

РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМ. А.И. ГЕРЦЕНА
ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ТЕХНОЛОГИЧЕСКОГО ОБРАЗОВАНИЯ



Обзор интегрированной среды разработки PyCharm

Работу выполнили:
Лебедева Наталия,
Трофимцова Екатерина,
очная форма обучения,
курс: 2; группа: 2об_ИВТ-1

Санкт-Петербург
2025

Что такое PyCharm?



Продукт: PyCharm от JetBrains.

Назначение: Интегрированная среда разработки (IDE) для языка Python.

Характеристика: Это мощная и функциональная IDE, признанная одной из лучших для профессиональной разработки.

Ключевые преимущества

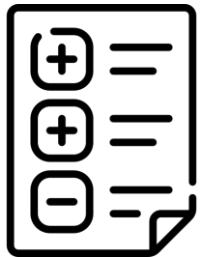
Умное редактирование: Автодополнение, анализ кода на лету, быстрые исправления.

Быстрая навигация: Мгновенный переход к объявлениям функций и классов.

Встроенные инструменты: Отладчик, тестирование, система контроля версий (Git).

Поддержка технологий: Django, Flask, Jupyter, базы данных и многое другое.

Настраиваемость: Огромное количество плагинов и тем оформления.



Версии PyCharm



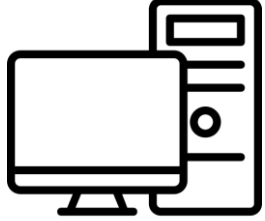
PyCharm Community (Бесплатная):

- Идеальна для чистого Python и обучения.
- Открытый исходный код.
- Нет продвинутых функций для веб-разработки.



PyCharm Professional (Платная):

- Полная функциональность.
- Поддержка веб-фреймворков (Django, Flask).
- Инструменты для научной разработки и работы с БД.



Требования для работы

Аппаратные (рекомендуемые):

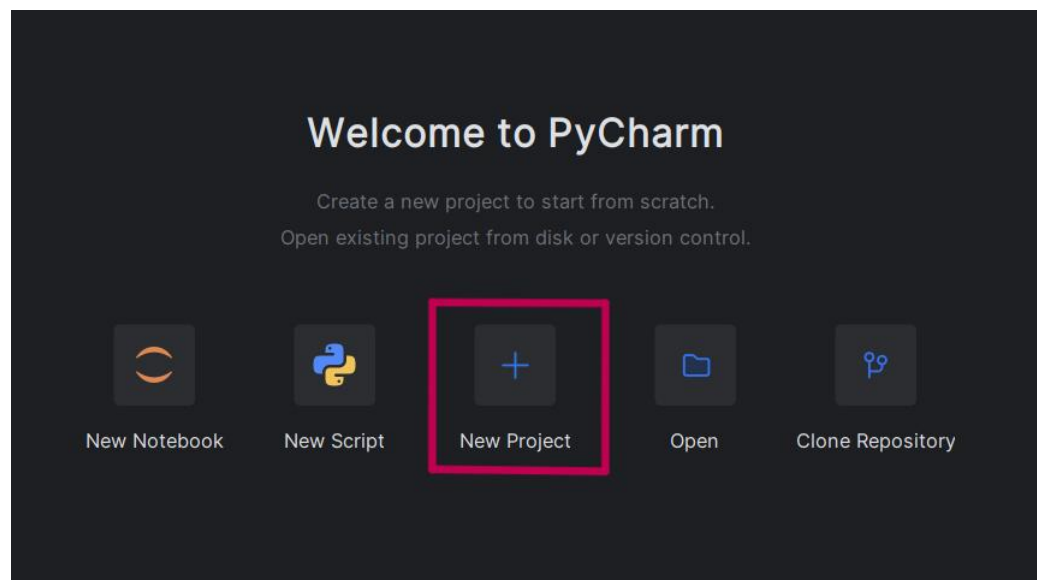
- **Память (RAM):** Не менее 8 ГБ (4 ГБ для IDE).
- **Диск:** 5 ГБ свободного места (желательно SSD).
- **Монитор:** Разрешение 1920x1080 для комфортной работы.

Программные:

- **ОС:** Windows 10/11 (64-bit), macOS 10.14+, Linux.
- **Python:** Версии 2.7, 3.6 и выше.
- **Дополнительно:** JRE (входит в установщик), Git, Docker.



