НОЧУ ДПО УЦ «Сетевая Академия»

УТВЕРЖДАЮ
Директор НОЧУ ДПО УЦ «Сетевая Академия»
_____/Шикова Ю.В./

Образовательная программа
дополнительного профессионального образования
(повышения квалификации)
«Программирование на Python: Продвинутый уровень (Python3_Adv_Advanced Python 3)»

Содержание

Описание образовательной программы

Настоящая образовательная программа повышения квалификации (далее – Программа) разработана в соответствии с:

1. Федеральным законом от 29 декабря 2012 г. N 273-ФЗ «Об образовании в Российской Федерации»
2. Приказом Минобрнауки России от 1 июля 2013 г. N 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»
3. Уставом НОЧУ ДПО УЦ «Сетевая Академия»

Структура Программы включает цели, планируемые результаты обучения, учебный план, календарный учебный график, рабочую программу, организационно-педагогические условия, формы аттестации и оценочные материалы.

Цели Программы содержат описание целевой аудитории, целей обучения и необходимых начальных знаний и навыков слушателей.

Планируемые результаты обучения представлены в виде перечня профессиональных компетенций в рамках имеющейся квалификации (с отсылкой к профессиональному стандарту), качественное изменение которых осуществляется в результате обучения.

Учебный план определяет перечень, трудоемкость, последовательность и распределение модулей, иных видов учебной деятельности обучающихся и формы аттестации.

Календарный учебный график определяет основные параметры учебного процесса при организации занятий по освоению настоящей Программы, включая формы обучения, расписание занятий очных групп и т.п.

Рабочая программа раскрывает рекомендуемую последовательность изучения разделов (модулей).

Описание организационно-педагогических условий реализации Программы определяет организационные и методические требования НОЧУ ДПО УЦ «Сетевая Академия» к организации и проведению обучения по Программе.

Формы аттестации и оценочные материалы определяют формы проведения промежуточной и итоговой аттестации по Программе и форму учебно-методических материалов, необходимых для проведения указанных видов аттестации.

Цели программы

Данная Программа предназначена для:

- разработчиков, администраторов операционных сетей и баз данных, аналитиков данных, как имеющих предварительный опыт программирования на каком-либо языке, так и без соответствующего опыта.

Целью обучения является формирование у слушателей знаний и и практических навыков продвинутого программирования на языке Python: использования продвинутых инструментов, библиотек и возможностей языка.

Для изучения данной Программы рекомендуется обладать следующими знаниями и навыками:

- уметь использовать управляющие конструкции: циклы, условные операции; коллекции: списки словари, кортежи; строковые операции; операции с датой и временем; определение и использование функций, задавать обработку исключительных ситуаций (exceptions), обработка файлов.

Планируемые результаты обучения

Реализация Программы направлена на повышение профессионального уровня в рамках имеющейся квалификации, определяемой профессиональным стандартом «06.001 Программист», утвержденным Приказом Минтруда России от 18.11.2013 N 679н (ред. от 12.12.2016) "Об утверждении профессионального стандарта "Программист"

Результатами обучения по Программе станут знания и умения, соответствующие следующим обобщенным трудовым функциям указанного профессионального стандарта:

- Разработка и отладка программного кода.
- Проверка работоспособности и рефакторинг кода программного обеспечения.
- Разработка требований и проектирование программного обеспечения.

Совершенствуемые компетенции в соответствии с трудовыми функциями профессионального стандарта:

Компетенция	**Содержание компетенции** Трудовые функции	**Код**
Разработка и отладка программного кода	Формализация и алгоритмизация поставленных задач	A/01.3
	Написание программного кода с использованием языков программирования, определения и манипулирования данными	A/02.3
	Оформление программного кода в соответствии с установленными требованиями	A/03.3
	Проверка и отладка программного кода	A/05.3
Проверка работоспособности и рефакторинг кода программного обеспечения	Разработка процедур проверки работоспособности и измерения характеристик программного обеспечения	B/01.4
	Разработка тестовых наборов данных	B/02.4
	Проверка работоспособности программного обеспечения	B/03.4
	Рефакторинг и оптимизация программного кода	B/04.4
Разработка требований и проектирование программного обеспечения	Анализ требований к программному обеспечению	D/01.6
	Проектирование программного обеспечения	D/03.6

После обучения слушатель сможет:

- Настраивать среду разработки Python и создавать простые программы.
- Применять среду и блокноты Jupyter (IPython notebooks).
- Использовать модуль Collections.
- Использовать возможности функционального программирования: lambda-выражения.
- Применять функции к каждому элементу коллекции (map), отбирать элементы по условию (filter).

