**📚 Методичка по работе с Git и PyCharm для проекта Filin\_AIB**

**1. Установка и настройка на домашнем компьютере**

**1.1. Установите необходимые программы:**

* **Git**: <https://git-scm.com/> (скачайте, установите всё по умолчанию)
* **PyCharm**: <https://www.jetbrains.com/pycharm/> (Community версия бесплатна)
* **Python**: <https://python.org/> (версия 3.8+)

**1.2. Проверьте установку Git:**

bash

git --version

**2. Перенос проекта на домашний компьютер**

**2.1. Склонируйте проект с GitHub:**

bash

# Откройте терминал в папке где хотите разместить проект

git clone https://github.com/sfirulin1973/Filin\_AIB.git

cd Filin\_AIB

**2.2. Настройте проект в PyCharm:**

1. **Откройте PyCharm**
2. **File → Open** → выберите папку Filin\_AIB
3. PyCharm автоматически распознает Git репозиторий

**3. Основные команды Git - Шпаргалка**

**📥 Получить последние изменения с GitHub:**

bash

git pull

*Делайте эту команду КАЖДЫЙ РАЗ перед началом работы!*

**📤 Сохранить и отправить свои изменения:**

bash

# Добавить все измененные файлы

git add .

# Создать коммит с описанием

git commit -m "Описание что вы сделали"

# Отправить на GitHub

git push

**🔍 Проверить статус:**

bash

git status

**📋 Посмотреть историю изменений:**

bash

git log --oneline

**4. Рабочий процесс день за днем**

**Утром на работе:**

bash

cd E:\PYTHON\Filin\_AIB

git pull

# Начинайте работать

**Вечером после работы:**

bash

git add .

git commit -m "Описание сделанной работы"

git push

**Дома:**

bash

cd C:\Users\ВашеИмя\Filin\_AIB # или другая папка

git pull

# Работаете дома

git add .

git commit -m "Домашние изменения"

git push

**Утром на работе снова:**

bash

git pull # получаете домашние изменения

# Продолжаете работу

**5. Настройка виртуального окружения**

**Создайте виртуальное окружение:**

bash

python -m venv .venv

**Активируйте его:**

* **Windows**: .venv\Scripts\activate
* **Mac/Linux**: source .venv/bin/activate

**Установите зависимости:**

bash

pip install -r requirements.txt

**6. Что делать в случае проблем**

**Если что-то пошло не так:**

bash

# Отменить последний коммит (локально)

git reset --soft HEAD~1

# Откатить конкретный файл

git checkout -- имя\_файла.py

# Посмотреть что изменилось

git diff

**Если конфликт при git pull:**

1. PyCharm покажет конфликтующие файлы
2. Выберите какие изменения оставить (ваши/удаленные/объединить)
3. Сохраните файлы
4. Выполните:

bash

git add .

git commit -m "Разрешение конфликта"

git push

**7. Полезные настройки Git**

**Настройте ваше имя и email (сделайте один раз):**

bash

git config --global user.name "Ваше Имя"

git config --global user.email "ваш@email.com"

**Включите кэширование пароля:**

bash

git config --global credential.helper cache

**8. Структура вашего проекта в PyCharm**

После открытия проекта вы увидите:

text

Filin\_AIB/

├── .venv/ # Виртуальное окружение (не в гите)

├── .git/ # Папка гита (не трогать!)

├── main\_Filin\_AIB.py # Основной файл

├── UI\_MAIN\_FILIN.ui # Файл интерфейса

├── requirements.txt # Зависимости

├── .gitignore # Что игнорировать в гите

└── Документация по проекту/ # Ваша документация

**9. Частые сценарии**

**Сценарий 1: Работал дома, пришел на работу**

bash

git pull

**Сценарий 2: Закончил работу, хочу сохранить**

bash

git add .

git commit -m "Добавил новую функцию X"

git push

**Сценарий 3: Хочу посмотреть что изменилось**

bash

git status

git diff

**Сценарий 4: Сделал ошибку, хочу откатиться**

bash

git checkout -- имя\_файла.py

**10. Визуальный контроль в PyCharm**

PyCharm показывает статус файлов цветами:

* **🟢 Зеленый** - новый файл
* **🔵 Синий** - измененный файл
* **🔴 Красный** - не отслеживается в гите
* **⚫ Серый** - игнорируется

**🎯 Золотые правила:**

1. **Перед работой ВСЕГДА** делайте git pull
2. **После работы ВСЕГДА** делайте git add ., git commit, git push
3. **Пишите понятные описания** в коммитах
4. **Не коммитьте** пароли и конфиденциальные данные
5. **Работайте только** в своей копии проекта

Эта методичка покрывает 95% повседневных задач. Сохраните её и обращайтесь когда нужно! 😊

**📋 У вас чистый PySide6 проект!**

Ваш requirements.txt показывает, что проект использует только **PySide6** и его компоненты. Это хорошо!

**🚀 Инструкция для домашнего компьютера:**

**Шаг 1: Клонируйте проект**

bash

git clone https://github.com/sfirulin1973/Filin\_AIB.git

cd Filin\_AIB

**Шаг 2: Откройте в PyCharm**

* **File → Open** → выберите папку Filin\_AIB
* PyCharm автоматически предложит настроить интерпретатор

**Шаг 3: Настройте виртуальное окружение в PyCharm**

1. При открытии проекта PyCharm покажет: **"No Python interpreter configured"**
2. Нажмите **"Configure Python interpreter"**
3. Выберите:
   * **New environment** → Location: \.venv
   * Base interpreter: ваш Python 3.8+
   * ✅ **Install packages from requirements.txt**
4. **Нажмите OK**

**Шаг 4: PyCharm сделает всё автоматически!**

* Создаст виртуальное окружение .venv
* Установит PySide6==6.9.2 и все зависимости
* Готово к работе!

**🔧 Альтернатива через терминал:**

bash

# Создать виртуальное окружение

python -m venv .venv

# Активировать

# Windows:

.venv\Scripts\activate

# Mac/Linux:

source .venv/bin/activate

# Установить зависимости

pip install -r requirements.txt

**✅ Проверка установки:**

bash

python -c "import PySide6; print('PySide6 установлен!')"

python main\_Filin\_AIB.py

**💡 Важные моменты:**

**Если PyCharm не предлагает автоматическую установку:**

1. **Вручную установите зависимости:**

bash

pip install -r requirements.txt

**Если возникают ошибки совместимости:**

bash

# Можно установить последнюю версию PySide6

pip install PySide6

**Для обновления зависимостей на рабочем компьютере:**

bash

# На рабочем компьютере обновите requirements.txt

pip freeze > requirements.txt

git add requirements.txt

git commit -m "Обновлены версии зависимостей"

git push

**🎯 Преимущества вашего текущего setup:**

1. **Чистые зависимости** - только PySide6, нет лишнего
2. **Конкретные версии** - гарантированная совместимость
3. **Простая установка** - одна команда pip install -r requirements.txt

**📞 Если что-то пойдет не дома:**

**Ошибка "Python not found":**

* Убедитесь, что Python установлен
* Проверьте путь в PyCharm: **File → Settings → Python Interpreter**

**Ошибка с PySide6:**

bash

# Попробуйте переустановить

pip uninstall PySide6

pip install PySide6

**Ваш проект должен развернуться без проблем - зависимости минимальные и четко указаны!** 🎉