

# Инструкция по выполнению лабораторных работ (общая)

Соколов А.П.\*; Першин А.Ю.†  
кафедра «Системы автоматизированного проектирования»  
МГТУ им. Н.Э. Баумана

2021-09-28

## 1 Введение

Инструкция разработана для ознакомления с принципами выполнения лабораторных работ в дистанционном формате с использованием специализированных программных средств и сети интернет в рамках следующих курсов, проводимых на кафедре «Системы автоматизированного проектирования» МГТУ им. Н.Э. Баумана (с указанием соответствующих сборников постановок задач на лабораторные работы):

- «Вычислительная математика» (бакалавриат) [1];
- «Методы математического моделирования сложных процессов и систем» (магистратура) [2];
- «Управление проектированием информационных систем» (магистратура) [3].

Лабораторная работа является формой самостоятельного выполнения заданий, поставленных преподавателем.

Управление процессом реализации лабораторных работ включает необходимость решения следующих задач:

№	Задача	Метод реализации
1	Контроль за выполнением	согласно заранее определённым срокам
2	Представление к проверке	создание Merge Request в GitLab
3	Проверка подготовленных отчета и исходных кодов (защита результатов лабораторной работы)	peer-review
4	Хранение отчетов и реализованных исходных кодов	GitLab, Nextcloud

\* доцент РК-6, к.ф.-м.н., email: alsokolo@bmstu.ru

† ассистент РК-6, Ph.D., email: apershin@bmstu.ru

Хранение и организация отчетов о лабораторных работах и им соответствующих исходных кодов осуществляется с помощью системы контроля версий **git** и веб-приложения **GitLab**: <https://sa2systems.ru:88>. Доступ организован централизованным образом с использованием единой учетной записи студента, зарегистрированной в базе данных сервиса **OpenLDAP** администратором (учетные данные следует уточнять у сотрудников кафедры).

## 2 Общие требования к знаниям

Помимо специальных знаний, получаемых обучающимися на лекционных занятиях по соответствующей дисциплине, для выполнения лабораторных работ обучающийся должен обладать дополнительными знаниями:

- приветствуются знания языка вёрстки  $\text{\LaTeX}$ ;
- знать принципы работы системы контроля версий **git**;
- владеть навыками использования системы контроля версий **git** (уметь применять команды `git clone`, `git checkout`, `git add`, `git commit`, `git pull`, `git push` и пр.).

## 3 Общие принципы выполнения заданий

Разработку документов и исходных кодов программ, реализующих те или иные программные модули, функции, численные методы, следует осуществлять с использованием системы контроля версий **Git**.

Все лабораторные работы следует выполнять с использованием удалённого репозитория, предназначенного для хранения материалов по соответствующей дисциплине (табл. 1).

Таблица 1: Адреса репозиториев

Дисциплина	@cpxsln@	URL
«Вычислительная математика» (бакалавриат)	educmm	<a href="https://sa2systems.ru:88/edu/educmm.git">https://sa2systems.ru:88/edu/educmm.git</a>
«Методы математического моделирования сложных процессов и систем» (магистратура)	edumms	<a href="https://sa2systems.ru:88/edu/edumms.git">https://sa2systems.ru:88/edu/edumms.git</a>
«Управление проектированием информационных систем» (магистратура)	edupmi	<a href="https://sa2systems.ru:88/edu/edupmi.git">https://sa2systems.ru:88/edu/edupmi.git</a>
«Модели и методы анализа проектных решений» (бакалавриат)	edumma	<a href="https://sa2systems.ru:88/edu/edumma.git">https://sa2systems.ru:88/edu/edumma.git</a>

Перед выполнением лабораторных работ по определённой дисциплине соответствующий репозиторий следует клонировать, выполнив команду: `$git clone <repo _url>` (выполняется один раз).

Обновление загруженных материалов следует осуществлять с применением команды `$git pull`

Для отслеживания изменений, управления процессом разработки, а также для сдачи результатов выполнения следует использовать систему **GitLab**, развернутую по адресу: <https://sa2systems.ru:88>.

После размещения финальных версий документов (и исходных кодов программ) в своей ветке разработки следует выполнить команду `merge request`, что приведёт к уведомлению преподавателя(ей) о необходимости осуществить проверку. Доступ к **GitLab** осуществляется с использованием единой учетной записи обучающегося: **логин: rk6\_student** **пароль: 2afeu33f**.

