

Министерство образования Республики Беларусь
Учреждение образования «Полоцкий государственный университет
имени Евфросинии Полоцкой»

Кафедра технологий программирования

О Т Ч Е Т

о прохождении учебной ознакомительной практике
студента 2 курса группы 21-ИТ-1
Макеёнка Дениса Игоревича
(фамилия, имя, отчество)

в период с 20.06.2022 по 03.07.2022

Наименование базы практики:

Учреждение образования
«Полоцкий
государственный университет имени
Евфросинии Полоцкой»

Руководитель от кафедры:
(ученая степень, звание)

Гурьева Н. А.
Кандидат физико-математических
наук, доцент кафедры технологий
программирования

Практика защищена
с оценкой _____

Дата защиты _____

Полоцк 2022

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ «ПОЛОЦКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ ЕВФРОСИНИИ ПОЛОЦКОЙ»

ФАКУЛЬТЕТ
ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ
УТВЕРЖДАЮ:

Зав. кафедрой технологий программирования

_____ В.М. Чертков

«___» _____ 2022 г.

ЗАДАНИЕ
по учебной ознакомительной практике

для специальности 1-40 01 01 «Программное обеспечение информационных технологий»

Студенту Макеёнок Денис Игоревич

(фамилия, имя, отчество)

Группы 21-ИТ-1

Предприятие (организация) Полоцкий государственный университет имени
Евфросинии Полоцкой

- 1. Тема практики (работы):** учебная ознакомительная практика
- 2. Срок сдачи студентом законченной работы:** 14.09.2022 г.
- 3. Исходные данные к работе:** Справочная документация к ПО и средствам программирования
- 4. Содержание работы по основной части**
 - 4.1. Знакомство с IT-индустрией на примере компании «EPAM Systems»:** история развития, общие сведения о компании; организационная структура компании
 - 4.2 Изучение программного обеспечения, используемого в учебной практике:** анализ используемых программных продуктов и технологий; среды разработки и языки программирования
 - 4.3 Индивидуальное задание:** Прохождение курсов на платформах learn.epam.com, training.by, обучение системе управления версий Git.
- 5. Форма и вид материалов, представляемых по окончании практики:** направление (дневник) на практику, письменный отчет о выполнении практики.

Дата выдачи задания 20.06.2022 г.

Задание принял к исполнению

Студент Макеёнок Д.

(подпись)

Руководитель практики

от кафедры Гурьева Н.А.

(подпись)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 ХАРАКТЕРИСТИКИ ПРЕДПРИЯТИЯ	6
1.1 История развития, общие сведения о предприятии	6
1.2 Организационная структура организации	7
2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ	8
2.1 Что такое система контроля?	8
2.2 Зачем нужна система контроля?	8
2.3 Выбор и обоснование средств разработки	9
3 ПРАКТИЧЕСКАЯ ЧАСТЬ	11
3.1 Постановка цели	11
3.2 Этапы работы	11
ЗАКЛЮЧЕНИЕ	15
СПИСОК ИСТОЧНИКОВ	16

					<i>МДИ.21114.00.ПЗ</i>			
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Подпись</i>	<i>Дата</i>				
<i>Разраб.</i>		<i>Макеёнок Д.И.</i>			<i>Отчет по учебной ознакомительной практике.</i>	<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Провер.</i>		<i>Гурьева Н.А.</i>					4	16
<i>Т. Контр.</i>						<i>Учреждение образования "Полоцкий Государственный университет имени Евфросинии Полоцкой" гр.21-ИТ-1</i>		
<i>Реценз.</i>								
<i>Н. Контр.</i>								
<i>Утверд.</i>								

ВВЕДЕНИЕ

Практика проходила в Полоцком государственном Университете имени Евфросинии Полоцкой от компании ЕРАМ. Компанией был предложен новый формат летней практики – IT Warm-up, где нам в live – сессиях рассказывали про историю IT-компании Ерам, а также погружали нас в разные направления такие как: Платформа Java, front-end разработка, Автоматизированное тестирование, C# и платформа .NET, Cloud & DevOps, Функциональное тестирование, Data & Analytics.