Начало работы: Создание проекта

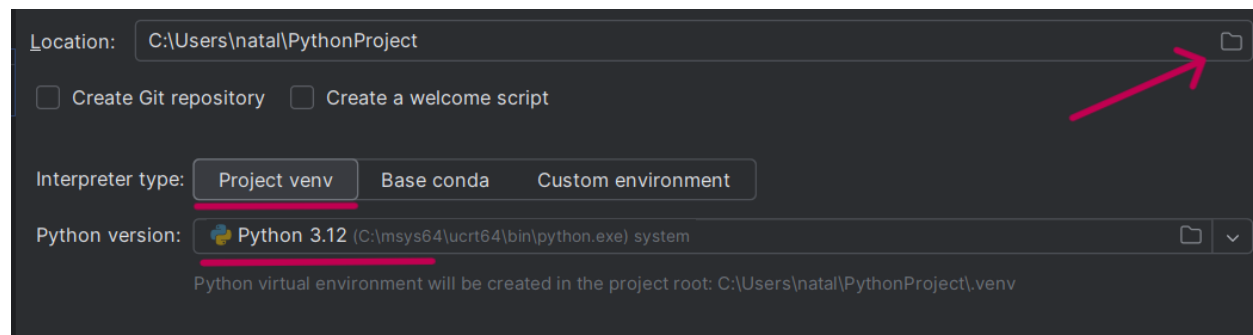


1. Запустите PyCharm.

2. Нажмите «**New Project**».

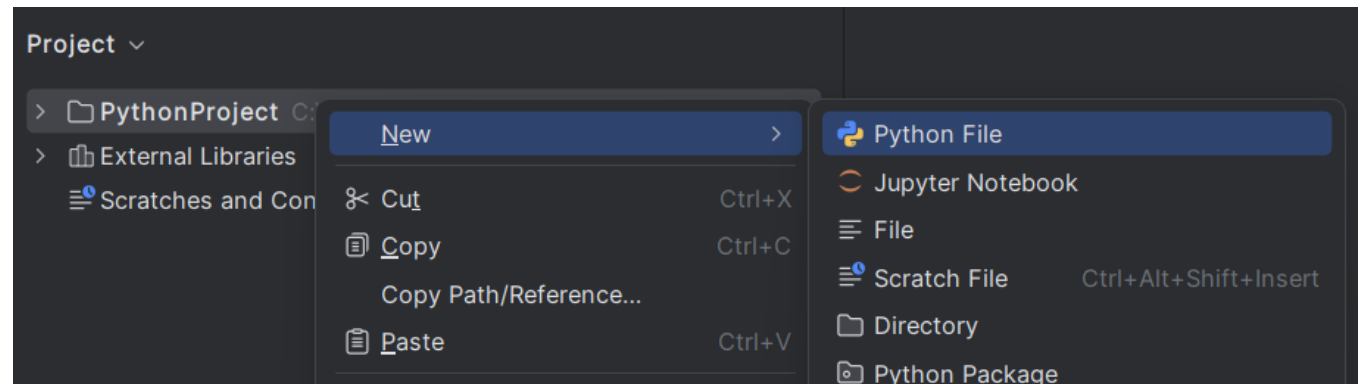
3. Укажите:

- Расположение на диске.
- Тип проекта (Pure Python, Django и др.).
- Интерпретатор Python (рекомендуется создать виртуальное окружение).



Создание и редактирование файла

1. В панели проекта кликните правой кнопкой по папке.
2. Выберите **New → Python File**.
3. Введите имя файла.
4. Начинайте писать код в открывшемся редакторе.



Интеллектуальное кодирование



Автодополнение (Ctrl+Space): Подсказывает переменные, функции, классы.

Умное завершение (Ctrl+Shift+Space): Фильтрует методы по типу переменной.

Навигация (Ctrl+ЛКМ): Мгновенный переход к определению функции или класса.

Рефакторинг (Shift+F6): Безопасное переименование переменных во всем проекте.



Встроенный отладчик

1. Установите точку останова (щелчок на левом поле).
2. Запустите программу в режиме отладки (**Shift+F9**).
3. Используйте шаги:
 - **Step Over (F8)**
 - **Step Into (F7)**
 - **Step Out (Shift+F8)**
4. Следите за значениями переменных в панели **Debugger**.

Форматирование кода



Автоформатирование (Ctrl+Alt+L): Приводит код к единому стилю.

PEP 8: IDE анализирует код на соответствие стандарту и предлагает исправления (**Alt+Enter**).

Интеграция с Black: Настройка автоматического форматирования при сохранении.

Действия при сохранении: Автоматический рефакторинг и оптимизация импортов.