**Замечание 1** Студенческий доступ к прочим сервисам, включая облачный сервис кафедры

<https://archrk6.bmstu.ru>, также осуществляется с использованием этой учетной записи.

**Для каждой реализуемой лабораторной работы должны быть разработаны:**

1. исходные коды программы, реализующей решение поставленной задачи;
2. отчет о проведенной лабораторной работе согласно шаблону, размещенному в облачном хранилище кафедры (табл. 2).

Таблица 2: Доступные шаблоны отчетов о ЛР

Шаблон в формате .doc	<a href="https://archrk6.bmstu.ru/index.php/f/85083">https://archrk6.bmstu.ru/index.php/f/85083</a>
Шаблон в формате L <sup>A</sup> T <sub>E</sub> X	<a href="https://archrk6.bmstu.ru/index.php/f/844352">https://archrk6.bmstu.ru/index.php/f/844352</a>

### 3.1 Требования к именам. Форматы идентификаторов

Листинг 1: Формат имени **ветки** (branch) разработки обучающегося (**НЕ** совпадает с именем **запроса на слияние**)

```
1 @student_branch_sid@ = @year@_@group@_@surname@_@fname@
```

где @year@ – текущий год; @group@ – идентификатор учебной группы (английскими буквами, регистр нижний, например: rk6\_62b, rk6\_12m); @surname@ – фамилия (английскими буквами, регистр нижний); @fname@ – первые буквы имени и отчества (английскими буквами, регистр нижний).

Например: 2021\_rk6\_52b\_ivanovii.

### Листинг 2: Формат имени запроса на слияние (Merge Request)

```
1 @merge_request_sid@ = @year@_@group@_@surname@@@fname@_(  
    @semid@ | @labid@ | @exmid@)  
2 @semid@ = sem@i@_tsk@j@  
3 @labid@ = lab@k@  
4 @exmid@ = exm@l@
```

где @i@ – порядковый номер семинара, @k@ – порядковый номер лабораторной работы, @j@ – порядковый номер представляемой задачи в рамках семинара @i@ согласно заранее предоставленному сборнику задач, @l@ – порядковый номер попытки сдачи экзамена, остальные обозначения были введены выше; символ ‘|’ обозначает альтернативный выбор одного из доступных значений, представленных в круглых скобках.

Например: 2020\_rk6\_62b\_ivanovii\_sem1\_tsk6, 2021\_rk6\_52b\_petrovii\_sem4\_tsk2, 2021\_rk6\_52b\_petrovii\_sem6 (если сдаётся несколько задач по семинару 6), 2021\_rk6\_52b\_petrovii\_sem4\_tsk10\_lab3, 2022\_rk6\_52b\_sidorovaa\_lab1 и т.п.

**Замечание 2** Важной особенностью системы *GitLab* является невозможность одновременного создания нескольких *Merge Request (MR)* из одной и той же ветки. Поэтому в случае одновременного выполнения нескольких заданий (в рамках семинарских и лабораторных работ) будет возможным создать только один *MR*. В указанном случае постфикс *Merge Request* будет различным (см. примеры выше).

### Листинг 3: Формат имени файла, содержащего решение конкретной задачи семинарского занятия, отчет о лабораторной работе или ответ на билет экзамена

```
1 @report_filename@ =  
    @cpxsl@_@tml@_@year@_@group@_@surname@@@fname@_(  
    @semid@ | @labid@ | @exmid@).pdf  
2 @cpxsl@ = ( educmm | edumms | edupmi | edumma | ... )  
3 @tml@ = ( sem | lab | exm)
```

где @cpxsl@ – идентификатор дисциплины; @tml@ – идентификатор типа задания (для семинаров всегда sem).

Например: educmm\_sem\_2021\_rk6\_53b\_sidorovas\_sem5\_tsk6.pdf; educmm\_lab\_2021\_rk6\_53b\_sidorovas\_lab3.pdf; educmm\_lab\_2021\_rk6\_53b\_sidorovas\_lab3.pdf  
textsf

## 3.2 Структура каталогов для размещения результатов выполнения

В процессе выполнения лабораторных работ обучающимся формируются различные документы среди которых: исходные коды разработанных программ, отчет о выпол-

ненной лабораторной работе. Все документы следует размещать в git-репозитории соответствующей дисциплины с использованием единой структуры каталогов (рисунок 1).

Конкретные имена представленных файлов в подкаталогах (рисунок 1) не формализованы и могут быть названы произвольным образом.

