделайте пуш.
3. За тимлида заапрувьте пул-реквест на GitHub.

**Выполнение**

1. Подтяните изменения в ветку develop (у вас должны появиться правки другого программиста).



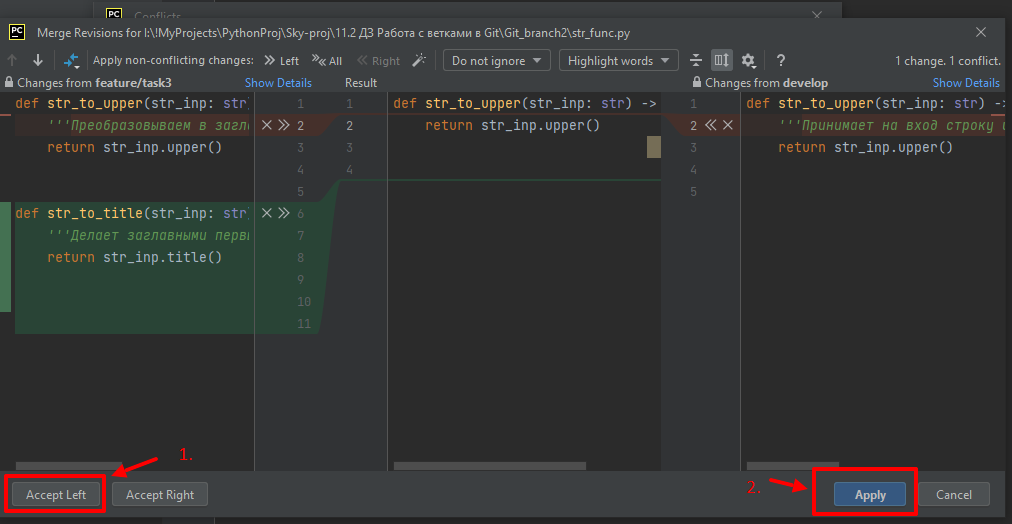
1. Сделайте мердж develop в текущую ветку, где вы ведете разработку (вам предложат порешать конфликт).



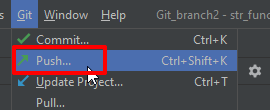
Изображение выглядит как текст

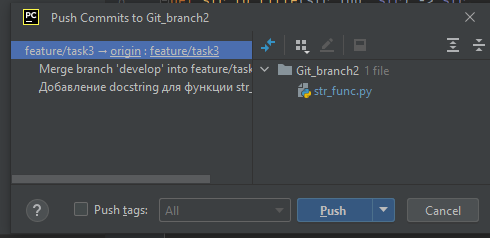
Автоматически созданное описание

1. Решите конфликт, отклонив описание функции другого программиста.



1. Сделайте пуш.





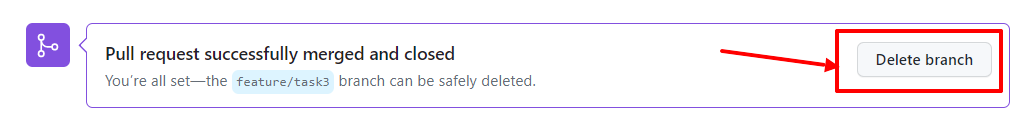
1. За тимлида заапрувьте пул-реквест на GitHub.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание





**\* Дополнительное задание**

Команды разработчиков постоянно коммитят новый функционал в ветку develop. Но периодически эти наработки надо выкатывать и конечным пользователям. Подготовленные для пользователей версии продуктов (релизы) коммитят в ветку main. Пользователи могут находить какие-то баги в приложении, эти баги быстро чинятся в ветке hotfix.

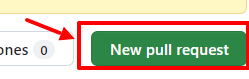
Сделайте мердж ветки develop в main. Из main сделайте checkout в hotfix ветку, сделайте в коде минимальное улучшение, сымитировав фикс бага, и закоммитьте всё обратно в main.

**Выполнение**

1. Сделайте мердж ветки develop в main.

Изображение выглядит как стрела

Автоматически созданное описание



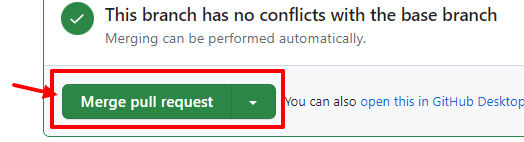
Изображение выглядит как стол

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

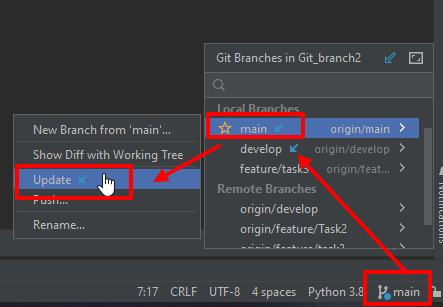
Автоматически созданное описание

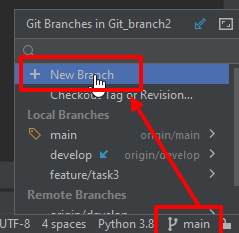


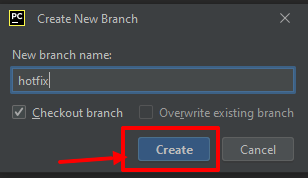
Изображение выглядит как текст

Автоматически созданное описание

1. Из main сделайте checkout в hotfix ветку







1. Сделайте в коде минимальное улучшение, сымитировав фикс бага

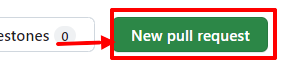
Изображение выглядит как текст

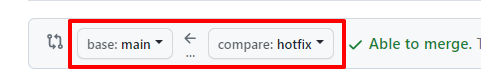
Автоматически созданное описание

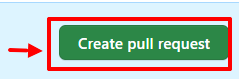
1. Закоммитьте всё обратно в main

Изображение выглядит как текст

Автоматически созданное описание







Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание