**1 вопрос**

## Процедурные объявления

Объявление процедуры сообщает компилятору об имени процедуры и о том, как вызвать процедуру. Фактическое тело процедуры может быть определено отдельно.

Объявление процедуры имеет следующий синтаксис –

procedure name(argument(s): type1, argument(s): type 2, ... );

Обратите внимание, что **название процедуры не связано ни с каким типом** . Для описанной выше процедуры *findMin ()* ниже приводится объявление –

procedure findMin(x, y, z: integer; var m: integer);

**2 вопрос**

## Определение процедуры

В Паскале процедура определяется с помощью ключевого слова **процедуры** . Общая форма определения процедуры следующая:

procedure name(argument(s): type1, argument(s): type 2, ... );

< local declarations >

begin

< procedure body >

end;

Определение процедуры в Pascal состоит из **заголовка** , локальных **объявлений** и **тела** процедуры. Заголовок процедуры состоит из ключевого слова **процедуры** и имени, присвоенного процедуре. Вот все части процедуры –

* **Аргументы** – Аргументы (аргументы) устанавливают связь между вызывающей программой и идентификаторами процедур, а также вызывают формальные параметры. Правила для аргументов в процедурах такие же, как и для функций.
* **Локальные объявления** – Локальные объявления относятся к объявлениям для меток, констант, переменных, функций и процедур, которые применимы только к основной части процедуры.
* **Тело процедуры** – **Тело** процедуры содержит набор операторов, которые определяют, что делает процедура. Он всегда должен быть заключен между зарезервированными словами начало и конец. Это часть процедуры, в которой выполняются все вычисления.

**3 вопрос**

**Объявление процедуры находится в части Interface модуля, а определение процедуры находится в части Implementation.**

**4 вопрос**

Запись – это структура, состоящая из фиксированного числа компонент, называемых полями.

**var** <имя\_записи> = **record**

<имя\_поля1>:<тип\_поля1>;

<имя\_поля2>:<тип\_поля2>;

...

**end**;

**5 вопрос**

Полями называют компоненты записи. Каждое поле определяется определенным именем. Поле записи включает имя поля, за которым через символ «:» указывают тип данного поля.

**6 вопрос**

Класс — это множество объектов, имеющих общую структуру и общее поведение. Класс — описание (абстракция), которое показывает, как построить существующую во времени и пространстве переменную этого класса, называемую объектом. Сам класс в итоге определяется как список своих членов, а именно полей (свойств) и методов/функций/процедур.

Различие Записи и Классов в их назначении. Запись нужна для структурирования разнородных типов данных для описания одного и того же объекта. Классы нужны для создания структур данных, но так же и для обработки этих данных обработки.

TMyClass = **class**(TObject)

private

*{Описанные в этой секции элементы не доступны извне (за пределами класса, но доступны в пределах модуля).}*

*{Здесь обычно находятся поля класса.}*

strict private

*{Для версии Delphi 2007 и выше. Описанные в этой секции элементы доступны только внутри класса}*

protected

*{Описанные в этой секции элементы доступны только классу и всем его потомкам.}*

public

*{Описанные в этой секции элементы доступны всем.}*

published

*{Описанные в этой секции элементы доступны всем и отображаются в Object Inspector'e.}*

**end**;

* TMyClass — имя класса;
* class — ключевое слово, начинающее определение класса (в старых версиях также было ключевое слово object);
* TObject — класс-предок, если есть [наследование](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D1%81%D0%BB%D0%B5%D0%B4%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5));
* private, protected, public, published — ключевые слова, определяющие иерархический доступ к полям и методам в виде обозначения секций [областей доступа](https://ru.wikipedia.org/w/index.php?title=%D0%9E%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%B0&action=edit&redlink=1).

**7 вопрос**

**Ме́тод** — это [функция](https://ru.wikipedia.org/wiki/%D0%A4%D1%83%D0%BD%D0%BA%D1%86%D0%B8%D1%8F_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) или [процедура](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%B0_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)), принадлежащая какому-то [классу](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D0%B0%D1%81%D1%81_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) или [объекту](https://ru.wikipedia.org/wiki/%D0%9E%D0%B1%D1%8A%D0%B5%D0%BA%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)).