- Использовать продвинутые возможности функций сортировки.
- Искать и вычленять нужные элементы текста при помощи регулярных выражений.
- Работать с базами данных.
- Работать с данными в текстовых файлах в форматах CSV, JSON и XML.
- Писать программы, используя объектно-ориентированный стиль программирования.
- Тестировать корректность работы своих программ.
- Использовать отладку для поиска логических ошибок в своих программах.
- Работать с различными кодировками текста, включая кодировки Unicode.

Учебный план

Учебный план Программы определяет перечень, трудоемкость, последовательность и распределение модулей, иных видов учебной деятельности обучающихся и формы аттестации.

№ п/п	Наименование разделов (модулей)	Всего, час	В том числе		Форма аттестации
			Лекции	Практические занятия	
1.	Тетради Jupyter (IPython notebooks).	3	2,25	0,75	Опрос, практические занятия
2.	Продвинутые возможности Python для работы с коллекциями.	5	4	1	Опрос, практические занятия
3.	Регулярные выражения.	3	2,25	0,75	Опрос, практические занятия
4.	Работа с данными: базы данных, файлы CSV, JSON, XML.	3	2,25	0,75	Опрос, практические занятия
5.	Классы и объекты.	4	3	1	Опрос, практические занятия
6.	Тестирование и отладка.	3	2,25	0,75	Опрос, практические занятия
7.	Кодировки и Unicode.	2	1,5	0,5	Опрос, практические занятия
8.	Итоговая аттестация	1	-	1	Тестирование
9.	**Итого:**	**24**	**17,5**	**6,5**	

Допускается формирование индивидуального учебного плана для каждого слушателя в пределах осваиваемой Программы в порядке, установленном Положением об организации образовательного процесса в НОЧУ ДПО УЦ «Сетевая Академия».

Календарный учебный график

Учебный год: круглогодичное обучение.

Продолжительность Программы: 40 академических часов.

Форма организации образовательного процесса: очная, очно-заочная (вечерняя) и заочная формы обучения, в том числе, с применением дистанционных образовательных технологий и электронного обучения.

Сменность занятий (при очной форме обучения): I смена.

Количество учебных дней в неделю при очном обучении: 3 дня.

Начало учебных занятий: 9.30

Окончание учебных занятий: 17.00

Продолжительность урока: 45 минут (1 академический час).

Продолжительность перемен: 15 минут, перерыв на обед – 60 минут.

Расписание занятий для очных групп:

	№ урока	**Время**
Конкретный день недели согласовывается во время учебного процесса	1-2	09:30 - 11:00
	3-4	11:15 - 12:45
	5-6	13:45 - 15:15
	7-8	15:30 - 17:00

Рабочая программа

Модуль 1: Тетради Jupyter (IPython notebooks).

- Введение в IPython notebook.
- *Лабораторная работа 1: Создание первой тетради Jupyter.*
- *Лабораторная работа 2: Экспериментируем с IPython notebook.*
- Упрощенный язык разметки *markdown*.
- «Магические» команды (*magic commands*).
  - Automagic
  - Autosave
  - Команды работы с директориями.
  - Обращение к трём последним строкам ввода и вывода.
  - История команд.
  - Работа с закладками.
  - Переменные окружения.
  - Загрузка и запуск программного кода из файлов.
  - Выполнение команд оболочки и операционной системы.
  - Дополнительные «магические» команды.
- Получение сведений из справочной системы.

Модуль 2: Продвинутые возможности Python для работы с коллекциями.