Целью практик было ознакомление с кампанией и её основными структурами. Развитие личных качеств и получение практических навыков в своей профессии, прохождение курсов и выбор дальнейшего направления работы.

Для успешного прохождения практики надо было ознакомиться с системами контроля, выбрать самую удобную и изучить основы работы с ней для дальнейшего использования.

Заданием практики было прохождение курсов на платформах learn.eram.com, training.by, обучение системе управления версий Git.

					МДИ.21114.00.ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

1.ХАРАКТЕРИСТИКИ ПРЕДПРИЯТИЯ

1.1 История развития, общие сведения о предприятии

Компания ЕРАМ была основана в 1993 году двумя одноклассниками Аркадием Добкиным и Леонидом Лознером. Название компании происходило от «Effective Programming for America». Первые офисы были открыты в США и Белоруссии. Позже были открыты офисы в Австралии, Болгарии, Великобритании, Германии, Канаде, Китае, Мексике, Нидерландах, России, Швеции, Швейцарии и других странах.

В 1993 году Добкин открыл компанию ЕРАМ и хотели доказать, что талант советского разработчика нужен западному миру. ЕРАМ создавалась как программная инжиниринговая компания для построения больших решений и помощи корпорациям. 27 лет назад никто не понимал, зачем строить и продавать собственные продукты. Первым крупным клиентом ЕРАМ стал модный швейцарский бренд Bally. Вторая удача – сделка с Colgate-Palmolive. Следующие десять лет они помогали технологическим и «программистским» компаниям создавать софт. Их фокус – инжиниринг и дизайн элементов продукта. Это была хорошая школа. Они научились понимать проблемы бизнеса и индустрии.

К началу 2000 года в ЕРАМ работали 200–300 человек – костяк компании. У них было два пути: или переходить в семейную историю, зарабатывая деньги для личной свободы, или рисковать, двигая большие компании. Они выбрали второй вариант – выходить на мировой рынок. На пути к цели ЕРАМ пережил кризисы: крах доткомов и спад после терактов 11 сентября 2001 года. Несмотря на это, в 2004 году число сотрудников выросло до 400, а выручка увеличилась до \$3 млн.

В конце 2000-х произошла цифровая революция. На рынок вышли новые игроки – Amazon, Google и eBay. Они захватывали рынки, предлагая гибкие решения под запрос аудитории. Большие корпорации проигрывали – они не умели привлекать и удерживать клиентов. Корпорации нуждались в новом софте, других партнерах и внутренних ИТ-департаментах. Они пришли к ЕРАМ за уникальными гибкими системами, которые можно перестраивать и масштабировать. Построение платформы e-commerce для ритейлера занимает несколько лет и стоит \$10–50 млн. Потом ее нужно поддерживать, потому что Amazon, Google, eBay постоянно обновляются. С новыми клиентами компания начала расти на 40–50% в год.

К 2012 году ЕРАМ расширил штат до 7000 человек и вышел на IPO на Нью-Йоркскую фондовую биржу. 95% доходов приходило из Северной Америки и Западной Европы, где работали всего 2% сотрудников. Тогда они поставили новую цель – повышать компетенции. Мы развивали разработку, консалтинг, дизайн, чтобы охватить полный цикл работ в разных категориях. За восемь лет компания расширила штат в Америке и Европе до 11–12% от общего числа

					МДИ.21114.00.ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

сотрудников. Раньше 80% заказов были от продуктовых и технологических компаний, сейчас 30-35% – продуктовые и 65% – глобальные корпорации.

1.2 Организационная структура организации

В настоящее время штат насчитывает более 58,800 сотрудников, а каждый из новых специалистов принес в команду свой уникальный опыт и наработки.

