Частное учреждение профессионального образования

«Высшая школа предпринимательства (колледж)»

(ЧУПО «ВШП»)

Кафедра информационных технологий

Инструментальные средства разработки программного обеспечения

Для разрядки и настройки

Что будет на стажировке:

- Будем учиться разработке на Java. Будет много практической работы, работы с наставниками.
- Будем знакомиться с инструментами разработки, основами проектной работы.
- Будем пробовать свои силы в учебном проекте, где все, как в настоящей работе – заказчик, команда, дедлайны и т.п.

Как все будет проходить:

- Стажировка длится 4 месяца.

Занятия будут проходить по большей части в дистанционном формате, но иногда мы будем встречаться в офисе компании.

- Занятия будут проходить несколько раз в неделю во второй половине дня. Плюс будет много самостоятельной работы, которую нужно будет делать в любое удобное время.
- Стажировка предполагает много самостоятельной работы по программе, работу с наставниками, работу в команде.
- Кроме учебы, мы будем знакомить стажеров с компанией, нашими проектами, нашими подходами в разработке ПО.
- Тех, кто успешно пройдет всю стажировку, мы пригласим на постоянную работу в компанию.
- Начало стажировки – середина октября 2023.

Что нужно, чтобы попасть на стажировку:

- Владеть основами языка программирования Java. Знание основ Java Core: синтаксис, ветвление , циклы, ООП. Будет плюсом, если знаешь коллекции и генерики, функциональные интерфейсы и лямбда выражения, Stream API.
- Быть выпускником (любого года) или студентом старших курсов ИТ-специальностей вуза или суза.
- Иметь возможность уделять стажировке не менее 20 часов в неделю.
- Иметь большое желание стать программистом, профессионально развиваться в это области.
- Быть готовым к постоянному самообразованию.
- Иметь ноутбук или ПК для выполнения заданий стажировки.

Мы также готовы принять на стажировку выпускников непрофильных специальностей вузов, которые изучают язык Java самостоятельно и хотят далее заниматься разработкой программного обеспечения.

Что такое «система контроля версий»?

Повторим термины

- Git или Гит
- Репозиторий Git
- Локальный репозиторий
- Удалённый репозиторий
- Форк (Fork)
- Клонирование (Clone)
- Ветка (Branch)
- Мастер (Master) или Мейн (Main)
- Коммит (Commit)
- Мёрдж (Merge)

Повторим основные команды Git

git init

git status

git add

git commit

git log

Состояния файлов – что такое?

Повторим команды Git для работы с ветками

git branch

git checkout

git merge

git checkout -b new-feature

git add <file>

git commit -m "feat: start a feature"

git add <file>

git commit -m "feat: finish a feature"

git checkout main

git merge new-feature

git branch -d new-feature

Командная строка

Командная строка или **интерфейс командной строки** (англ. CLI, Command Line Interface) - текстовое приложение для просмотра, обработки и манипулирования файлами на вашем компьютере.

Она делает то же, что и **Проводник** в Windows.

Основные команды
Перемещение по папкам

pwd — узнать полный путь к папке, в которой сейчас находимся.
cd название_папки — для перехода в выбранную папку.
cd .. — переместиться на одну папку назад.
cd / — переместиться на одну папку вперед.
cd - — вернуться к предыдущей папке.
cd ~ — перейти в папку, которая назначена домашней.

cd /d (перейти на диск D)
cd code/'First Task' (перейти в папку "code", а затем в папку "First Task" — в коде она указана в кавычках, так как содержит пробелы)

Основные команды
Обзор папки

dir — какие файлы есть в папке
tree - показать дерево папок

Быстрые клавиши:
cd ~/D + tab — покажет, какие папки начинаются на букву "D".
cd ~/Des + tab — терминал сам допишет вместо "Des" — "Desktop".

Основные команды
Создание и удаление файлов и папок через консоль

touch index.html — создать файл index.html

mkdir project — создать папку с именем project

type text.txt — вывод содержимого файла на экран

del index.html — удалить файл index.html
erase index.html — удалить файл index.html

rmdir project — удалить папку с именем project
rd project — удалить папку с именем project

Несколько полезных команд

cls - очистить консоль

systeminfo - информация о системе

ipconfig - информация о сетевых настройках

tasklist - список запущенных процессов

exit - выход

GitHub

Крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки.

Кроме размещения кода, участники могут общаться, комментировать правки друг друга, а также следить за новостями знакомых.

Для проектов есть личные страницы, небольшие Вики и система отслеживания ошибок.