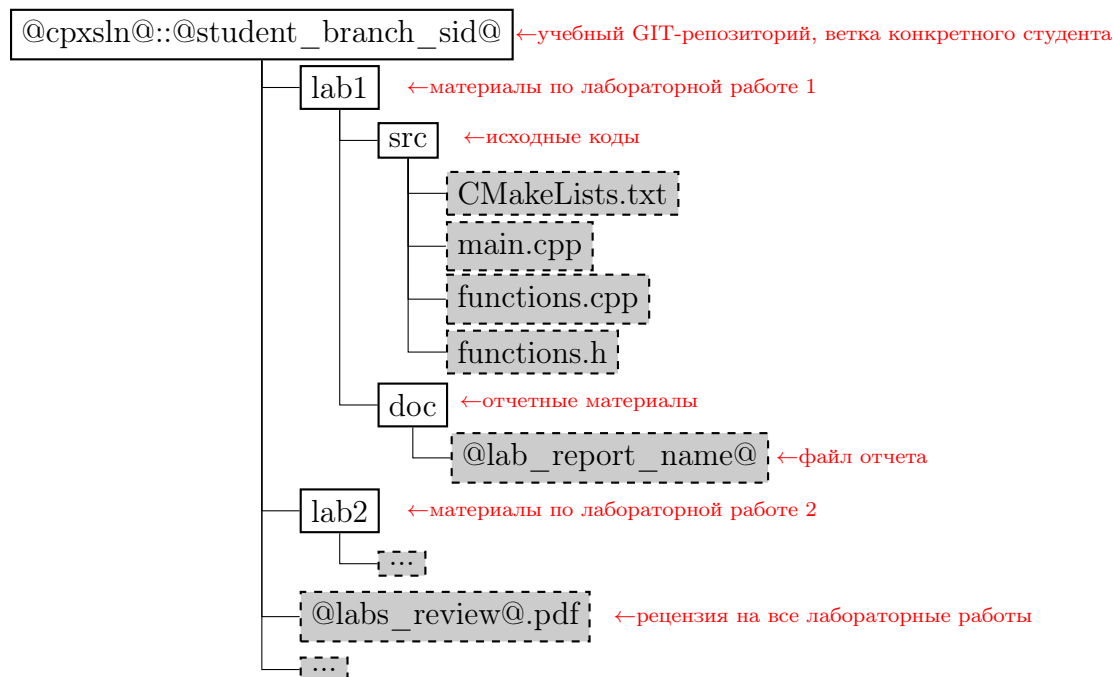


Рис. 1: Структура файловой системы при выполнении лабораторных работ

Пример размещения файлов представлен на рис. 2.

## 4 Принципы выполнения лабораторной работы

Следующий алгоритм предполагает, что удаленный Git-репозиторий был клонирован и размещен локально.

---

### Алгоритм 1 Выполнение лабораторной работы (осуществляет студент)

---

- 1: Создать свою git-ветку разработки и присвоить ей имя в формате согласно листингу 1<sup>1</sup>.
  - 2: Разместить (обновить после отладки) материалы лабораторной работы в отдельном каталоге с использованием структуры каталогов согласно рисунку 1<sup>2</sup>.
  - 3: Уведомить преподавателя о готовности лабораторной работы к сдаче, создав запрос на слияние merge request<sup>3</sup> из своей ветки в master ветку с использованием системы GitLab и обязательно создать комментарий о необходимости провести проверку.
-