Отладка



- Точки останова: обычные, условные, «только лог»
- Опция **Suspend**: остановка всех потоков (All) или только текущего (Thread)
- Дополнительные параметры: вывод сообщения, стек вызовов, лог выражений, удаление после первого срабатывания

main.py:5

☒ Enabled

☒ Suspend: ☐ All ☒ Thread

☐ Condition:

Log: ☐ "Breakpoint hit" message ☐ Stack trace

☐ Evaluate and log:

☐ Remove once hit

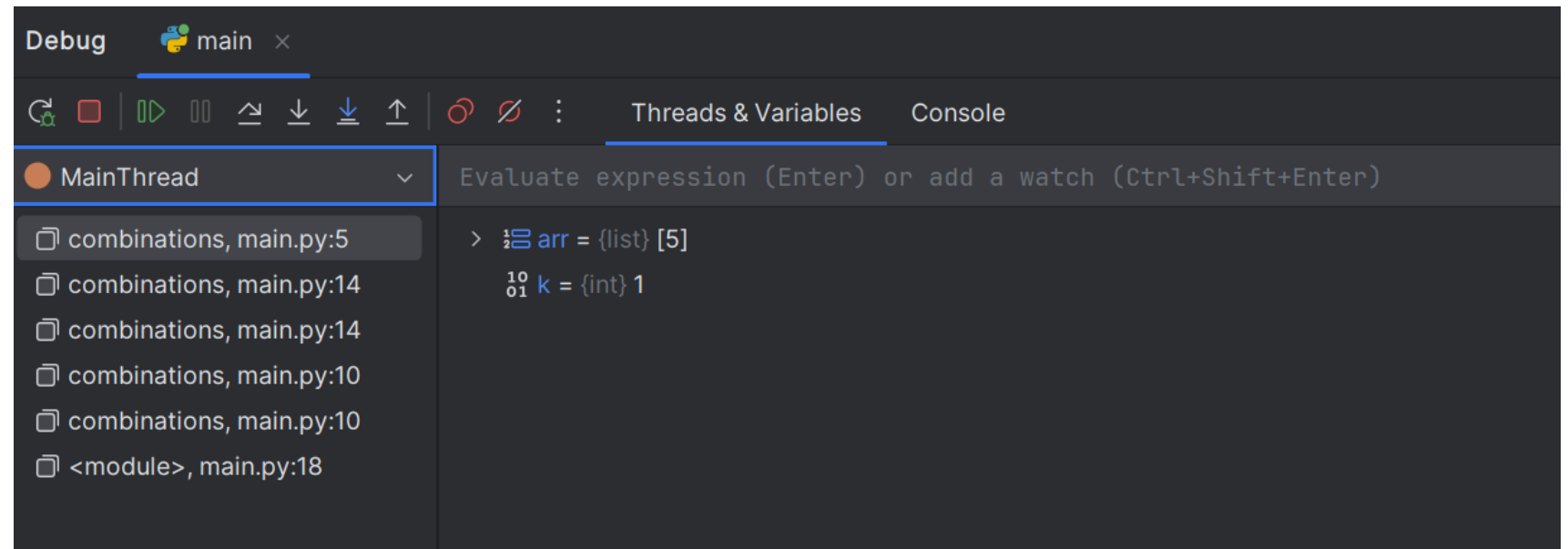
Disable until hitting the following breakpoint:

After hit: ☒ Disable again ☐ Leave enabled

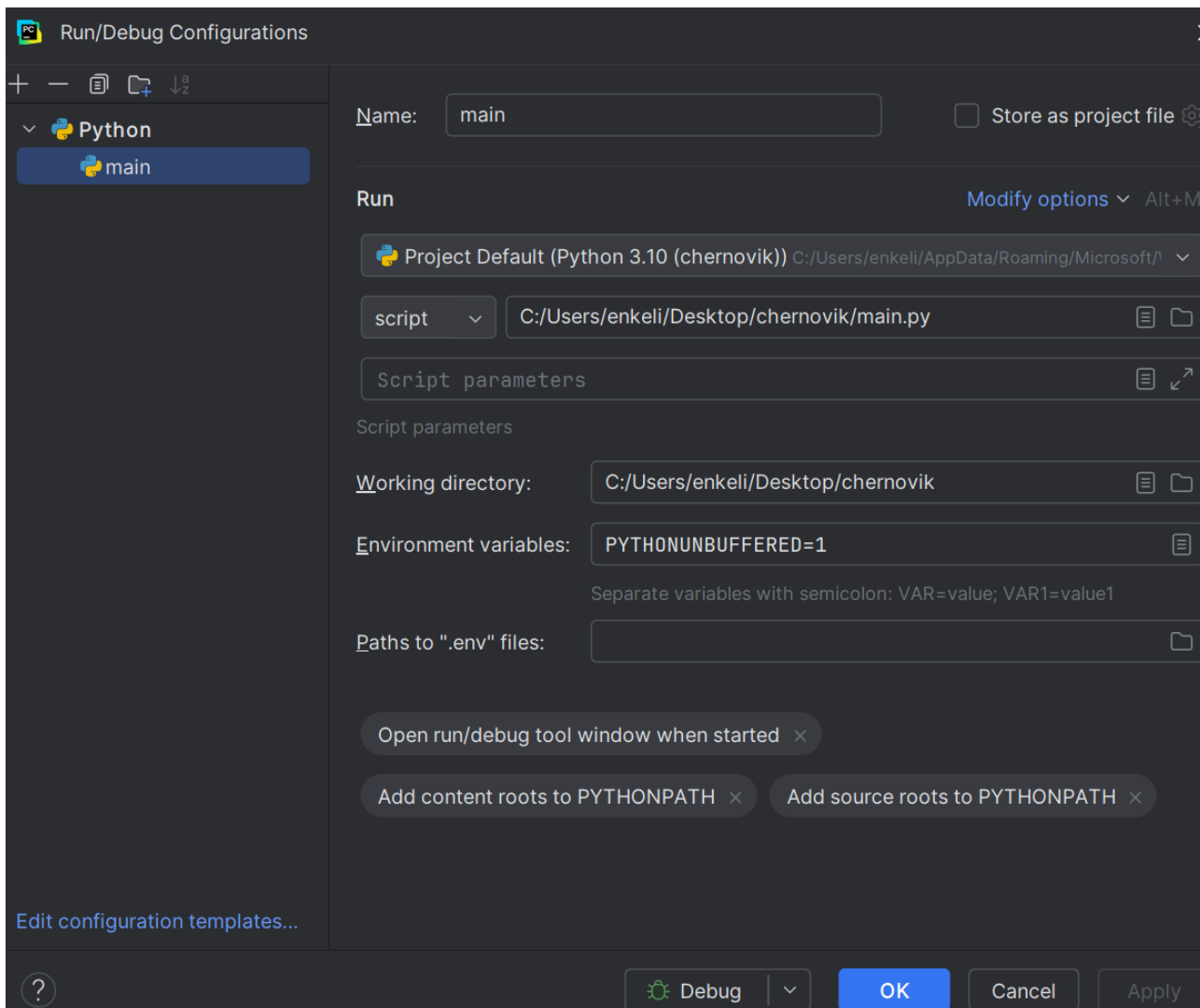
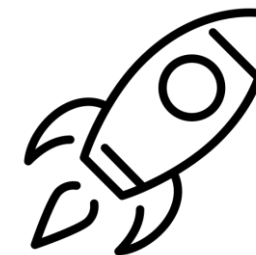
Панель отладки



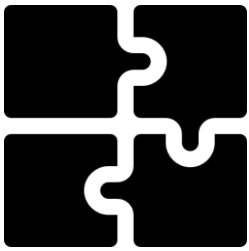
- Variables – просмотр значений переменных
- Frames – стек вызовов
- Watches – наблюдения за выражениями
- Пошаговое выполнение: Step Over / Step Into / Step Out / Step Into My Code
- Force Run to Cursor – ускоренное выполнение до курсора



Запуск программы

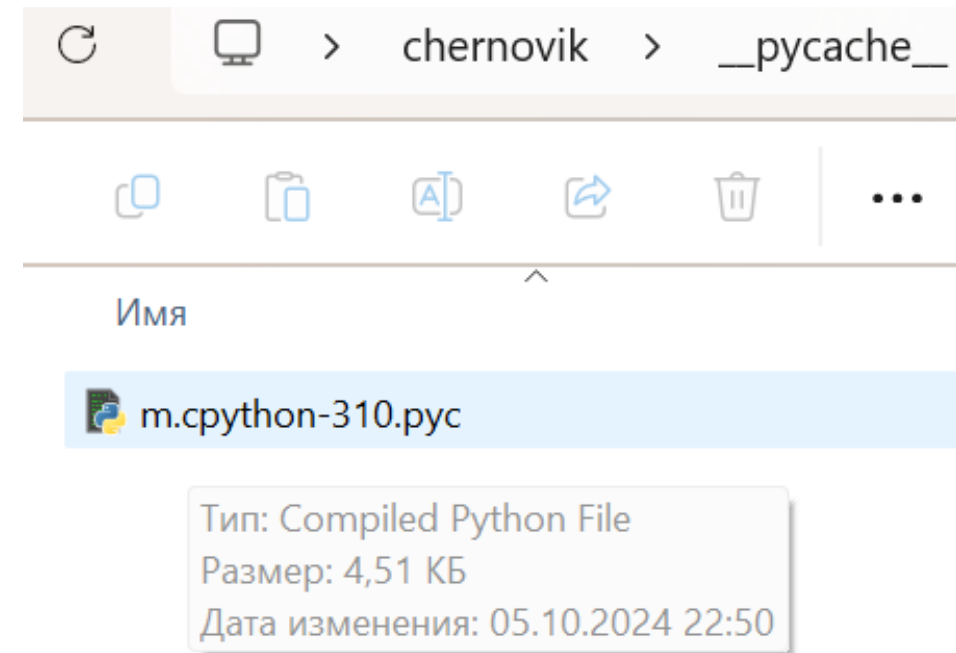


- Механизм **Run/Debug Configurations** управляет параметрами выполнения
- Можно задать несколько конфигураций для разных сценариев
- Настраиваются:
 - интерпретатор Python
 - исполняемый файл или модуль
 - аргументы командной строки (**Script parameters**)
 - рабочая директория (**Working directory**)
 - переменные окружения (**Environment variables**)
 - файл .env для автоматической подгрузки