Выполняя проекты для крупнейших корпораций, и сотрудничая с ведущими мировыми разработчиками программного обеспечения, EPAM Systems приобрела уникальный опыт в таких областях как:

- разработка по заказам крупнейших производителей ПО программного обеспечения для систем корпоративного планирования (ERP), управления жизненным циклом изделий (PLM); корпоративных информационных порталов (EIP), систем управления отношениями с клиентами (CRM), серверов интеграции приложений (EAI), систем управления контентом (CMS), систем управления знаниями (KMS);

- разработка приложений, соответствующих требованиям новейших сервис-ориентированных архитектур (SOA -- service oriented architecture);

- создание и развертывание электронных систем управления закупками и сбытом;

- построение порталов крупных предприятий и холдингов с развитыми средствами анализа данных и управления знаниями;

- интеграция приложений в распределенных системах (в том числе насчитывающих сотни производственных площадок, сотни унаследованных приложений и десятки ERP-систем), проектирование, консолидация и настройка корпоративных справочников и каталогов;

- внедрение ERP, PLM, CRM, SCM решений и систем аналитики, стратегического планирования и бюджетирования в ряде отраслей;

- анализ инфраструктуры и информационных ресурсов, проектирование и реинжиниринг бизнес-процессов, управление проектами модернизации и развития информационных систем.

					МДИ.21114.00.ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

2 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

2.1 Что такое система контроля?

Выбрать и научиться пользоваться самой лучшей системой контроля. Контроль версий, также известный как управление исходным кодом, - это практика отслеживания изменений программного кода и управления ими. Системы контроля версий - это программные инструменты, помогающие командам разработчиков управлять изменениями в исходном коде с течением времени. В свете усложнения сред разработки они помогают командам разработчиков работать быстрее и эффективнее. Системы контроля версий наиболее полезны командам DevOps, поскольку помогают сократить время разработки и увеличить количество успешных развертываний.

Программное обеспечение контроля версий отслеживает все вносимые в код изменения в специальной базе данных. При обнаружении ошибки разработчики могут вернуться назад и выполнить сравнение с более ранними версиями кода для исправления ошибок, сводя к минимуму проблемы для всех участников команды.

2.2 Зачем нужна система контроля?

Практически во всех программных проектах исходный код является сокровищем: это ценный ресурс, который необходимо беречь. Для большинства команд разработчиков программного обеспечения исходный код - это репозиторий бесценных знаний и понимания проблемной области, которые они скрупулезно собирали и совершенствовали. Контроль версий защищает исходный код от катастрофических сбоев, от случайных ухудшений, вызванных человеческим фактором, а также от непредвиденных последствий.

Разработчики программного обеспечения, работающие в командах, постоянно пишут новый исходный код и изменяют существующий. Код проекта, приложения или программного компонента обычно организован в виде структуры папок или «дерева файлов». Один разработчик в команде может работать над новой возможностью, а другой в это же время изменять код для исправления несвязанной ошибки, то есть каждый разработчик может вносить свои изменения в несколько частей дерева файлов.

Контроль версий помогает командам решать подобные проблемы путем отслеживания каждого изменения, внесенного каждым участником, и предотвращать возникновение конфликтов при параллельной работе. Изменения, внесенные в одну часть программного обеспечения, могут быть не совместимы с изменениями, внесенными другим разработчиком, работавшим параллельно. Такая проблема должна быть обнаружена и решена согласно регламенту, не

					МДИ.21114.00.ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

создавая препятствий для работы остальной части команды. Кроме того, во время разработки программного обеспечения любое изменение может само по себе привести к появлению новых ошибок, и новому ПО нельзя доверять до тех пор, пока оно не пройдет тестирование.

2.3 Выбор и обоснование средств разработки

Mercurial

Распределенная система контроля версий Mercurial разрабатывалась Мэттом Макалом. Первоначально, она была создана для эффективного управления большими проектами под Linux'ом, а поэтому была ориентирована на быструю и надежную работу с большими репозиториями. На данный момент mercurial адаптирован для работы под Windows, Mac OS X и большинство Unix систем.

Большая часть системы контроля версий написана на языке Python, и только отдельные участки программы, требующие наибольшего быстродействия, написаны на языке Си.

Из достоинств: Быстрая обработка данных, кроссплатформенная поддержка, возможность работы с несколькими ветками проекта, Простота в обращении.

