1. **Паспорт Образовательной программы «Основы программирования на Python»**

|  |  |
| --- | --- |
| **Версия программы** | 1 |
| **Дата Версии** | 19.10.2020 |

1. **Сведения о Провайдере**

|  |  |  |
| --- | --- | --- |
| 1.1 | Провайдер | Образовательная автономная некоммерческая организация дополнительного профессионального образования "Высшая школа информационных технологий и безопасности" (ОАНО ДПО «Выштех») |
| 1.2 | Логотип образовательной организации |  |
| 1.3 | Провайдер ИНН | 7703434727 |
| 1.4 | Ответственный за программу ФИО | Лагутина Мария Андреевна |
| 1.5 | Ответственный должность | Академический директор |
| 1.6 | Ответственный Телефон | +79096389557 |
| 1.7 | Ответственный Е-mail | m.lagutina@hackeru.com |

1. **Основные Данные**

|  |  |  |
| --- | --- | --- |
| **№** | **Название** | **Описание** |
| 2.1 | Название программы | Основы программирования на Python |
| 2.2 | Ссылка на страницу программы | Ссылка на программу - intropythoncourse.ru |
| 2.3 | Формат обучения | Электронное обучение |
|  | Подтверждение от ОО наличия возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа | Подтверждаем наличие возможности реализации образовательной программы с применением электронного обучения и (или) дистанционных образовательных технологий с возможностью передачи данных в форме элементов цифрового следа |
| 2.4 | Уровень сложности | Базовый |
| 2.5 | Количество академических часов | 72 |
|  | Практикоориентированный характер образовательной программы: не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и (или) выполнению практических заданий в режиме самостоятельной работы (кол-во академических часов) | Не менее 50 % трудоёмкости учебной деятельности отведено практическим занятиям и выполнению практических заданий в режиме самостоятельной работы:  45 ак. часов практических занятий, в том числе:  36 практико-ориентированных заданий и кейсов (см. п. 7), что составляет не менее 50% трудоемкости учебной деятельности. |
| 2.6 | Стоимость обучения одного обучающегося по образовательной программе, а также предоставление ссылок на 3 (три) аналогичные образовательные программы иных организаций, осуществляющих обучение, для оценки объективности стоимости или обоснование уникальности представленной образовательной программы в случае отсутствия аналогичных образовательных программ на рынке образовательных услуг | 25 000 руб.  Ссылки на 3 аналогичные программы:   1. Программа повышения квалификации “Программирование на Python”, 40 ак.часов, стоимость 20000 руб. <https://busedu.hse.ru/catalog/338121148.html#audience> 2. Программирование на языке Python. Уровень 1. Базовый курс, 40 ак. часов, стоимость 20200 руб. <https://www.specialist.ru/course/python1-a> 3. Введение в машинное обучение на Python, 16 ак. часов, стоимость 30 600 руб <https://edu.softline.com/vendors/softline/vvedenie-v-mashinnoe-obuchenie-na-python/> |
| 2.7 | Минимальное количество человек на курсе | 30 |
| 2.8 | Максимальное количество человек на курсе | Не ограничено |
| 2.9 | Данные о количестве слушателей, ранее успешно прошедших обучение по образовательной программе | 2 300 |
| 2.10 | Формы аттестации | Входное тестирование, промежуточное тестирование, итоговое тестирование, выходная диагностика |
|  | Указание на область реализации компетенций цифровой экономики, к которой в большей степени относится образовательная программа, в соответсреподававии с Перечнем областей | Программирование и создание ИТ-продуктов |

1. **Аннотация программы**

В соответствии с профессиональным стандартом «Программист», программа повышения квалификации «Основы программирования на Python» направлена на формирование профессиональных компетенций в области разработки и отладки программного кода.

Язык программирования Python позволяет быстро создавать прототипы ИТ-продуктов, выпускать на рынок минимально жизнеспособную версию продукта и получать обратную связь от реальных пользователей.

Для успешного освоения программы обучающимся необходимо владеть базовыми компетенциями цифровой грамотности, в том числе навыками владения ПК на уровне среднего или продвинутого пользователя, базовыми знаниями в области любого языка программирования.

В процессе обучения по программе обучающимся рекомендуется прослушать и конспектировать все видеолекции, выполнять по каждой теме практические задания с автоматизированной проверкой и проработать самостоятельно дополнительные материалы, выполнить тесты для самоконтроля, для успешного завершения обучения, сдать итоговый тест/выходную диагностику (не менее 60% верных ответов).

Обучающиеся в результате освоения программы будут знать конструкции языка, типы и структуры данных, функции, научатся применять объектно-ориентированное и функциональное программирование, узнают про особенности реализации Python, научатся писать асинхронный и многопоточный код.