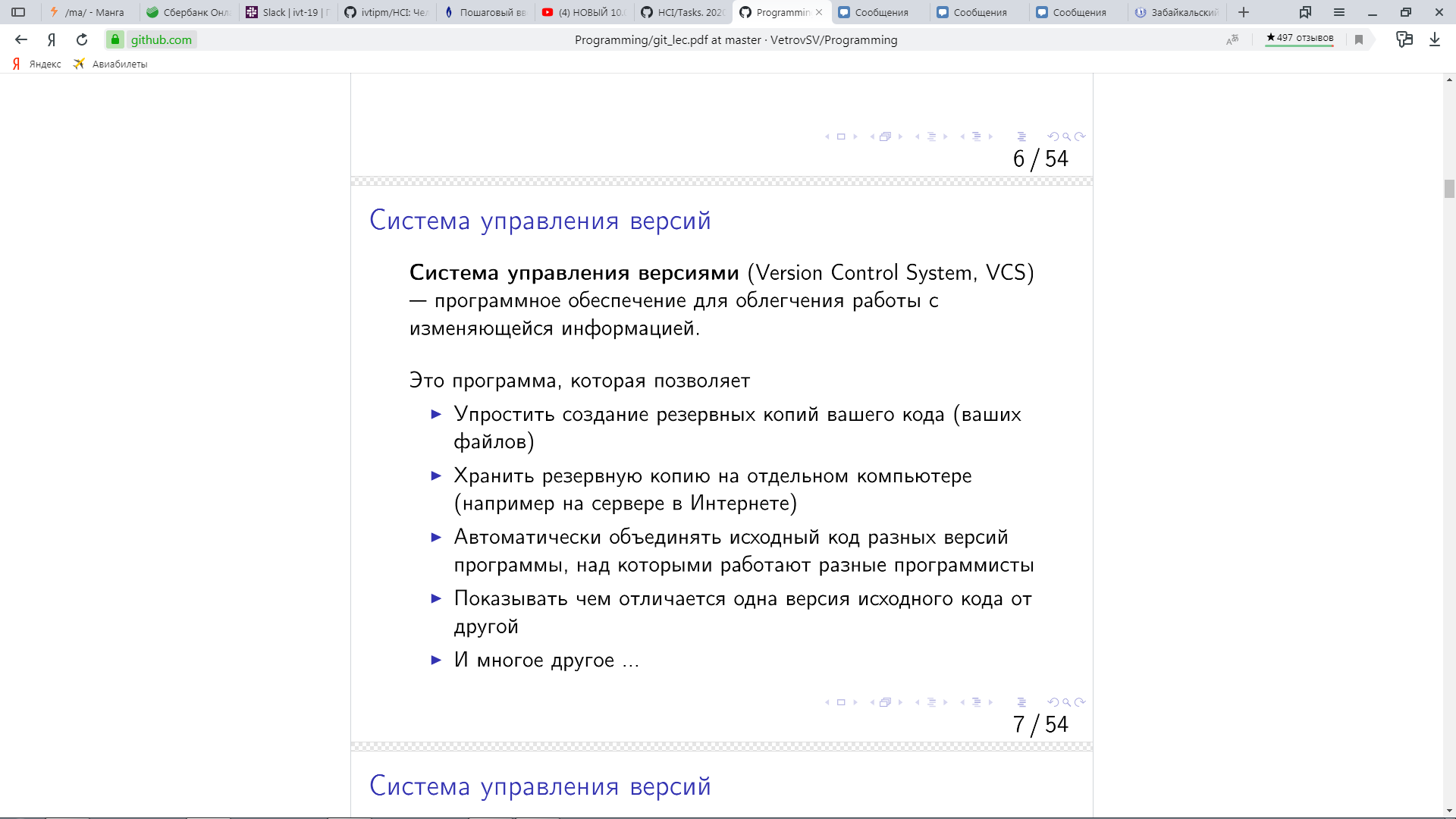
Как и процедура в [процедурном программировании](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), метод состоит из некоторого количества [операторов](https://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)) для выполнения какого-то действия и имеет набор [входных аргументов](https://ru.wikipedia.org/wiki/%D0%90%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5)).

