**Обзор программного продукта PyCharm**

**Программный продукт:** PyCharm (от компании JetBrains)

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

**Общая характеристика**

PyCharm — это мощная и функциональная IDE, признанная одной из лучших для профессиональной разработки на Python. Она предоставляет полный набор инструментов для эффективного написания, отладки, тестирования и развертывания кода.

**Ключевые особенности и преимущества [5]:**

* **Умное редактирование кода:** Ядро IDE включает в себя продвинутый редактор с подсветкой синтаксиса, автодополнением кода, анализом кода на лету и быстрыми исправлениями ошибок.
* **Навигация по проекту:** Позволяет мгновенно переходить к объявлению класса, функции или метода, быстро находить файлы и символы в проекте.
* **Интегрированный отладчик и тестирование:** Встроенные инструменты для отладки кода и запуска тестов непосредственно из среды разработки.
* **Поддержка фреймворков и технологий:** Имеет полноценную поддержку популярных веб-фреймворков (Django, Flask), систем контроля версий (Git, SVN), баз данных, инструментов для работы с данными (Jupyter Notebook, Anaconda) и многого другого.
* **Система управления версиями:** Встроенная интеграция с Git, GitHub и другими системами, позволяющая выполнять коммиты, просматривать историю и разрешать конфликты без выхода из IDE.
* **Настраиваемость:** Огромный набор плагинов и тем оформления через встроенный Marketplace, что позволяет адаптировать среду под конкретные нужды разработчика.
* **Доступные версии:**
  + **PyCharm Professional:** Платная версия с полной функциональностью, включая поддержку веб-фреймворков и инструментов для научной разработки.
  + **PyCharm Community:** Бесплатная версия с открытым исходным кодом. Идеальна для чистого программирования на Python и обучения, но не имеет продвинутых функций для веб-разработки и работы с базами данных.

PyCharm позволяет значительно ускорить процесс разработки, снизить количество ошибок и повысить читаемость кода за счёт автоматизации рутинных операций и предоставления контекстной помощи.

**Необходимое для работы программное и аппаратное обеспечение (Requirements)** **[3]**

**Аппаратные требования (рекомендуемые):**

* **Процессор:** Специальных требований нет, однако для ускорения работы предпочтительнее использовать многоядерный процессор.
* **Оперативная память:** Желательно, чтобы общий объем RAM составлял не менее 8 ГБ, при этом для самой программы должно быть доступно 4 ГБ свободной памяти.
* **Диск:**  Потребуется около 2,5 ГБ места для установки и еще 1 ГБ под кэш. Рекомендуется использовать SSD с 5 ГБ свободного места.
* **Монитор:** Минимально поддерживаемое разрешение — 1024×768, но для удобства работы с кодом и панелями инструментов лучше использовать экран с разрешением 1920×1080.

**Программные требования:**

* **Операционная система:**
  + Microsoft Windows 10 / 11 (64- битные версии)
  + macOS 10.14 и новее
  + Linux со средой GNOME, KDE или Unity DE. Требуется наличие библиотеки Glibc версии не ниже 2.14.
* **Для выполнения Python-кода:**
  + Среда разработки поддерживает Python 2.7, а также Python 3.6 и все более поздние версии.
* **Дополнительное ПО:**
  + **Java Runtime Environment (JRE):** Требуется для работы самой IDE (обычно входит в установщик).
  + **Git / SVN:** Для интеграции с системами контроля версий.
  + **Docker:** Для использования технологии контейнеризации.

**Функции**

**Создание проекта:**

1. При запуске PyCharm пользователь видит окно приветствия, где может создать новый проект, выбрав **New Project**.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

1. Открывается диалоговое окно, где необходимо указать:
   * **Расположение проекта** на диске.
   * **Тип проекта** (чистый Python, Django, Flask и т.д.).
   * **Интерпретатор Python:** Можно выбрать уже установленный на системе, создать новый виртуальное окружение (virtualenv, pipenv, conda) или использовать Docker-контейнер. Создание виртуального окружения — это лучшая практика, которую PyCharm поддерживает «из коробки».



1. После нажатия «Create» IDE генерирует структуру папок, активирует виртуальное окружение и настраивает интерпретатор.

**Начало работы и создание файлов:**

Перед вами открывается чистое рабочее пространство проекта. Чтобы начать кодировать, нужно создать Python-файл. Для этого:

* + Кликните правой кнопкой мыши на имя папки проекта в боковой панели **Project**.
  + Выберите **New** -> **Python File**.
  + Введите имя файла. PyCharm автоматически добавит расширение .py.

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Теперь в центральном редакторе у вас есть пустой файл, готовый для написания кода. Вы можете начать вводить свои первые команды [2].

**Кодирование:**

Функциональность кодирования в PyCharm реализована через интеллектуальный редактор, который выступает в роли активного помощника разработчика.