После успешного окончания курса слушатель сможет использовать полученный опыт для разработки проектов на языке Python различной сложности.

2. ШАБЛОН ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ (ДПО)

**Образовательная автономная некоммерческая организация дополнительного профессионального образования "Высшая школа информационных технологий и безопасности" (ОАНО ДПО «Выштех»)**

ИНН: 7703434727 ОГРН: 1177700018210

127204, г. Москва, Долгопрудненское шоссе, д. 3, этаж 2, пом. VII, комн.22Г

УТВЕРЖДАЮ:

Исполнительный директор

ОАНО ДПО «ВЫШТЕХ»

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.И. Поддубный-Тумаларян

«19» октября 2020 г.

**Дополнительная профессиональная программа повышения квалификации**

**«Основы программирования на Python»**

72 часа

**Москва, 2020**

**ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ**

**1.Цель программы**

Программа повышения квалификации "Основы программирования на Python" направлена на освоение профессиональных компетенций в области разработки и отладки программного кода на языке Python по направлению развития цифровой экономики "Программирование и создание ИТ-продуктов".

Программа повышения квалификации "Основы программирования на Python" также формирует профессиональные компетенции для решения прикладных задач с помощью языка Python.

**2.Планируемые результаты обучения:**

2.1.Знание (осведомленность в областях)

2.1.1. особенностей языка программирования Python и областей его применения;

2.1.2. синтаксиса основных конструкций языка программирования Python (управление потокам выполнения программы, объявление переменных, функций, классов);

2.1.3. основных типов данных, используемых в стандартной библиотеке Python (числа, строки, списки, кортежи, словари, множества, функции, классы);

2.1.4. основных возможностей, синтаксиса и области применения модулей стандартной библиотеки Python (os, sys, json, csv, tempfile, unittest, decimal);

2.1.5. особенности функционального и объектно-ориентированного подходов в программировании, их различия и области применения;

2.1.6. целей и задач тестирования программ, методики построения наборов тестов и способы их реализации в коде с помощью фреймворка unittest;

2.1.7. основных возможностей и синтаксиса библиотеки requests по работе с API

2.2. Умение (способность к деятельности)

2.2.1. устанавливать, настраивать среду разработки Python и работать в ней;

2.2.2. выбирать архитектурный стиль приложения исходя из технического задания по созданию программного продукта;

2.2.3.производить декомпозицию задачи и разрабатывать алгоритм простых приложений;

2.2.4. создавать несложные скрипты и консольные приложения, автоматизирующие повседневные задачи;

2.2.5. использовать при разработке программного приложения библиотеки requests;

2.2.6. работать с данными, представленными в форматах json и csv (получать, обрабатывать, сохранять);

2.2.7. проводить отладку приложения с помощью модуля pdb;

2.2.8. с помощью фреймворка unittest создавать наборы тестов для приложения, написанного на языке Python;

2.2.9. использовать при разработке программного приложения модули стандартной библиотеки Python;

2.2.10. делать запросы к API интернет сервисов и обрабатывать полученные данные;

2.2.11. работать с документацией Python;

2.2.12. читать и анализировать код написанный на языке Python.

2.3.Навыки (использование конкретных инструментов)

2.3.1 работы в командной строке;

2.3.2. работы в интерпретаторе Python;

2.3.3. работы в среде разработки Python;

2.3.4. работы с официальной документаций по языку програмирования Python;

2.3.5. чтения кода, написанном на языке программирования Python;

2.3.6. создания консольных приложений и модулей на языке программирования Python;

2.3.7. создания тестов для python-приложений с использованием фреймворка unittest;

2.3.8. отладки кода с помощью модуля pdb.

**3.Категория слушателей** (возможно заполнение не всех полей)

3.1. Образование: среднее профессиональное образование / высшее образование

3.2. Квалификация: не ниже 3

3.3. Наличие опыта профессиональной деятельности: специальные требования отсутствуют

3.4. Предварительное освоение иных дисциплин/курсов /модулей

для успешного прохождения программы слушатели должны обладать: навыками владения ПК на уровне среднего или продвинутого пользователя; первичными знаниями и умениями в области алгоритмизации и программирования (в рамках школьной программы); первичными знаниями об операционных системах Unix и работе в командной строке; навыками работы с текстами на английском языке (перевод текста со словарем)