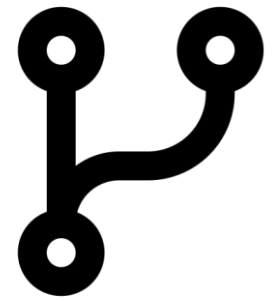


Выполнение и сборка проекта

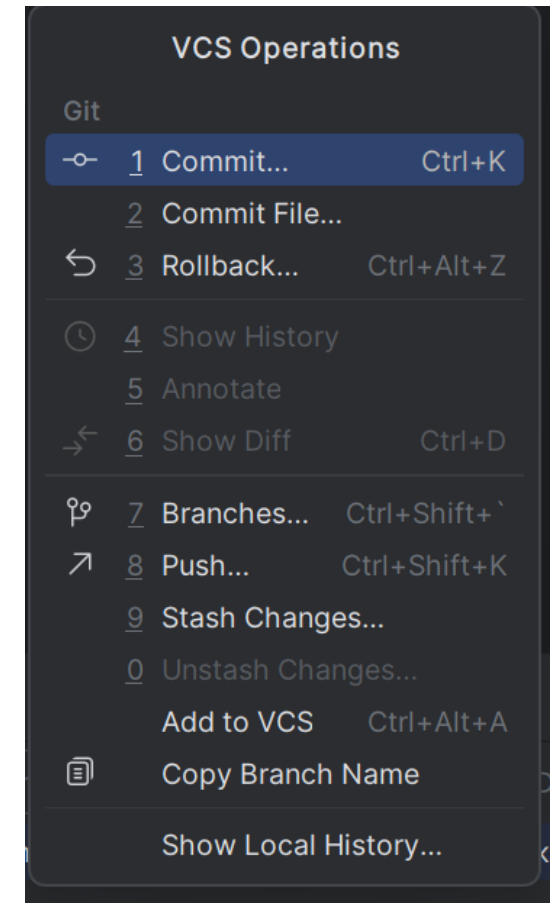
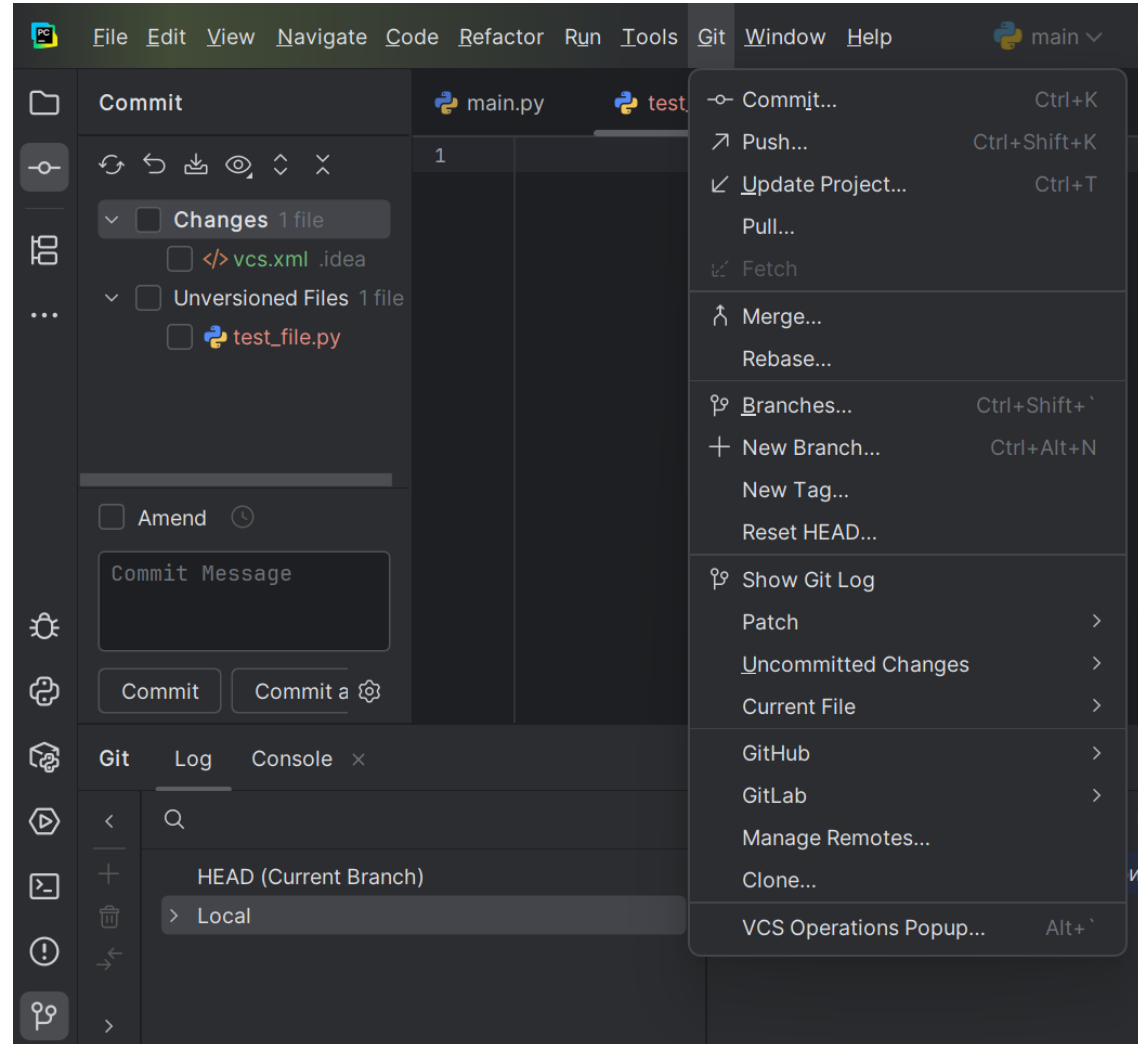
- Python — интерпретируемый язык, компиляции в привычном смысле нет
- При первом импорте создаются файлы байткода .pyc в папке __pycache__ (ускоряют запуск)
- В PyCharm под «сборкой» понимают упаковку проекта (wheel, sdist)
- Используются стандартные инструменты: setuptools, poetry, pdm