Вывод: Простой и отточенный интерфейс, и набор команд, возможность импортировать репозитории с других систем контроля версий, - сделают переход на Mercurial и обучение основным особенностями безболезненным и быстрым.

Git

Git – это гибкая, распределенная (без единого сервера) система контроля версий, дающая массу возможностей не только разработчикам программных продуктов, но и писателям для изменения, дополнения и отслеживания изменения «рукописей» и сюжетных линий, и учителям для корректировки и развития курса лекций, и администраторам для ведения документации, и для многих других направлений, требующих управления историей изменений.

У каждого разработчика, использующего Git, есть свой локальный репозиторий, позволяющий локально управлять версиями. Затем, сохраненными в локальный репозиторий данными, можно обмениваться с другими пользователями.

Достоинства: Гибкая система ветвления проектов и слияния веток между собой. Наличие локального репозитория, содержащего полную информацию обо всех изменениях, позволяет вести полноценный локальный контроль версий и заливать в главный репозиторий только полностью прошедшие проверку изменения. Высокая производительность и скорость работы. Удобный и интуитивно понятный набор команд. Множество графических оболочек,

					МДИ.21114.00.ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

позволяющих быстро и качественно вести работы с Git'ом. Широкая распространенность, легкая доступность и качественная документация.

Вывод: Git – гибкая, удобная и мощная система контроля версий, способная удовлетворить абсолютное большинство пользователей. Если вы ведете большой проект, территориально удаленный, и тем более, если часто приходится разрабатывать программное обеспечение, не имея доступа к другим разработчикам. Git – один из лидеров систем контроля версий.

По этим причинам был выбран Git для практического задания.

					МДИ.21114.00.ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

3 ПРАКТИЧЕСКАЯ ЧАСТЬ

3.1 Постановка цели

В период прохождения практики было решено изучить Git. Для понимания работы с самой распространённой системой контроля и дальнейшего его использования. Для этого Ерам предоставил отдельную Live-сессию, а также онлайн курс по изучению работы с Git. В ходе практики будут рассмотрены следующие действия:

- Генерация ключа ssh.
- Создание нового репозитория на github.com и клонирование его локально на свой компьютер.
- Создание файла локально и отправка его на сервер с коммитом.
- Создание коммитов и редактирование текстов в веб-интерфейсе github.
- Запуск и использование локальных интерфейсов git gui и gitk.
- Создание файла .gitignore и его использование.

3.2 Этапы работы

Для работы с Git'ом нужно было скачать GitBash на компьютер и сгенерировать ключ ssh. (Рисунок 1)

```
$ ssh-keygen -t rsa -C "21it1.makeenok.d@pdu.by"
Generating public/private rsa key pair.
Enter file in which to save the key (/c/Users/morim/.ssh/id_rsa):
Created directory '/c/Users/morim/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /c/Users/morim/.ssh/id_rsa
Your public key has been saved in /c/Users/morim/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:+7MGRepMJ4KmKF/GJF4e9sxU+w7zwyItu3pe1ZX77hc 21it1.makeenok.d@pdu.by
The key's randomart image is:
+---[RSA 3072]-----+
|
| . . .
| . . + .
| . B o = o +
| o X * =S= .o|
| o o = + *... . E.|
|.. o .*. = . o|
| . .+.o ..|
| . =B+. .+|
+---[SHA256]-----+
```

Рисунок 3.1 – генерация ssh ключей

После чего авторизовать публичный ключ на github.com. А также указать свой user.name и user.email в git.

Следующий этап — это создание нового репозитория на github.com и клонирование его локально на компьютер. Создаём файл song.txt и пишем туда песню. Делаем коммит с помощью команды `git commit -m "Текст"` с названием "Написал песню" и отправляем его на сервер через команду `git push`. (Рисунок 2)