- Продвинутые списковые включения (list comprehensions).
  - Краткое повторение основ списковых включений.
  - Списковые включения со многими циклами.
  - *Лабораторная работа 3: «Подбрасывание пяти игральных кубиков».*
- Модуль collections.
  - Именованные кортежи (named tuple).
  - Словари с указанием значения по умолчанию (defaultdict).
  - *Лабораторная работа 4: Создание словаря defaultdict.*
  - Счётчики. Класс collections.Counter.
  - *Лабораторная работа 5: Использование объекта класса Counter.*
- Отображение и фильтрация.
  - Map (функция, итерируемая последовательность).
  - Filter (логическая-функция, итерируемая последовательность).
- Lambda-функции.
  - Использование lambda-функций с map() и filter().
- Изменяемые (mutable) и неизменяемые (immutable) объекты встроенных типов.
  - Строки (str).
  - Списки (list).
- Сортировка.
  - Сортировка элементов списка «на месте» без образования копии. Метод sort().
  - Функция sorted().

- *Лабораторная работа 6: Переход от использования метода list.sort() к функции sorted (итерируемая последовательность)*
- Сортировка последовательности, состоящей из последовательностей.
- Сортировка последовательности, состоящей из словарей.
- Распаковка последовательностей для передачи в качестве набора аргументов в вызове функции.
- *Лабораторная работа 7: Преобразование строки в объект datetime.date.*

- Модули и пакеты.
  - Модули.
  - Пакеты.
  - Путь поиска модулей и пакетов.

Модуль 3: Регулярные выражения (regular expressions).

- Синтаксис регулярных выражений.
- Использование регулярных выражений.
- Ссылки на группы (backreference).
- Средства Python для поддержки регулярных выражений.

Модуль 4: Работа с данными: базы данных, файлы CSV, JSON, XML.

- Реляционные базы данных.
- Документ PEP 0249: Интерфейс программиста (API) Python для работы с базами данных (версия 2.0).
- Модуль PyMySQL.
- Возвращение словарей (dictionary) вместо кортежей (tuple).
- sqlite3
- *Лабораторная работа 8: Выполнение оператора SELECT в базе данных sqllite3.*
- Передача параметров.
- Размещение базы данных SQLite в памяти.
- Выполнение нескольких запросов сразу.
- *Лабораторная работа 9: Вставка данных (insert) в таблицу базы данных.*
- Текстовые файлы с разделителями (CSV-comma separated values).
  - Чтение из csv-файла.
  - Поиск данных в csv-файле.
  - *Лабораторная работа 10: Сравнение данных в csv-файле.*
  - Создание нового файла csv.
  - Варианты формата CSV.
- Получение данных с веб-сайтов.
  - Пакет requests.
  - Пакет beautiful soup.
  - XML.
  - *Лабораторная работа 11: Применение модулей requests и Beautiful Soup.*
  - JSON.

Модуль 5: Классы и объекты.

- Атрибуты.
- Поведение объектов.
- Понимание отличия классов от объектов. Создание своих собственных классов.
- Атрибуты и методы.
- *Лабораторная работа 12: Добавление метода roll() к классу Die.*
- Приватные атрибуты.
- Свойства (properties).
- *Лабораторная работа 13:* Свойства. Объект, который отслеживает свою историю.
- Снабжение классов документацией.
  - Использование строк документации.
  - *Лабораторная работа 14*: *Снабжение класса Die строкой документации.*
- Наследование.
  - Перекрытие метода (overriding a method).
  - Расширение класса (extending a class).
  - *Лабораторная работа 15: Расширение класса Die.*
  - Расширение метода.
  - *Лабораторная работа 16*: Расширение метода roll().
  - Статические (static) методы.
  - Атрибуты класса и методы класса. Подклассы.
  - Абстрактные классы и методы.
  - Понимание декораторов (decorators).

Модуль 6: Тестирование и отладка.

- Тестирование производительности (performance).
  - time.perf_counter
  - Модуль timeit.
- Модуль unittest.
  - Файлы с тестами unittest.
  - *Лабораторная работа 17: Исправление неправильно работающей функции.*
  - Специальные возможности unittest. Методы testCase.
  - Методы assert.

Модуль 7: Кодировки и Unicode.