**4.Учебный план программы «Основы программирования на Python»**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль** | **Всего, час** | **Виды учебных занятий** | | |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Введение в Python | 10 | 2 | 6 | 2 |
| 2 | Организация кода и окружение | 7 | 1 | 5 | 1 |
| 3 | Коллекции | 5 | 1 | 3 | 1 |
| 4 | Функции | 9 | 2 | 5 | 2 |
| 5 | Классы и объекты | 6 | 1 | 4 | 1 |
| 6 | Наследование | 5 | 2 | 2 | 1 |
| 7 | Работа с ошибками | 9 | 1 | 7 | 1 |
| 8 | Особые методы классов | 7 | 1 | 5 | 1 |
| 9 | Структурное программирование | 4 | 1 | 2 | 1 |
| 10 | Отладка и тестирование | 9 | 2 | 6 | 1 |
| **Итоговая аттестация** | | 1 | Тестирование | | |
| **Всего по программе, час** | | 72 |  | | |

**5.Календарный план-график реализации образовательной программы**

Дата начала обучения: 01.11.2020  
Дата завершения обучения: 15.11.2020

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Наименование учебных модулей** | **Трудоёмкость (час)** | **Сроки обучения** |
| 1 | Введение в Python | 10 | 01.11.2020- 02.11.2020 |
| 2 | Организация кода и окружение | 7 | 03.11.2020 |
| 3 | Коллекции | 5 | 04.11.2020 |
| 4 | Функции | 9 | 05.11.2020- 06.11.2020 |
| 5 | Классы и объекты | 6 | 07.11.2020 |
| 6 | Наследование | 5 | 08.11.2020 |
| 7 | Работа с ошибками | 9 | 09.11.2020- 10.11.2020 |
| 8 | Особые методы классов | 7 | 11.11.2020- 12.11.2020 |
| 9 | Структурное программирование | 4 | 13.11.2020 |
| 10 | Отладка и тестирование | 9 | 14.11.2020 - 15.11.2020 |
|  | Итоговая аттестация | 1 | 15.11.2020 |
| **Всего:** | | **72** | **15 календарных дней** |

**6.Учебно-тематический план программы «Основы программирования на Python»**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ п/п** | **Модуль / Тема** | **Всего, час** | **Виды учебных занятий** | | | **Формы контроля** |
| **лекции** | **практические занятия** | **самостоятельная работа** |
| 1 | Введение в Python | 10 | 2 | 6 | 2 | Тест “Проверка установки Python”  Тест на типы и конструкции |
| 1.1 | Установка Python 3 | 1,5 | 0,4 | 1 | 0,1 |
| 1.2 | Выбор среды разработки | 0,9 | 0,3 | 0,5 | 0,1 |
| 1.3 | Работа в терминале | 1,1 | 0,2 | 0,7 | 0,2 |
| 1.4 | Базовые типы: числовые типы | 1,3 | 0,3 | 0,5 | 0,5 |
| 1.5 | Базовые типы: логический тип | 1,3 | 0,2 | 1 | 0,1 |
| 1.6 | Базовые типы: строки и байтовые строки | 1,6 | 0,1 | 1 | 0,5 |
| 1.7 | Базовые типы: объект None | 1 | 0,2 | 0,6 | 0,2 |
| 1.8 | Конструкции управления потоком | 1,3 | 0,3 | 0,7 | 0,3 |
| 2 | Организация кода и окружение | 7 | 1 | 5 | 1 | Тест по модулям 1, 2, 3, задания на программирование: “Сумма цифр в строке”, “Рисуем лестницу”, “Корни квадратного уравнения” |
| 2.1 | Модули и пакеты | 1,8 | 0,4 | 1 | 0,4 |
| 2.2 | Виртуальное окружение (Virtualenv) | 2,5 | 0,2 | 2 | 0,3 |
| 2.3 | Объектная структура в Python | 2,7 | 0,4 | 2 | 0,3 |
| 3 | Коллекции | 5 | 1 | 3 | 1 | Тест по коллекциям |
| 3.1 | Списки и кортежи | 2 | 0,5 | 1 | 0,5 |
| 3.2 | Словари | 1,2 |  | 1 | 0,2 |
| 3.3 | Множества | 1,8 | 0,5 | 1 | 0,3 |
| 4 | Функции | 9 | 2 | 5 | 2 | Тест по функциям, Тест модулям 4,5, задания на программирование: “Key-value хранилище”, “Декоратор to\_json” |
| 4.1 | Функции | 1,5 | 0,2 | 1 | 0,3 |
| 4.2 | Файлы | 2,1 | 0,6 | 1 | 0,5 |
| 4.3 | Функциональное программирование | 1,6 | 0,4 | 1 | 0,2 |
| 4.4 | Декораторы | 1,8 | 0,3 | 1 | 0,5 |
| 4.5 | Генераторы | 2 | 0,5 | 1 | 0,5 |
| 5 | Классы и объекты | 6 | 1 | 4 | 1 | Тест “классы и объекты” Тест “классы и объекты” |
| 5.1 | Классы и экземпляры | 3,3 | 0,7 | 2 | 0,6 |
| 5.2 | Методы | 2,7 | 0,3 | 2 | 0,4 |
| 6 | Наследование | 5 | 2 | 2 | 1 | Тест “Наследование” |
| 6.1 | Наследование в Python | 2,5 | 1 | 1 | 0,5 |
| 6.2 | Композиция классов, пример | 2,5 | 1 | 1 | 0,5 |
| 7 | Работа с ошибками | 9 | 1 | 7 | 1 | Тест модулям 6,7,8 , задания на программирование: “Реализация простого класса для чтения из файла”, “Классы и наследование” |
| 7.1 | Классы исключений и их обработка | 4 | 0,6 | 3 | 0,4 |
| 7.2 | Генерация исключений | 5 | 0,4 | 4 | 0,6 |
| 8 | Особые методы классов | 7 | 1 | 5 | 1 | Тест “Особые методы классов”, Тест по модулю 9, задание на программирование ”Файл с магическими методами” |
| 8.1 | Магические методы | 2,5 | 1 | 1 | 0,5 |
| 8.2 | Итераторы | 2,3 |  | 2 | 0,3 |
| 8.3 | Контекстные менеджеры | 2,2 |  | 2 | 0,2 |
| 9 | Структурное программирование | 4 | 1 | 2 | 1 | Тест на знание PEP 8, Тест “Поиск ошибок в структуре программы” |
| 9.1 | Культура программирования | 4 | 1 | 2 | 1 |
| 10 | Отладка и тестирование | 9 | 2 | 6 | 1 | Тест на модульное тестирование,  Тест Зачем тестировать программы, задание на программирование “Задача по созданию модульного теста функции factorize” |
| 10.1 | Зачем нужно тестировать программы | 0,9 | 0,4 |  | 0,5 |
| 10.2 | Модульное тестирование и Test-Driven Development | 1 | 0,5 |  | 0,5 |
| 10.3 | Тема 11.3.Библиотека doctest | 2,2 | 0,2 | 2 |  |
| 10.4 | Библиотека unittest | 3,6 | 0,6 | 3 |  |
| 10.5 | Отладка | 1,3 | 0,3 | 1 |  |
|  | Итоговая аттестация | 1 |  |  |  | Тест |