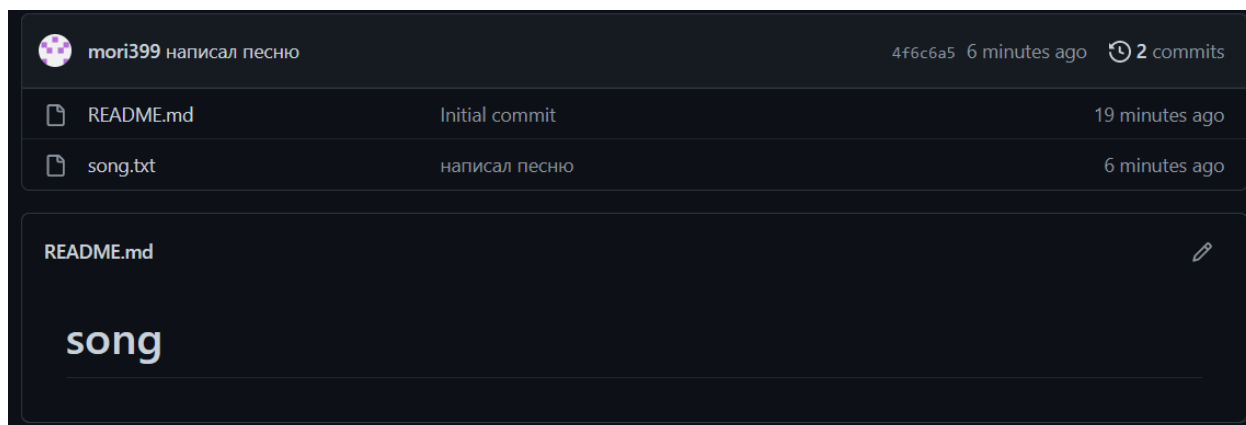


Рисунок 3.2 – файл song.txt попал на github

Проверяем что файл действительно на сервере. Используя веб-интерфейс гитхаба добавляем вторую половину текста песни (Рисунок 3) и делаем коммит с названием "дописал песню" (Рисунок 4). В локальном репозитории делаем pull и убеждаемся, что коммит, который мы создали на github, подтянулся и у нас полный текст песни.