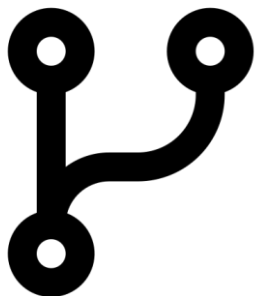


Версионирование



- Поддержка Git, Mercurial, Subversion
- Полный цикл: инициализация, коммиты (в т. ч. amend), staging по чанкам
- История изменений
- Меню Git и панель коммита
- VCS Operations Popup – быстрый доступ к действиям



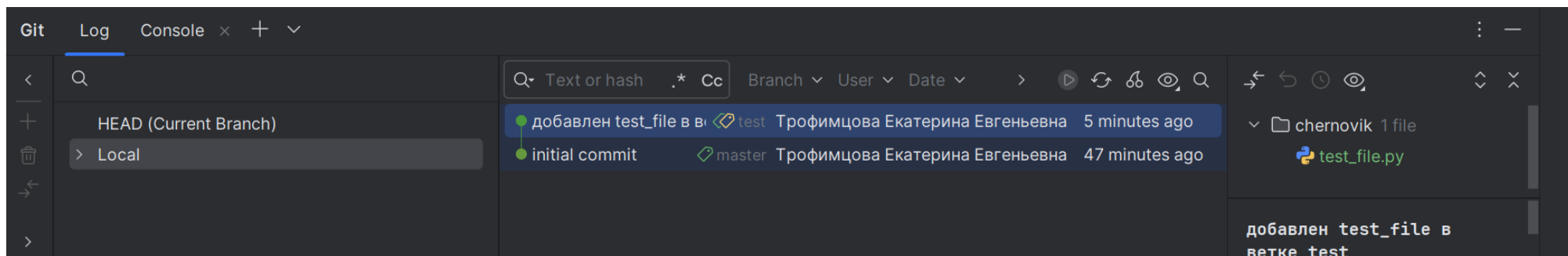
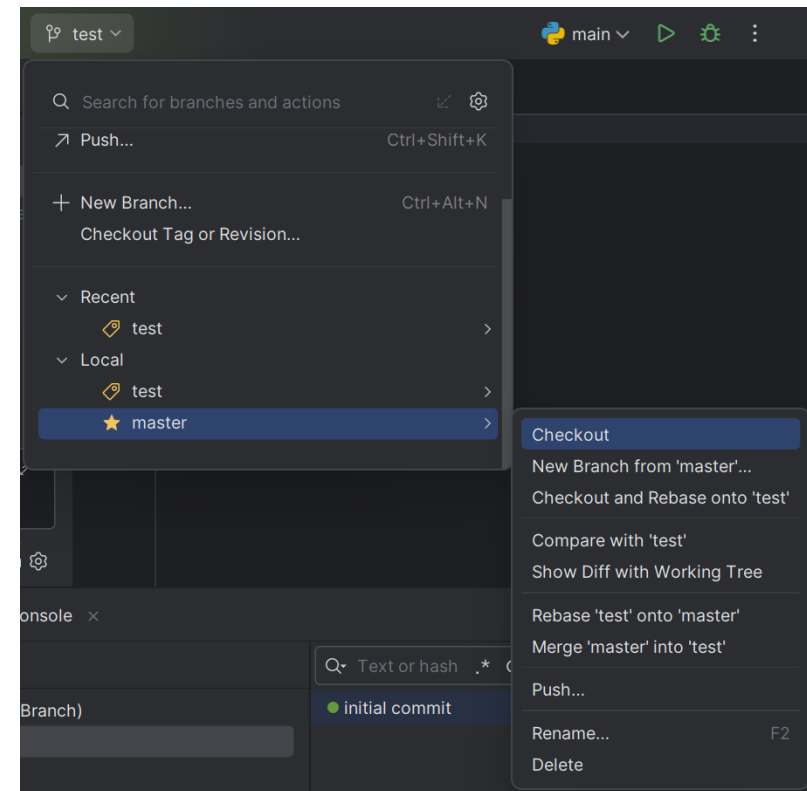