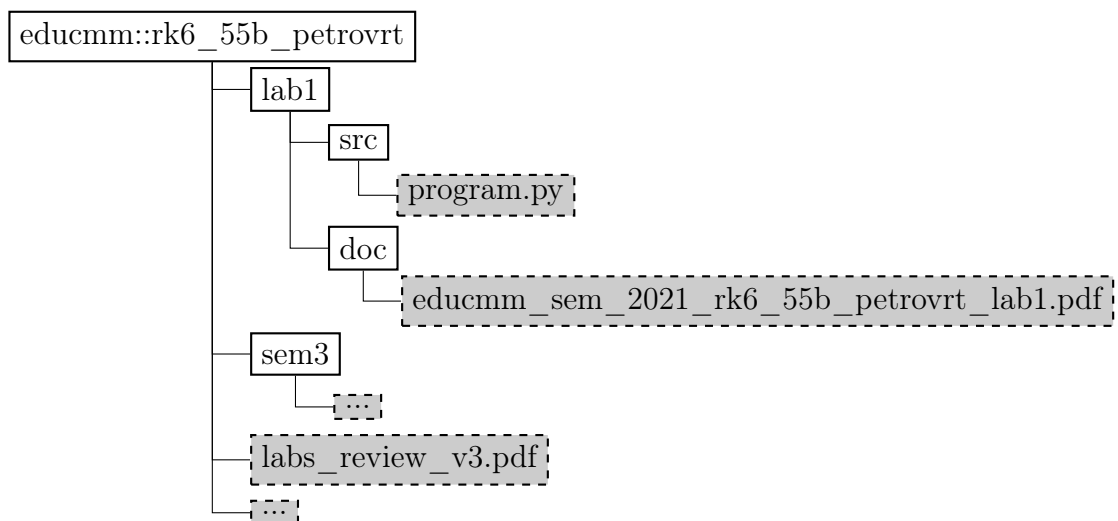


Рис. 2: Пример файловой структуры: размещение отчета о лабораторной работе 1, а также соответствующего исходного кода к ней, студентом группы РК6-55Б Петровым Р.Т., выполнившим данное задание в 2021 году в рамках дисциплины «Вычислительная математика»

Выполнение лабораторных работ возможно не по порядку, все выполняемые лабораторные работы могут быть привязаны к одному запросу *merge request*.

Рекомендуется при каждой операции *commit* указывать автора, который выполняет эту операцию, а также дату операции.

## 5 Алгоритм проведения проверки лабораторной работы

Алгоритм проведения проверки поэтапный. После того как студент загрузил **первую** версию отчетных материалов, включая: отчет и исходные коды созданных программ, а также сформировал *merge request*, преподаватель начинает проверку.

---

**Алгоритм 2** Проверка лабораторной работы (осуществляет преподаватель)

---

- 1: Преподаватель изучает представленный на проверку материал и формирует рецензию на *первую* версию отчета, заполняя стандартную форму.
  - 2: В процессе проверки заполняется интерактивная форма рецензии<sup>4</sup>. При ответе на ряд вопросов в рецензии автоматически выставляется оценка (по 100 балльной шкале).
  - 3: Полученному баллу, также автоматически, ставятся в соответствие еще две оценки:  
а) балл от 0 до  $\left] \frac{56}{4} \right]$ , формирующий финальный накопленный балл за семестр нарастающим итогом; б) оценка по 5-и балльной шкале, обеспечивающая автоматический ответ на вопрос о защите (**удовл.**, **хор.**, **отл.**) или не защите (**неуд.**) лабораторной работы.
  - 4: Данная рецензия сохраняется вместе с первой версией отчета путём её загрузки в репозиторий с использованием системы контроля версий.
- 

**Замечание 3 (Важные особенности проведения проверки первой лабораторной)** 1.  
*Преподаватель должен проверить **только** первую ЛР.*

2. *Лабораторная работа считается защищённой, если была получена оценка **удовл.**, **хор.** или **отл.**.*
3. *Если студента не устраивает выставленная оценка, то студент вправе внести корректировки в предоставленные материалы и попросить провести проверку повторно, однако, проверка будет в зависимости от следующих обстоятельств:*
  - *получена **положительная** оценка: на усмотрение преподавателя и при наличии соответствующей возможности.*
  - *получена **неудовлетворительная** оценка: лабораторную работу следует переделать и далее будет проведена повторная проверка.*

В качестве проверяющих могут выступать: студенты этого же потока, магистры и аспиранты.

Детальные требования к отчетным материалам по лабораторным работам представлены в интерактивной форме рецензии и могут корректироваться от года к году.

## 6 Проверка лабораторной работы в формате peer-review (экспертная оценка равными)

По итогам первой лабораторной работы первые несколько студентов выбираются в качестве помощников преподавателя для проверки следующей лабораторной. Данные студенты освобождаются от выполнения работы. Оставшиеся студента равномерно случайным образом распределяются по выбранным помощникам-проверяющим. В течение недели выбранные помощники должны проверить все сданные работы.

