

Предисловие

Язык программирования Питон (Python) на современном этапе развития средств разработки программных приложений является одним из массовых и популярным среди сообщества IT-специалистов. Широкое использование Python в учебном процессе в высших, средних специальных заведениях, а также в школах требует специального методического обеспечения учебного процесса, так как информация по методам программирования на Python носит, как правило, разрозненный и методически непродуманный характер (хотя подобной информации существует огромное множество). На начальных этапах освоения средств языка Python перед обучающимися остро встает вопрос выбора среды программирования, на основе которой осуществляется интегрированная разработка приложений на языке Python. Наиболее часто выбор приводит к интегрированной среде разработки (Integrated development environment, сокращенно — IDE) под названием PyCharm, которая является кросс-платформенной средой разработки с реализацией в Windows, MacOS и Linux. Среда PyCharm представляет собой достаточно сложную программу, которая обеспечивает выполнение множества действий по разработке программного обеспечения (ПО). Среда PyCharm интегрирует утилиты (вспомогательные системные программы):

- текстовый редактор;
- транслятор (компилятор и интерпретатор);
- отладчик;
- средства автоматизации сборки.

Опыт преподавания основ программирования показывает, что на первоначальных этапах изучения и освоения технологий разработки прикладного ПО обучающийся сталкивается с некоторыми сложностями работы в среде программирования, в частности в среде PyCharm. Эти сложности обусловлены низким уровнем знания английского языка, недостаточными навыками работы с компьютером и современным программным обеспечением. Для преодоления подобных трудностей на начальных ста-

дях освоение среды PyCharm и предназначено настоящее руководство для установки и эффективной работы со средствами IDE, ориентированными на разработку программ на языке Python. С этой точки зрения среда PyCharm содержит все необходимые средства для профессиональной разработки прикладного ПО. Использование PyCharm в процессе разработки программы позволяет обеспечить *высокую продуктивность*, обусловленную освобождением от рутинных операций и сосредоточением на выполнении более важных этапов, что приводит к экономии времени, а также *повышение качества кода*, что позволяет писать красивый программный код с одновременным контролем его качества путем проверки соответствия требованиям PEP8 — руководства по написанию кода на языке Python. *Помощь при написании кода* обеспечивается встроенным механизмом анализа программного кода, поиском ошибок и их быстрым исправлением, удобной навигацией по коду и другими функциями.

Эффективная разработка программ на языке Python часто реализуется с помощью интерактивной оболочки Jupyter Notebook, которая является развитием проекта IPython путем введения дополнительного командного синтаксиса, подсветки кода и пр. В отличие от IPython, система Jupyter Notebook в качестве «ядра» допускает использование Python, Java, R, Julia, MathLAB и других языков (всего более ста языков программирования). Использование Jupyter Notebook позволяет не только проводить интерактивную разработку программного кода, но также представлять данные и проводить их анализ. Весьма важным и удобным элементом среды можно считать ноутбуки (блокноты), которые являются файлами с расширением `ipynb` и которые позволяют объединять повествовательный текст, визуализацию, математические уравнения, мультимедиа, исходный код, входные и выходные данные в единый документ. Файлы `ipynb` допускается экспортировать в форматы `pdf` и `html`.

На основе Jupyter Notebook разработан сервис Google Colaboratory (сокращенно Google Colab), ориентированный на упрощение исследований в области машинного и глубокого обучения. Google Colab является облачным сервисом и предназначается в первую очередь для реализации образовательных проектов, анализа данных и изучения языка Python. Данный сервис позволяет организовать глубокое обучение нейросетей. Google Colab можно использовать на бесплатной основе, что делает его привлекательным для широкой аудитории. Сервис позволяет обеспечивать

пользователю применение таких важных ресурсов, как центральный процессор, графический процессор и тензорный процессор. Центральный процессор выполняет операции по обработке данных. Графический процессор реализует быструю обработку за счет параллельных вычислений путем использования видеокарты. Тензорный процессор (разработка Google) предназначен для тренировки нейросетей. Среда позволяет использовать широкий спектр библиотек языка Python: Keras, Tensor, Flow, Numpy, Pandas, Matplotlib и других. Однако сервис Google Colab не гарантирует предоставление ресурсов. Кроме того, на использование предоставляемых ресурсов действуют лимиты, например бесплатное использование ресурсов ограничено временем не более 12 часов. Более подробно ограничения представлены в разделе 10. Блокноты Google Colab можно загружать и сохранять на широком спектре устройств.

Введение

В следующих разделах учебного пособия первоначально рассмотрены основные правила работы со средой разработки PyCharm, которые необходимы для выполнения цикла практических работ по программированию на языке Python. Прежде чем перейти непосредственно к описанию теоретических сведений и практическим рекомендациям по выполнению практических работ, необходимо уточнить некоторые детали.

1. Предполагается, что цикл практических работ реализован с использованием алгоритмического языка Python версии 3.5.2, которая на момент постановки практических работ являлась одной из последних версий языка программирования. При этом следует иметь в виду, что работа с другой версией языка Python сопряжена с возможностью появления некоторых различий, что свойственно для различных версий языка Python.

2. В цикле практических работ использована свободно распространяемая интегрированная среда разработки IDE (Integrated Development Environment) PyCharm Community Edition. Скачать ее можно по адресу

<http://www.jetbrains.com/pycharm/download>

Среда доступна для Windows, Linux, MacOS. Однако, если вам не нравится IDE, выбранная авторами, вы легко можете выбрать для себя другую среду разработки. Методика работы в различных средах отличается мало, более важным является соответствие версии интерпретатора Python.

3. В состав большинства IDE по умолчанию не входит интерпретатор языка Python. Его необходимо скачать и установить заранее. Сделать это можно с официального сайта

<https://www.python.org/downloads/>

4. Практические работы ориентированы на использование среды PyCharm, запущенной в ОС Windows. Для Linux и MacOS возможны свои особенности и отличия.

5. Рассмотренный материал не охватывает всего разнообразия возможностей, предоставляемых языком Python, как и не описывает всех методов эффективного написания программ на этом языке. За дополнительной информацией вы можете обратиться к книгам самоучителям, в том числе к тем, которые упоминаются в методических указаниях. Также крайне необходимо умение правильно искать информацию с помощью поисковых систем в Интернете, что помогает оперативно решать возникающие трудности.

6. В процессе выполнения практических работ, как правило, требуется делать отчеты по проделанной работе. В отчет обычно входит схема алгоритма. Для удобного создания схемы в электронном виде с возможностью ее дальнейшего экспорта в векторный или растровый формат изображения вы можете воспользоваться бесплатным сервисом www.draw.io в виде онлайн-редактора на сайте или офлайн-надстройки для браузера (тестировалось для Google Chrome).