Практические заданий для начального уровня по дисциплине «Инструментальные средства разработки ПО».

Блок 1: Работа с текстовым редактором/IDE (Visual Studio Code)

Задание 1: «Знакомство с средой разработки»

Цель: научиться создавать проект и пользоваться базовыми возможностями редактора.

Задача:

1. Установите редактор Visual Studio Code.

2. Установите в него рекомендуемые расширения для Python (или другого выбранного языка), например, Python, Pylance.

3. Создайте на диске папку `MyFirstProject`.

4. Откройте эту папку в VSCode.

5. Создайте в ней файл `main.py`.

6. Напишите код: `print("Моя первая программа!")`

7. Запустите код прямо из редактора (кнопка «Play» в правом верхнем углу или через меню).

Критерий успеха: В терминале VSCode появилась надпись «Моя первая программа!».

Блок 2: Система контроля версий (Git)

Задание 2: «Первый коммит»

Цель: Понять, как сохранять изменения в коде с помощью Git.

Задача:

1. В терминале VSCode перейдите в папку вашего проекта (`MyFirstProject`).

2. Инициализируйте git-репозиторий: `git init`

3. Проверьте статус репозитория: `git status` (вы увидите неотслеживаемый файл `main.py`).

4. Добавьте файл в отслеживаемые: `git add main.py`

5. Снова проверьте статус (`git status`). Теперь файл зеленый.

6. Сохраните (закоммитьте) это состояние: `git commit -m "Добавил первую программу"`

Критерий успеха: Команда `git log` показывает один коммит с вашим сообщением.

Задание 3: «Работа с GitHub»

Цель: Научиться публиковать код в интернете.

Задача:

1. Создайте аккаунт на GitHub.com (если нет).

2. Создайте новый репозиторий с именем `MyFirstProject` (оставьте все настройки по умолчанию).

3. Следуя инструкциям на GitHub, свяжите ваш локальный репозиторий с удаленным (команды `git remote add origin...` и `git branch -M main`).

4. Отправьте ваш код на GitHub: `git push -u origin main`

Критерий успеха: Ваш файл `main.py` виден в репозитории на GitHub.

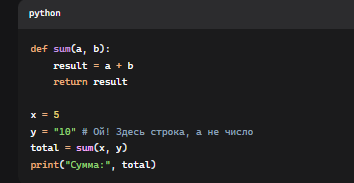
Блок 3: Базовая отладка (Debugging)

Задание 4: «Ищем ошибку»

Цель: Научиться пользоваться точками останова (breakpoints) и пошаговым выполнением кода.

Задача:

1. В файл `main.py` добавьте код с небольшой ошибкой:



2. Запустите программу. Увидите ошибку `TypeError`.

3. Теперь поставьте точку останова (щелкните слева от номера строки) на строке `total = sum(x, y)`.

4. Запустите программу в режиме отладки (жмите F5 или кнопку «Debug» и выберите «Python File»).

5. Программа остановится на вашей точке. Нажимайте F10 (Step Over) чтобы выполнять код по шагам.

6. Наведите курсор на переменные `x` и `y`, чтобы увидеть их значения и тип. Найдите причину ошибки.

Критерий успеха: С помощью отладчика вы обнаружили, что переменная `y` является строкой, а не числом.

Блок 4: Базовая автоматизация (Пакетные менеджеры)

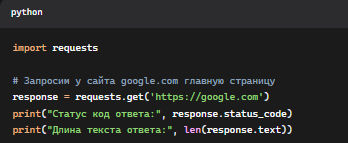
Задание 5: «Используем чужую библиотеку»

Цель: Понять, как устанавливать и использовать внешние библиотеки.

Задача (на примере Python):

1. В терминале VSCode установите библиотеку для генерации случайственных чисел (она очень популярна): `pip install requests`

2. Измените код в `main.py`:



3. Запустите программу.

Критерий успеха: Программа напечатала статус `200` (успех) и длину HTML-кода главной страницы Google.

Блок 5: Простое тестирование

Задание 6: «Напиши один тест»

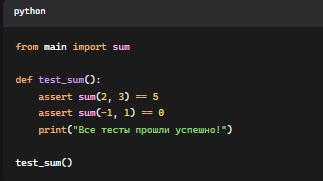
Цель: Понять, для чего нужны автоматические тесты.

Задача:

1. Установите библиотеку для тестирования: `pip install pytest`

2. Создайте новый файл `test\_my\_code.py`.

3. Напишите в нем простой тест для функции `sum` из прошлых заданий (предварительно исправьте функцию, чтобы она была в коде):



4. Запустите файл.

5. (Дополнительно) Попробуйте запустить его через `pytest` в терминале: просто напишите `pytest`. Он автоматически найдет и запустит ваш тест.

Критерий успеха: При запуске файла выводится сообщение «Все тесты прошли успешно!».

Итоговое задание для закрепления

Задание: «Собери все вместе»

Цель: Применить все полученные навыки в одном мини-проекте.

Задача:

1. Создай: Новую папку `NumberGame`. Открой ее в VSCode.

2. Напиши код: Простую игру "Угадай число", где компьютер загадывает число, а пользователь угадывает.

3. Закоммить: Инициализируй git и сделай несколько коммитов с осмысленными сообщениями.

4. Опубликуй: Создай репозиторий на GitHub и запуши код.

5. Протестируй: Напиши один тест для функции проверки числа (например, проверь, что функция правильно сравнивает число пользователя с загаданным).

6. Используй библиотеку: Используй библиотеку `random` для загадывания числа.

Критерий успеха: Весь код выложен на GitHub, есть как минимум 3 коммита, программа работает, тест проходит.