**8 вопрос**

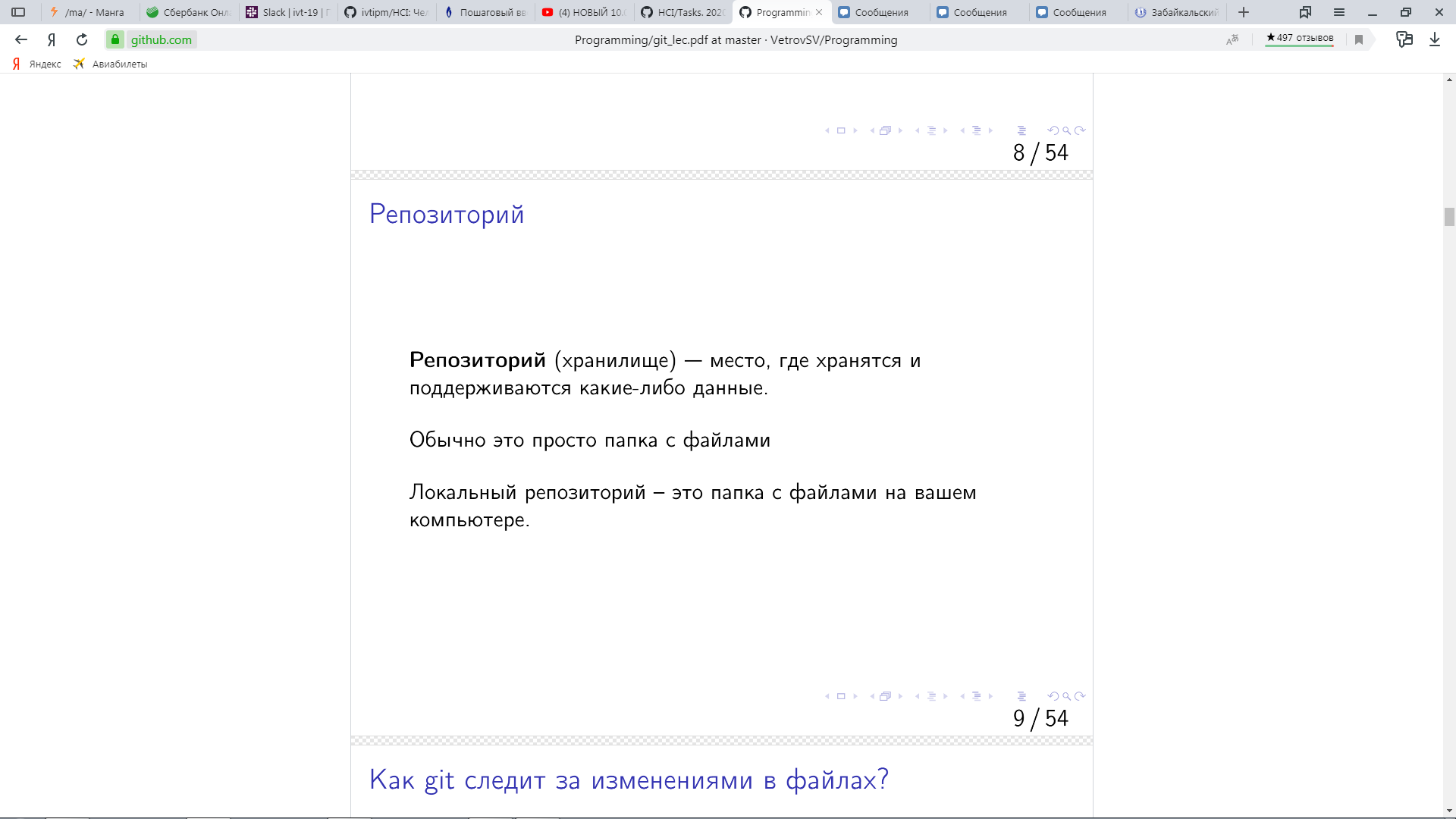
Объект - структура данных, которая содержит фиксированное число компонентов