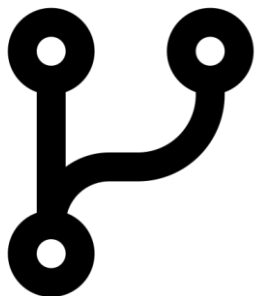
Работа с ветками и коммитами

- Работа с ветками через меню Branches: создание, переключение, слияние

- Журнал коммитов:

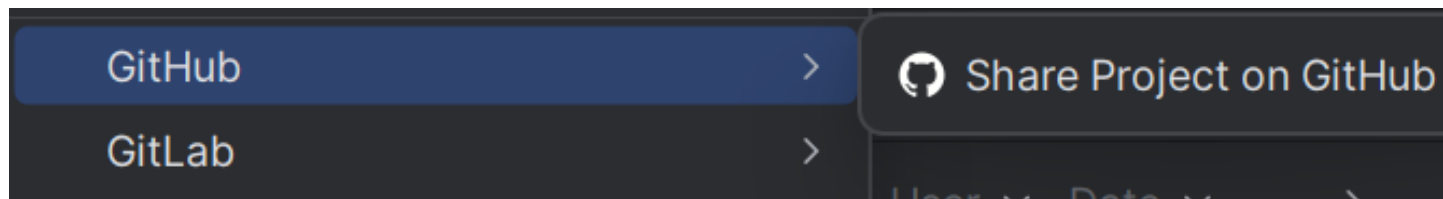
- комментарий к коммиту
- ветка, автор и время
- подробные изменения справа





Публикация в репозиторий

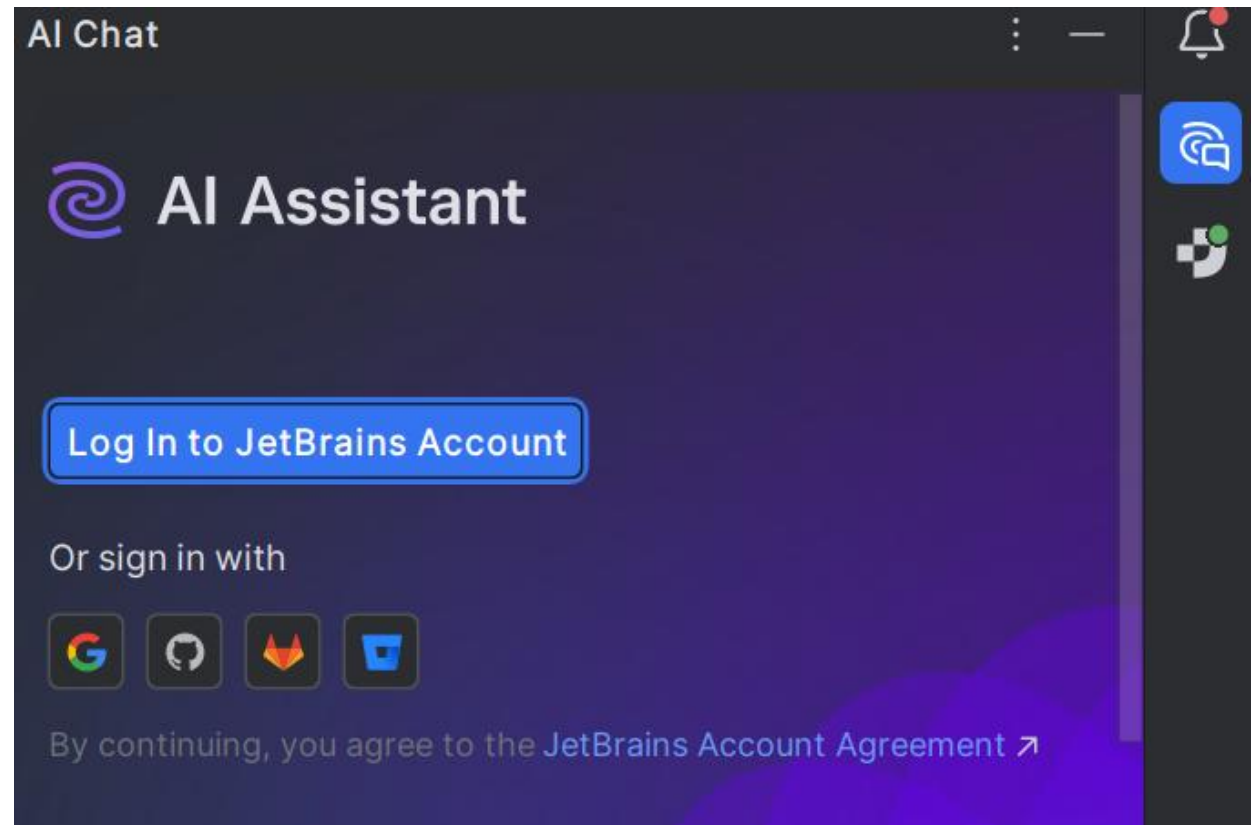
- Поддержка GitHub и GitLab
- Команда Share Project on GitHub/GitLab
- Создание нового удалённого репозитория прямо из IDE
- Автоматический Share/Push после первого коммита
- Публикация проекта