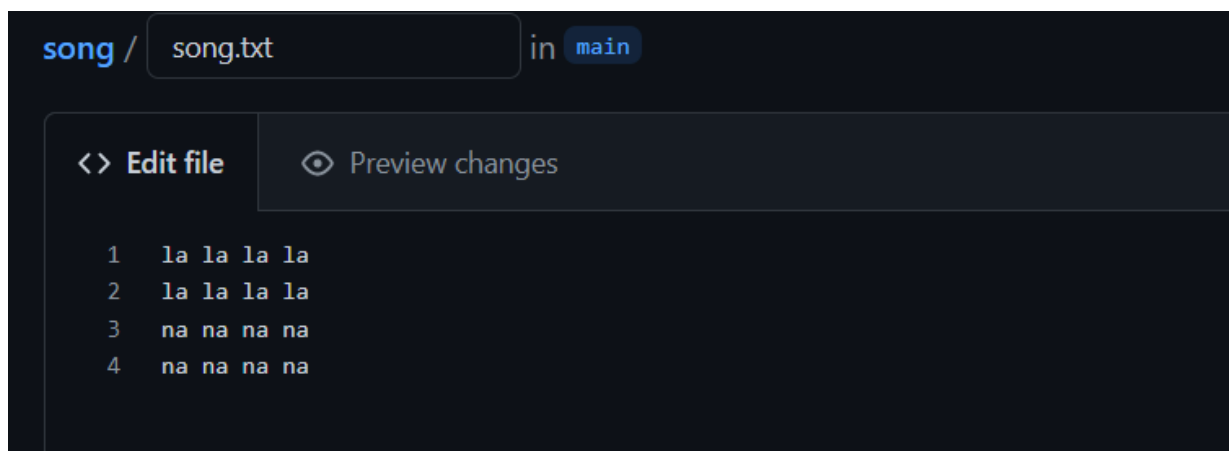


Рисунок 3.3 – Редактирование текста в веб-интерфейсе github

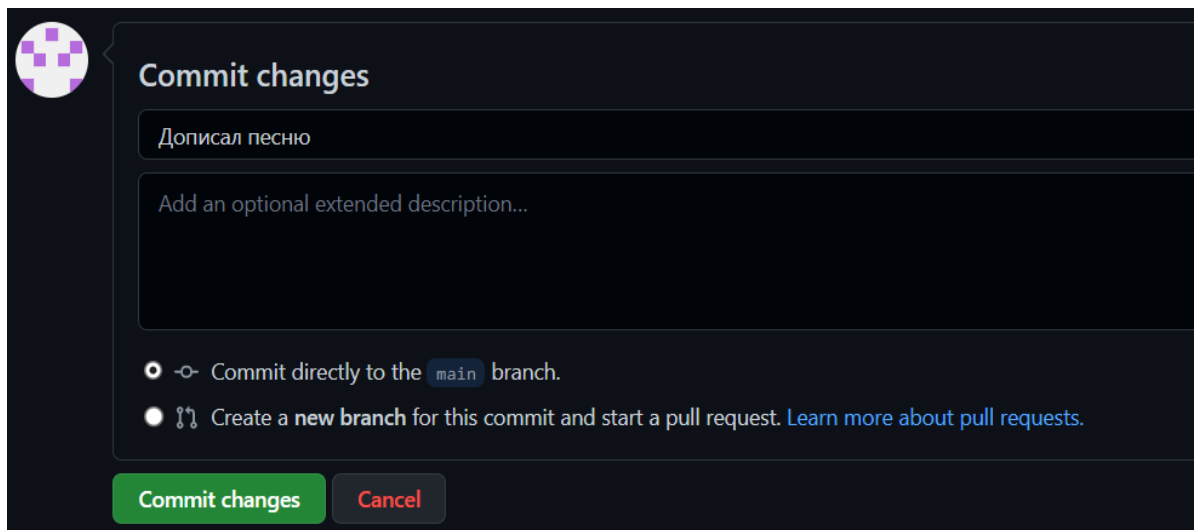


Рисунок 3.4 – отправка коммита в веб-интерфейсе github

Для упрощения работы с git с помощью команды `git gui` мы запускаем интерфейс (Рисунок 5), который отображает изменения файлов, позволяет делать коммиты и отправлять изменения на сервер. Также с помощью команды `gitk` можно просматривать историю коммитов.

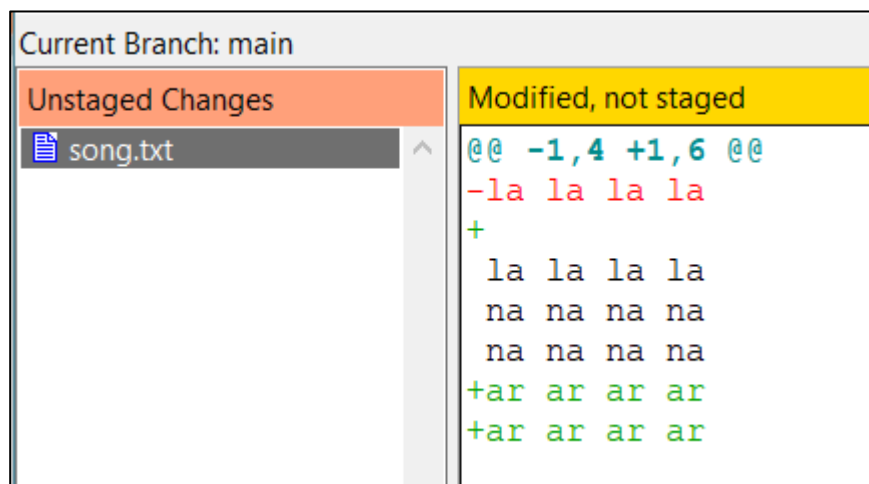


Рисунок 3.5 – интерфейс git gui

При необходимости скрыть файлы с каким-то расширением, мы пользуемся файлом `.gitignore` куда вписываем ненужные расширения (Рисунок 6)