- Биты и байты.
- Шестнадцатеричные числа.
- *Лабораторная работа 18: Использование функций преобразования hex(), bin(), ord(), chr(), int().*
- Кодировки.
  - Кодировки текста.
  - Кодирование и декодирование файлов в языке Python.
  - Преобразование из кодировки Windows в кодировку Unicode (UTF-8).
  - *Лабораторная работа 19: Нахождение проблем с кодировкой.*

Организационно-педагогические условия реализации Программы

При реализации Программы применяется форма организации образовательной деятельности, основанная на модульном принципе представления содержания образовательной программы и построения учебных планов, использовании различных образовательных технологий, в том числе дистанционных образовательных технологий и электронного обучения.

Организационные условия реализации программы в разных формах обучения регулируются следующими локальными нормативными актами:

- Положение об организации образовательного процесса в НОЧУ ДПО УЦ «Сетевая Академия».
- Положение о порядке применения электронного обучения, дистанционных образовательных технологий в НОЧУ ДПО УЦ «Сетевая Академия».

Учебные материалы по Программе включают: рабочую программу, раздаточные материалы по курсу, методические материалы по курсу, данные примеров по курсу. Учебное пособие по Программе выдается слушателям в бумажном или электронном виде в зависимости от формы обучения в порядке, установленном Положением о библиотеке в НОЧУ ДПО УЦ «Сетевая Академия».

Занятия по Программе проводятся преподавателями, предварительно подтвердившими свою квалификацию. В числе базовых требований ко всем преподавателям – требование обязательного прохождения программы «Андрагогика. Эффективное обучение взрослых» в форме учебного курса и пробной лекции, а также сдачи технических сертификационных тестов по продукту или технологии, рассматриваемым в курсе.

Формы аттестации и оценочные материалы

Освоение Программы сопровождается промежуточной аттестацией обучающихся в формах, определенных учебным планом, и в порядке, установленном Положением об организации образовательного процесса в НОЧУ ДПО УЦ «Сетевая Академия».

Освоение Программы завершается итоговой аттестацией обучающихся в форме, определенной учебным планом, и в порядке, установленном Положением об организации образовательного процесса в НОЧУ ДПО УЦ «Сетевая Академия».

Слушателям, успешно освоившим соответствующую Программу и прошедшим итоговую аттестацию, выдается удостоверение о повышении квалификации на бланке, образец которого самостоятельно устанавливается организацией.

Слушателям, не прошедшим итоговой аттестации или получившим на итоговой аттестации неудовлетворительные результаты, а также лицам, освоившим часть Программы и (или) отчисленным из организации, выдается справка об обучении или о периоде обучения по образцу, самостоятельно устанавливаемому организацией.

Оценочные материалы для промежуточной аттестации по Программе разрабатываются в форме лабораторных работ и/или контрольных вопросов после изучения каждого модуля.

Оценочные материалы для итоговой аттестации по Программе разрабатываются в форме теста.

Пример материалов для итоговой аттестации

1. **Вопрос:** Дан фрагмент кода

```
import re
pattern1= "weight:\s*(\d+)\s+age:\s*(\d+)"
datastring="weitgt:86 age:49"
match_obj=re.match(pattern1, datastring)
```

Перечислены варианты выражений, позволяющих выделить из строки целое число, содержащее возраст человека (49 в этом примере)
Какое выражение НЕ ПОДХОДИТ для данной цели?

Варианты ответов:

A. int (match_obj[2])
B. int (match_obj.group(2))
C. int (match_obj.groups()[1])
D. int (match_obj.groups()[2])

Правильные ответы: C

2. **Вопрос:** Дан список people пар (имя, вес)

people= [('Вася' , 110) , ('Коля' , 95) , ('Аня', 50)]

Какое из выражений создаст копию списка, упорядоченную по убыванию веса, оставив исходный спискок people без изменений?

Варианты ответов:

A. *sorted_by_weight = sorted(people, key=lambda item: item[1])*
B. *sorted_by_weight = sorted(people, key=lambda item: item[1] , reverse=True)*
C. *sorted_by_weight = people.sort(key=lambda item: item[1] , reverse=True)*
D. *sorted_by_weight = people.sort(key=lambda item: item[1])*

Правильные ответы: B

3. **Вопрос:** Какое предопределенное имя используется для специального метода, который вызывается после создания объекта и до того, как ссылку на него возвращают вызывающей подпрограмме?

Варианты ответов:

A. __constructor__()
B. __initialize__()
C. assign()
D. __init__()

Правильные ответы: D