GitHub

GitHub

1. Можно создавать **приватные репозитории**, которые будут видны только вам и выбранным вами людям. Раньше такая возможность была платной.
2. Есть возможность **прямого добавления новых файлов** в свой репозиторий через веб-интерфейс сервиса.
3. Код проектов можно **не только скопировать** через Git, но и **скачать в виде обычных архивов** с сайта.
4. Кроме Git, сервис **поддерживает получение и редактирование кода** через SVN и Mercurial.
5. На сайте есть **pastebin-сервис gist.github.com** для быстрой публикации фрагментов кода.
6. Файлы из репозитория могут автоматически публиковаться в виде статического сайта с помощью **GitHub Pages**.

Регистрируемся на GitHub

Профиль на **Гитхабе** и все проекты в нём — ваше публичное портфолио разработчика, поэтому нужно завести профиль, если у вас его ещё нет.

1. Зайдите на сайт **https://github.com** и нажмите кнопку Sign up.
2. Введите имя пользователя (понадобится в дальнейшей работе), адрес электронной почты (такой же, как при настройке Git) и пароль.
3. На почту придёт код активации — введите на сайте.
4. Появится окно с выбором тарифного плана. Если вы пользуетесь Гитхабом для учёбы, то укажите, что профиль нужен только для вас и вы студент.

Документация GitHub

https://docs.github.com/ru

Генерация SSH-ключей для GitHub

Для начала создания ключей введем следующую команду в Git консоль:

ssh-keygen -t rsa -b 4096 -C "ВАШ_EMAIL@mail.com"

В данном случае, E-mail будет использован как метка для удобства в дальнейшем использовании.
Далее можно указать путь для сохранения ключей. Или нажать Enter, чтобы установить в место, предлагаемое по умолчанию. В моем случае это будет **C:/Users/ilya/.ssh/id_rsa**
Теперь Git попросит нас ввести любую ключевую фразу для более надежной защиты вашего пароля. Можно пропустить этот этап просто нажав Enter:

Enter passphrase (empty for no passphrase):
Enter same passphrase again:

Добавляем SSH-ключ в ssh-agent

ssh-agent — программа для хранения и управления SSH-ключами. Давайте запустим её и добавим туда наш SSH-ключ.

eval "$(ssh-agent -s)"
Если в ответ терминал покажет надпись «Agent pid» и число — значит, всё ок, агент запущен.

Теперь добавим наш ключ:

ssh-add ~/.ssh/id_ed4545

Замените название id_ed4545 именем своего файла с ключом. Если вы устанавливали пароль на ключ, введите его два раза после ввода команды ***ssh-add***.

Копируем SSH-ключ

Введите команду ниже и ключ появится прямо в терминале — его нужно вручную скопировать в буфер обмена.

Ключ начинается с ssh-ed4545 или ssh-rsa (или похожей строки) — поэтому копируйте строку прямо с самого начала.

~ cat ~/.ssh/id_ed4545.pub

ssh-ed4545 AAAAC3NzaCZvnr4ax+Fr ВАШ_EMAIL@mail.com

Замените название id_ed4545 именем своего файла с ключом.

Добавляем SSH-ключ на GitHub

Это нужно сделать, чтобы **GitHub** вас узнавал.

Перейдите на страницу для работы с ключами в вашем профиле на **GitHub** и нажмите кнопку **New SSH key**.

В поле Title нужно добавить название нового ключа. Например, если вы

А ключ, который вы скопировали на прошлом шаге, вставьте в поле Key.

Теперь нажмите кнопку **Add SSH key** и, если потребуется, введите свой пароль от **GitHub**, чтобы подтвердить сохранение.

Если всё сделано верно, новый ключ появится в списке

ДЗ

Регистрируемся на ГитХабе

1. Установите git (например, от сюда http://git-scm.com/).
2. Создайте репозиторий.
3. Создайте в репозитории текстовый файл с текстом (2-3 абзаца текста).
4. Разбейте его создание на несколько этапов (добавление 2 абзаца). Коммитьте изменения (2-3 коммита)
5. Создайте две ветки.
6. В первой ветке «поработайте» над текстом (добавьте 2 абзаца).
7. Во второй уберите содержания (удалите 1 абзац)
8. Не забывайте разбивать эту «работу» на коммиты.
9. Слейте все изменения назад в ветку master.
10. Удалите вспомогательные ветки.
11. Изучите команду git log
12. Сделайте скриншот вывод команды git log, который подтвердил бы, что Вы сделали пункты 1-10
13. Сделайте архив из папки репозитория и принесите на занятие

Требования к именам коммитов

init: - используется для начала проекта/таска.
init: start task

feat: - это реализованная новая функциональность из технического задания.
feat: add basic page
feat: add button

fix: - исправил ошибку в ранее реализованной функциональности.
fix: change layout to fix bugs

Дополнительный материал

gosuslugi.ru/futurecode

ru.hexlet.io/courses/intro_to_git

practicum.yandex.ru/git-basics