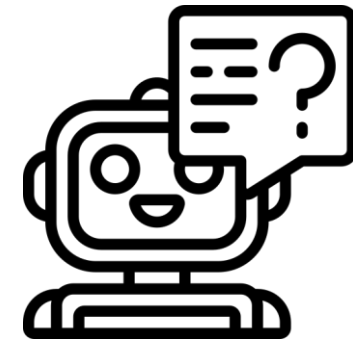
Доступ к AI-функциям



- Для работы нужен вход в аккаунт JetBrains
- Регистрация возможна через Google, GitHub, GitLab, Bitbucket
- Для активации требуется привязка банковской карты








Основные возможности AI Assistant



- Generate code from descriptions – создание кода по описанию
- Generate documentation – автоматическое формирование документации к коду
- Generate commit messages – генерация сообщений для коммитов
- Ask about project code – вопросы о проекте и анализ кода
- Explain runtime errors – объяснение ошибок выполнения и возможные решения

Features

-  Generate code from descriptions >
-  Generate documentation >
-  Generate commit messages >
-  Ask about project code >
-  Explain runtime errors >

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Ван, Х. И. PyCharm Профессиональная работа на Python : руководство / Х. И. Ван, К. Нгуен ; перевод с английского И. Л. Люско. — Москва : ДМК Пресс, 2024. — 618 с. — ISBN 978-5-93700-274-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/456716> (дата обращения: 17.09.2025). — Режим доступа: для авториз. пользователей.
2. Настройка IDE для Python: PyCharm. — Текст : электронный // Skypго : [сайт]. — URL: <https://sky.pro/wiki/python/nastrojka-ide-dlya-python-pycharm/> (дата обращения: 17.09.2025).
3. Пробуем Junie от JetBrains на реальной задаче (или как я попал в рассказ Азимова). — Текст : электронный // Хабр : [сайт]. — URL: <https://habr.com/ru/articles/904876/> (дата обращения: 22.09.2025).
4. Стуков, И. PyCharm: что это за среда разработки, как её установить и использовать / Skillbox Media / И. Стуков. — Текст : электронный // Skillbox Media : [сайт]. — URL: <https://skillbox.ru/media/code/pycharm-kak-eye-ustanovit-i-ispolzovat/> (дата обращения: 17.09.2025).
5. Pycharm - Краткое руководство - CoderLessons.com. — Текст : электронный // CoderLessons.com : [сайт]. — URL: <https://coderlessons.com/tutorials/python-technologies/uznaite-pycharm/pycharm-kratkoe-rukovodstvo> (дата обращения: 17.09.2025).
6. PyCharm: что за IDE, как установить и настроить, тарифы PyCharm. — Текст : электронный // МТС Exolve : [сайт]. — URL: <https://exolve.ru/blog/pycharm-dev-installation-usage/> (дата обращения: 17.09.2025).
7. Quick start guide | PyCharm Documentation. — Текст : электронный // JetBrains s.r.o. : [сайт]. — URL: <https://www.jetbrains.com/help/pycharm/quick-start-guide.html> (дата обращения: 22.09.2025).