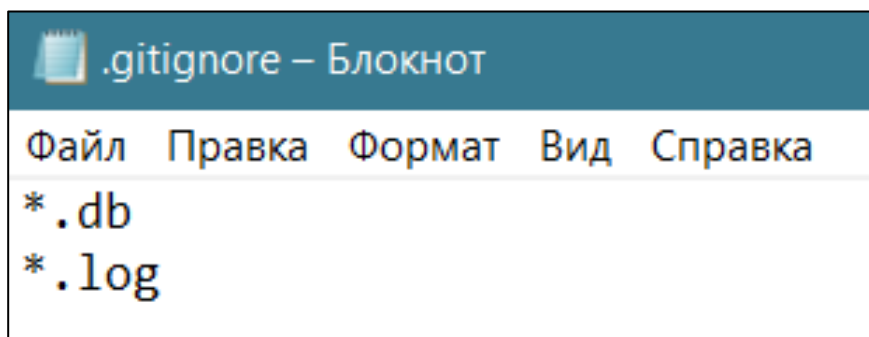


Рисунок 3.6 – содержимое файла .gitignore

Теперь любые файлы с расширением .db и .log не будут отображаться в git gui. Это делается для избегания ненужных коммитов и для удобства работы с файлами. (Рисунок 7)

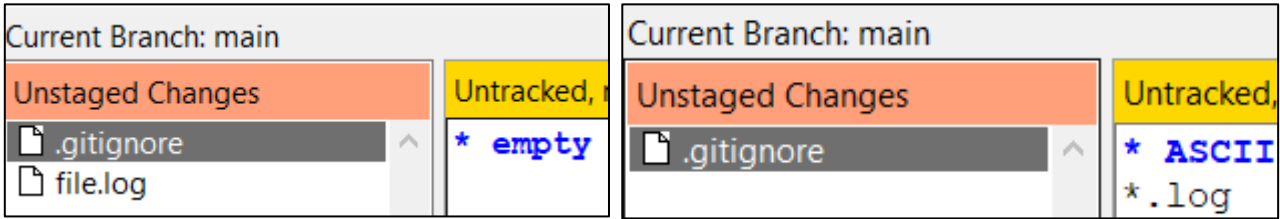


Рисунок 3.7 – пустой .gitignore и с условием

ЗАКЛЮЧЕНИЕ

В ходе учебной ознакомительной практики было видно, как работает коллектив в большой компании и насколько важно не стоять на месте и развиваться.

После прохождения практики стало понятно, что при работе над большим проектом в группе людей не обойтись без системы контроля, так как она даёт возможность разработчикам максимально, продуктивно не путаясь между собой, работать над проектом. В Git присутствуют все необходимые для этого инструменты. И будет крайне полезно хорошему специалисту уметь работать с этой системой контроля.

За время практики были пройдены курсы «Version Control with Git» и «Computer Science Basics». В которых мы ознакомились с основными технологиями и инструментами, используемыми в инженерной работе, а также получили навыки работы с Github.

					МДИ.21114.00.ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК ИСТОЧНИКОВ

- 1 Википедия [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/EPAM_Systems. – Дата доступа: 09.09.2022;
- 2 IT Warm-UP [Электронный ресурс]. – Режим доступа: <https://wearecommunity.io/events/it-warm-up/stream-2>. – Дата доступа: 09.09.2022;
- 3 Vitali Shulha Version Control with Git. Курс от компании EPAM;
- 4 История компании EPAM [Электронный ресурс]. – Режим доступа: <https://mc.today/srednij-chek-klientov-epam-systems-100-mln-v-god-kak-emigrant-iz-belarusi-sozdal-biznes-na-15-mlrd/>. – Дата доступа: 09.09.2022;
- 5 Организационная структура EPAM [Электронный ресурс]. – Режим доступа: https://studbooks.net/2164654/ekonomika/organizatsionno_ekonomicheskaya_harakteristika_predpriyatiya. – Дата доступа: 10.09.2022;
- 6 Контроль версий [Электронный ресурс]. – Режим доступа: <https://www.atlassian.com/ru/git/tutorials/what-is-version-control>. – Дата доступа: 10.09.2022;
- 7 Преимущества разных контролей версий [Электронный ресурс]. – Режим доступа: http://all-ht.ru/inf/prog/p_0_1.html. – Дата доступа: 10.09.2022.

					МДИ.21114.00.ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		16