Некоммерческое акционерное общество

Карагандинский технический университет имени Абылкаса Сагинова

Кафедра ИВС

**Лабораторная работа №4**

**Дисциплина: «**Современные технологии программирования»

**Тема:** «Управление Версиями Проекта»

**Подготовил:**

Студент группы

ИСМ-23-2

Сабиголла Г.К.

**Приняла:**

Доцент

Томилова Н. И.

Караганда 2023

**Личная карточка**

**потенциального**

**абитуриента**

Цель работы: приобрести практические навыки работы с системой контроля версий Git.

Порядок выполнения

1. Изучить инструкцию по работе с Git

2. Выполнить предложенные задания

3. Разбиться на группы и выбрать тему проекта (что-то простое, вроде крестиков-ноликов)

4. Распределить работу между членами группы (кто какую часть проекта будет писать)

5. Создать новый репозиторий для проекта

6. Разработать свою часть проекта

7. Выкачать последнюю версию проекта с удаленного репозитория и разрешить конфликты, если такие возникли

8. Сделать коммит и запушить коммит в удаленный репозиторий

9. Собрать все части проекта в мастере, выкачать изменения в локальный репозиторий и проверить работу целого проекта

Содержание отчета

1. Постановка задачи

2. Консольные команды и скриншоты с пояснением о выполнении каждого шага по работе с Git (создание репозитория, выкачка изменений с удаленного репозитория, коммит, пуш и тд)

3. Вывод

4. Ответы на контрольные вопросы

Тема варианта проекта

11. Игра в крестики/ нолики

Листинг кода

def print\_board(board):

for row in board:

print(" | ".join(row))

print("-" \* 5)

def check\_winner(board):

# Проверка строк

for row in board:

if row.count(row[0]) == len(row) and row[0] != ' ':

return True

# Проверка столбцов

for col in range(len(board[0])):

if all(board[row][col] == board[0][col] and board[row][col] != ' ' for row in range(len(board))):

return True

# Проверка диагоналей

if all(board[i][i] == board[0][0] and board[i][i] != ' ' for i in range(len(board))) or \

all(board[i][len(board)-i-1] == board[0][len(board)-1] and board[i][len(board)-i-1] != ' ' for i in range(len(board))):

return True

return False

def is\_board\_full(board):

return all(board[i][j] != ' ' for i in range(len(board)) for j in range(len(board[0])))

def tic\_tac\_toe():

board = [[' ' for \_ in range(3)] for \_ in range(3)]

current\_player = 'X'

while True:

print\_board(board)

row = int(input(f"Игрок {current\_player}, выберите строку (0, 1, 2): "))

col = int(input(f"Игрок {current\_player}, выберите столбец (0, 1, 2): "))

if board[row][col] == ' ':

board[row][col] = current\_player

if check\_winner(board):

print\_board(board)

print(f"Игрок {current\_player} победил!")

break

elif is\_board\_full(board):

print\_board(board)

print("Ничья!")

break

else:

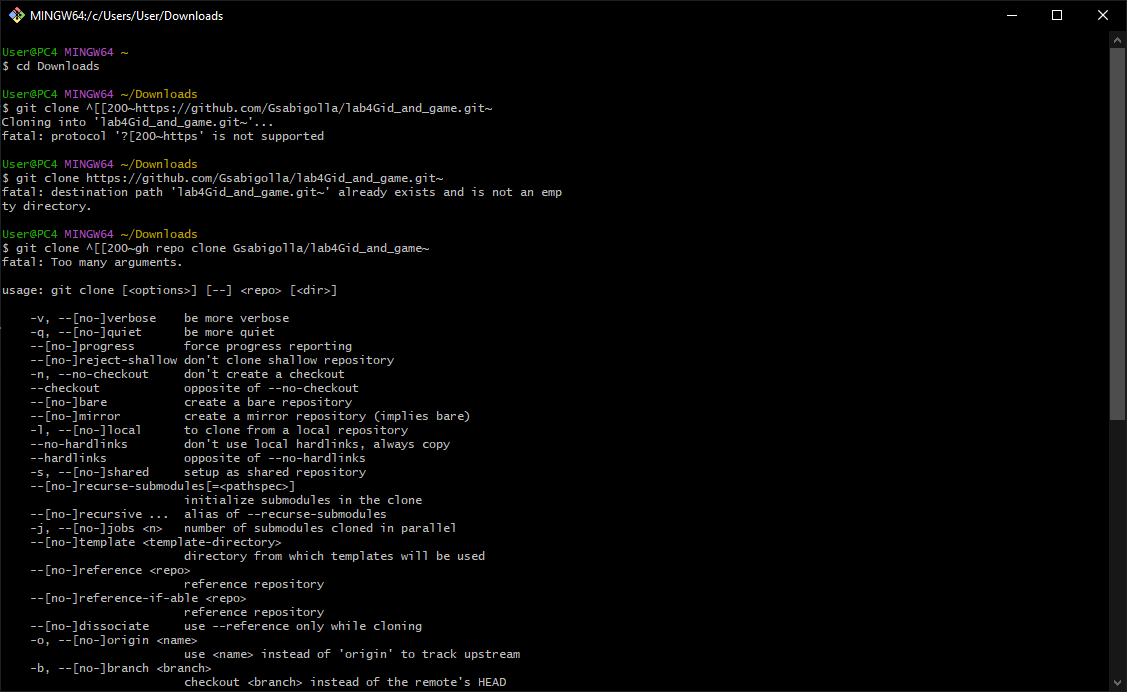
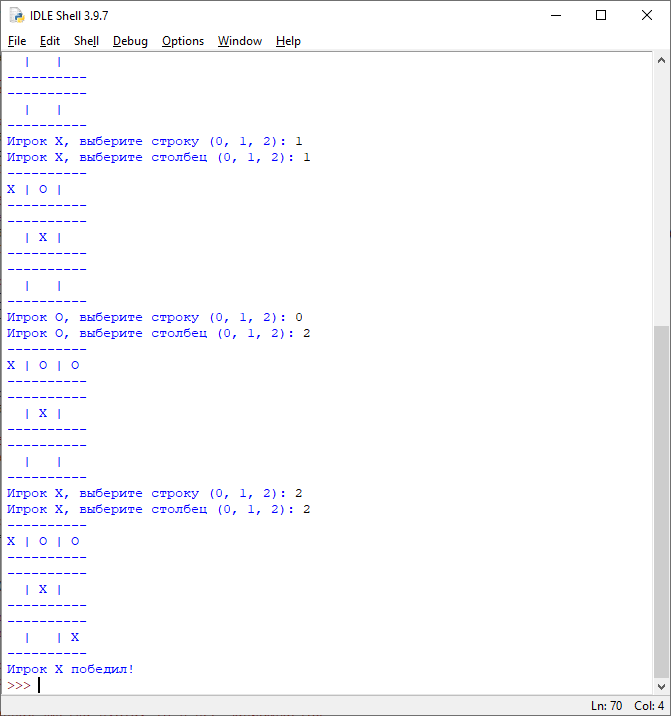
current\_player = 'O' if current\_player == 'X' else 'X'

else:

print("Эта клетка уже занята. Попробуйте снова.")

if \_\_name\_\_ == "\_\_main\_\_":

tic\_tac\_toe()



Контрольные вопросы

1. Что такое система контроля версий?

Система контроля версий (СКВ) - это система, которая записывает изменения в файлы или набор файлов с течением времени, чтобы вы могли вернуться к определенным версиям позже. Это позволяет отслеживать изменения, сравнивать версии, а также объединять изменения от нескольких человек.

2. Зачем нужна система контроля версий?

Система контроля версий позволяет эффективно управлять изменениями в коде и других файлах. Она предоставляет возможность отслеживать историю изменений, возвращаться к предыдущим версиям, объединять изменения от нескольких людей, а также обеспечивает безопасное и удобное сотрудничество.

3. Значение команды Add?

git add <имя файла> - для добавления одного файла

git add. -для добавления всех файлов в папке

4. Значение команды Commit?

git commit-m - для того чтобы зафиксировать изменения (commit это точка сохранения)

5. Значение команды Push?

git push origin <имя\_ветки> - чтобы ветка оказалась на удаленном репозиторий и там создать её

6. Значение команды Pull?

git pull - чтобы стинуть себе все изменнения из удаленного репозитория

7. Как создать новый репозиторий?

git init - создать локальный репозиторий

8. Что такое локальный репозиторий?

Локальный репозиторий - это копия проекта, хранящаяся на вашем компьютере. Здесь вы можете вносить изменения, фиксировать их и работать без подключения к удаленному репозиторию.

9. Что такое удаленный репозиторий?

Удаленный репозиторий - это хранилище проекта, расположенное на удаленном сервере. Он служит для совместной работы нескольких разработчиков и обеспечивает централизованное хранение изменений.

10. Что такое Git?

Git - это распределенная система контроля версий, разработанная для эффективного управления проектами любого размера. Она позволяет отслеживать изменения, возвращаться к предыдущим версиям, сливать изменения от нескольких источников и обеспечивает эффективное сотрудничество.