---

### Алгоритм 3 Проверка лабораторной работы (осуществляет выбранным помощником-бакалавром)

---

- 1: Проверяющий изучает представленный на проверку материал и формирует *рецензию* на *первую* версию отчета, заполняя стандартную форму рецензии (<https://archrk6.bmstu.ru/index.php/f/788873>).
  - 2: Для осуществления проверки *заполняется интерактивная форма* рецензии<sup>5</sup>, путём ответов на ряд вопросов в рецензии автоматически выставляется оценка (по 100 балльной шкале). В соответствии полученному баллу определяется оценка по 5-и балльной шкале. В зависимости от качества реализации, оценок, полученных по множеству критериев, срока сдачи и т.д. за текущую лабораторную работу выставляется финальная оценка (зачет/незачет) и число баллов, которые будут учитываться при сдаче экзамена.
  - 3: Данная рецензия сохраняется вместе с первой версией отчета путём её загрузки в репозиторий с использованием системы контроля версий.
- 

После преподаватель проверяет некоторое число работ. Некоторое количество лучших студентов из числа выполнивших работу назначаются проверяющими для следующей лабораторной. Оценка проверяющим выставляется на основании качества проверки работ. Сравниваются оценки, выставленные преподавателем и помощником-проверяющим. Если разность баллов превышает 10, то оставшиеся баллы умножаются на поправочный коэффициент и вычитаются из 100. Коэффициент вычисляется как  $2 / [\text{количество перепроверенных работ}]$ . Например: проверяющий выставил 85 баллов некоторому студенту, а преподаватель, перепроверивший данную лабораторную выставил 60. Разность между выставленными баллами равно 25, коэффициент равен 2. 25 на 15 больше десяти. В таком случае оценка проверяющего равна  $100 - 2 * 15 = 70$ . Другой пример: проверяющий выставил 80 и 65, а преподаватель - 85 и 80 соответственно. Разности баллов равны 5 и 15 баллов соответственно, коэффициент равен 1. В таком случае оценка проверяющего равна  $100 - 1 * 5 = 95$ .

Если студент набрал более 60-и баллов, то лабораторная работа считается защищённой. Если студента не устраивает выставленная оценка более 60-и баллов, то он вправе внести корректировки в предоставленные материалы и попросить провести проверку повторно (на усмотрение преподавателя и при наличии соответствующей возможности преподаватель может осуществить повторную проверку).

Если студент не набрал 60-и баллов, то работу следует доработать.

## 7 Установка Git системы контроля версий и Git-клиента

Для работы с удалённым Git репозиторием на локальной машине должно быть установлено программное обеспечение:

1. Git (система контроля версий, серверная часть)<sup>6</sup>: <https://git-scm.com/>.

---

<sup>6</sup>При работе в операционных системах Linux установка осуществляется командой: `$yum install`



2. Git клиент (для работающих под операционной системой Windows): <https://tortoisegit.org>.

## Список литературы

- [1] Першин А.Ю., Соколов А.П. Сборник постановок задач на лабораторные работы по курсу «Вычислительная математика»: Учебное пособие. [Электронный ресурс]. Москва, 2021. С. 54.
- [2] Соколов А.Ю. Сборник постановок задач на лабораторные работы по курсу «Методы математического моделирования сложных процессов и систем»: Учебное пособие. [Электронный ресурс]. Москва, 2021. С. 26.
- [3] Соколов А.П. Сборник постановок задач на лабораторные работы по курсу «Управление проектированием информационных систем»: Учебное пособие. [Электронный ресурс]. Москва, 2021. С. 25.

## А Частые ошибки

**Частая ошибка 1** *Создание файлов с именем “aix”, в том числе файлов, имена которых включают эти три буквы так, чтобы сразу после них следовала точка.*

### **Комментарий**

*В операционных системах семейства Windows запрещается создание файлов с такими именами, поэтому при размещении такого файла в git-репозитории с использованием, положим, операционной системы семейства Linux, такие данные будет невозможно клонировать/обновить в Windows.*

**Частая ошибка 2** *Создание очень длинных имён файлов или имён каталогов.*

### **Комментарий**

*В операционных системах семейства Windows есть ограничение на создание файлов, полный путь которых содержит более 260 символов. В то же время файловая система NTFS разрешает создавать такие файлы и каталоги, **но только программно**. В операционных системах семейства Linux и Mac OS таких ограничений нет, как и в используемых ими файловых системах. Таким образом, не следует создавать длинные имена файлов и каталогов и загружать их в git-репозитории. Такие файлы и каталоги будут удаляться без проведения проверки.*

---

git (CentOS, Red Hat), `$apt-get install git` (Debian, Ununtu) и т.д. Одновременно устанавливается git-клиент, работающий из командной строки.