**Вопросы по Git**

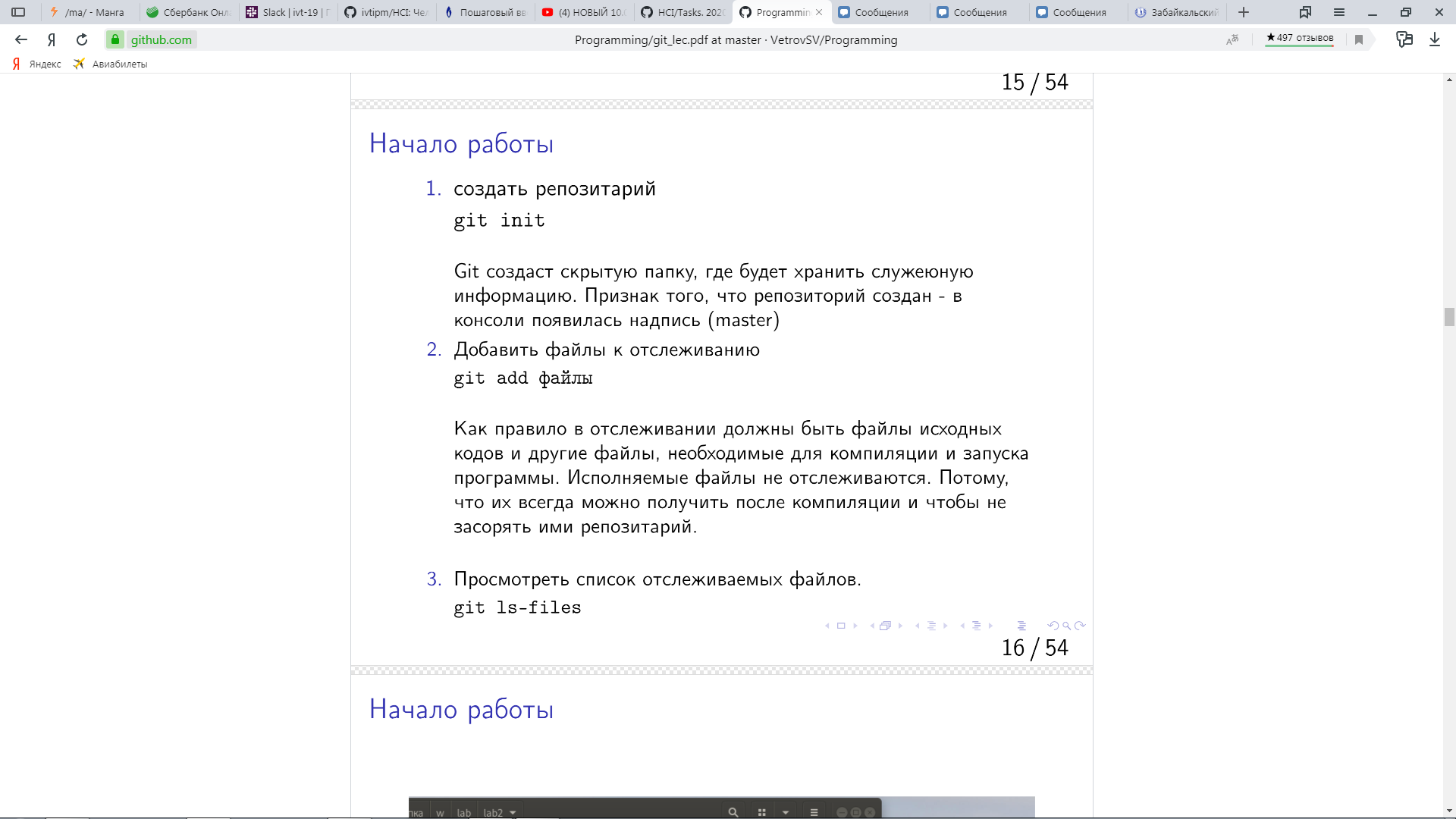
**1 вопрос**



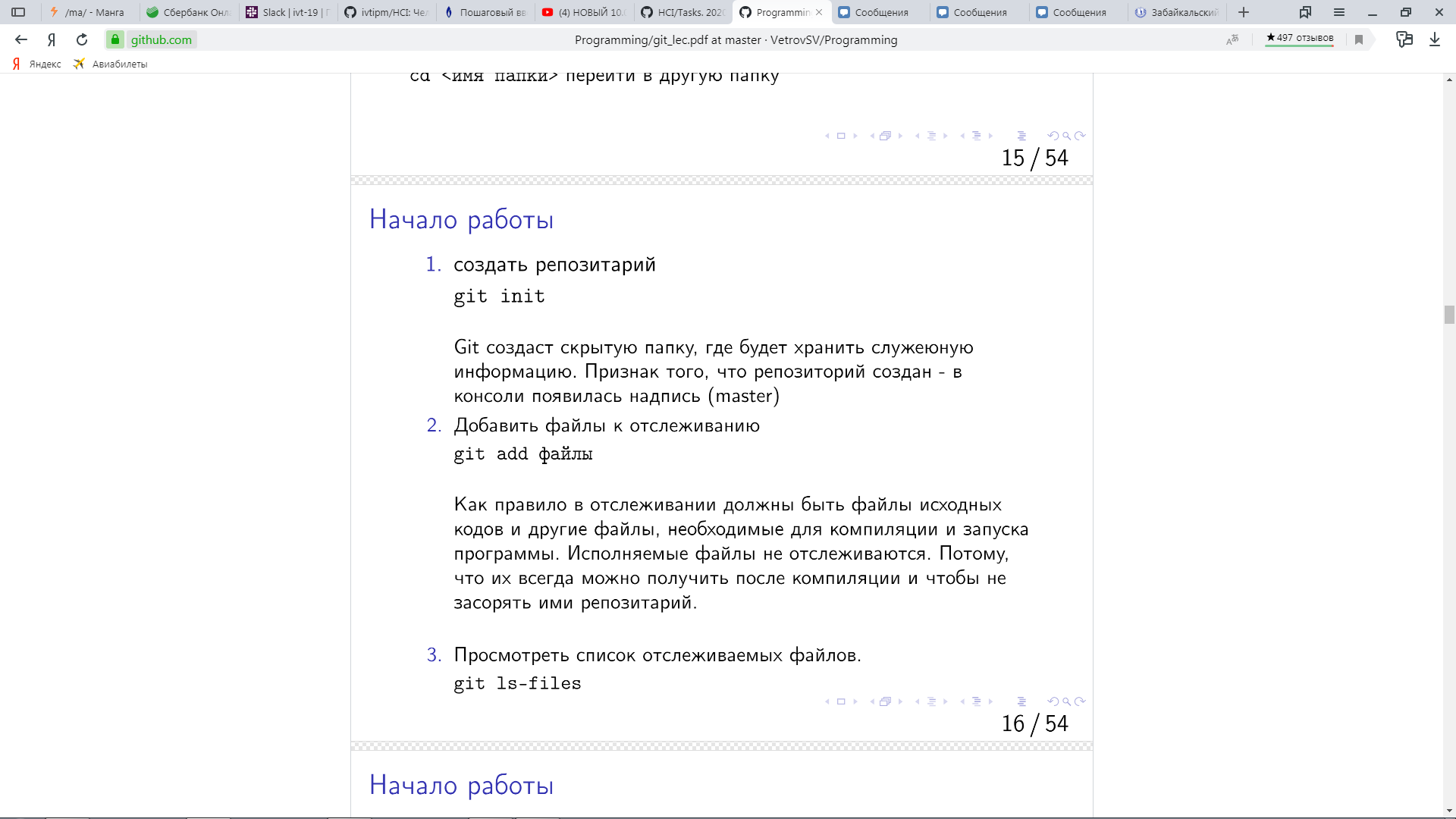
**2 вопрос**



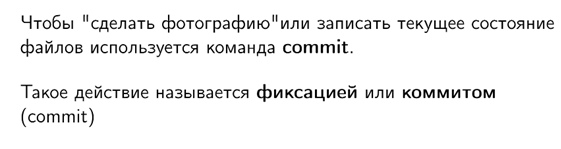
**3 вопрос**



**4 вопрос**



**5 вопрос**



Ответы по примеру программы  
1) В процедуре(DataFromForm) идёт одновременно запись и проверка вводимых данных, так же данные конвертируются из строкового типа в действительный тип.  
2) В таком случае вызовется процедура Show Message, с сообщением “Неправильно введён параметр…” и далее происходит прерывание процедуры(DataFromForm).