**1. Завершение кода**

Инструмент предназначен для ускорения написания кода за счет анализа контекста, импортированных библиотек и типов переменных.

* **Базовое завершение (**Ctrl + Space**)** отображает список всех доступных в текущем контексте методов, классов и переменных. Например, при вводе pri предлагается вариант print.
* **Умное завершение (**Ctrl + Shift + Space**)** является ключевой функцией. Оно фильтрует предложения по типу. После ввода my\_list. отображаются только методы, применимые к спискам (append, pop).

**2. Навигация по коду**

Инструменты навигации позволяют мгновенно перемещаться по кодовой базе.

* **Переход к определению (**Ctrl + ЛКМ**/**Ctrl + B**)** является основной операцией. Наведение на символ с зажатой Ctrl и щелчок мыши (или Ctrl+B) открывает файл и строку, где находится исходный код этого символа.
* **Поиск класса (**Ctrl + N**)** и **поиск файла (**Ctrl + Shift + N**)** используются для быстрого нахождения элементов в проекте.
* **Универсальный поиск (**Double Shift**)** открывает окно для поиска по всему проекту, включая классы, файлы и настройки IDE.

**3. Рефакторинг**

Набор функций для безопасного изменения структуры кода.

* **Переименование (**Shift + F6**)** позволяет изменить имя переменной, функции или класса во всех местах их использования в проекте.
* **Извлечение переменной (**Ctrl + Alt + V**)** заменяет выделенное выражение на новую переменную.
* **Извлечение метода (**Ctrl + Alt + M**)** преобразует выделенный блок кода в отдельную функцию.

**4. Встроенный отладчик**

Инструмент для пошагового выполнения программы и анализа ее состояния.

* **Точки останова (Breakpoints)** устанавливаются щелчком на левом поле редактора напротив нужной строки.
* **Запуск в режиме отладки** выполняется с помощью Shift + F9.
* **Шаги отладки:**
  + **Step Over (**F8**)** выполняет текущую строку, переходя к следующей.
  + **Step Into (**F7**)** позволяет войти внутрь вызываемой в строке функции.
  + **Step Out (**Shift + F8**)** выполняет оставшийся код текущей функции и возвращается к месту ее вызова.
* **Просмотр переменных** осуществляется в панели Debugger во вкладке Variables, где отображаются их текущие значения.

**Форматирование кода:**

**1. Автоматическое форматирование кода (**Ctrl + Alt + L**)**

Функция автоматически приводит код к единому стилю: расставляет отступы, пробелы вокруг операторов и правильные переносы строк. Форматирование применяется к выделенному фрагменту или ко всему файлу [1].

**2. Интеграция с PEP 8**

IDE постоянно анализирует код на соответствие стандарту PEP 8.

* Нарушения подчеркиваются волнистой серой линией.
* **Быстрое исправление (**Alt + Enter**)**, применяемое к подчеркнутой строке, позволяет автоматически устранить проблему (например, добавить пропущенные пустые строки). Это способствует практическому изучению стандартов кодирования.

**3. Настройка форматтера и интеграция с внешними инструментами**

* **Встроенные настройки** стиля находятся по пути File -> Settings -> Editor -> Code Style -> Python. Там настраиваются размер отступа, тип кавычек и правила переноса строк.
* **Интеграция с**black является распространенной практикой. После установки black в виртуальное окружение, он настраивается как File Watcher (в разделе Tools). Это позволяет автоматически форматировать код при каждом сохранении файла, гарантируя единообразие стиля.

**4. Автоматические действия при сохранении**

**Настройка осуществляется по пути:** File -> Settings -> Tools -> Actions on Save.

В этом разделе активируются автоматические операции, выполняемые при сохранении файла (Ctrl+S). Часто включаются опции:

* Reformat code (для форматирования кода).
* Optimize imports (для упорядочивания и удаления неиспользуемых импортов).

Это позволяет поддерживать код в чистоте и соответствии стилю без дополнительных ручных действий.

Активное использование инструментов кодирования и форматирования превращает среду разработки в активного участника процесса, который берет на себя рутинные задачи и способствует написанию чистого, корректного кода.

**Заключение**

PyCharm — это мощная и эффективная среда разработки для Python, которая сочетает интеллектуальные возможности редактирования кода с удобными инструментами отладки. Благодаря автоматизации рутинных задач и поддержке современных технологий, она подходит разработчикам любого уровня и значительно ускоряет процесс создания качественных приложений.

В ходе выполнения группового задания с Трофимцовой Е.Е. мною были разработаны содержание и оформление слайдов 2–10 итоговой презентации.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Ван, Х. I. PyCharm Профессиональная работа на Python : руководство / Х. I. 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 // Skypro : [сайт]. — URL: <https://sky.pro/wiki/python/nastrojka-ide-dlya-python-pycharm/> (дата обращения: 17.09.2025).
  3. 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).
  4. 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).
  5. 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).