**7. Учебная (рабочая) программа повышения квалификации «Основы программирования на Python»**

**Модуль 1.**  Введение в Python (10 часов)

Тема 1.1. Установка Python 3 (1,5 часа)

* история создания языка программирования Python
* установка Python 3.6

Тема 1.2. Выбор среды разработки (IDE) (0,9 час)

* выбор среды разработки (IDE)

Тема 1.3. Работа в терминале (1,1 час)

* основы работы в терминале
* интерпретатор Python и работа в нем

Тема 1.4 Базовые типы: числовые типы (1,3 час)

* представление целых и вещественных чисел в вычислительных системах
* объявление переменных
* операции с числами
* синтаксис: классы int, float, функция type
* автоматическое приведение типов
* базовые понятия о об изменяемых типах и объектной модели python

Тема 1.5 Базовые типы: логический тип (1,3 часа)

* булевый тип данных, bool, True, False
* логические операторы и их приоритет
* особенность приведения типов к булеву типу
* повторное использование кода, базовое представление о системе импорта в Python

Тема 1.6 Базовые типы: строки и байтовые строки (1,6 часа)

* строки и байтовые строки, классы str, byte
* пользовательский ввод-вывод, функции input, print
* виды кавычек и способы объявления строк
* операции над строками, срезы, конкатенация, итерация по строке
* операции кодирования строк, методы: encode, decode

Тема 1.7 Базовые типы: объект None (1 час)

* тип данных None, особенности сравнения None

Тема 1.8 Конструкции управления потоком (1,3 часа)

* конструкции управления потоком
* условный оператор if, синтаксис конструкции if-elif-else
* операторы цикла: while, for и особенности их применения в Python
* операторы break, continue

**Модуль 2.** Организация кода и окружение (7 часов)

Тема 2.1. Модули и пакеты (1,8 часа)

* система импорта в Python
* оператор import, способы импортирования объектов из стандартной библиотеки
* объединение модулей в пакеты, файл \_\_init\_\_.py
* модули стандартной библиотеки Python: sys, os, inspect
* синтаксис: конструкция “if \_\_name\_\_ == ‘\_\_main\_\_’:”

Тема 2.2. Виртуальное окружение (Virtualenv) (2,5 часа)

* проблема конфликта версий модулей Python при разработке приложений, понятие виртуального окружения
* создание виртуального окружения, модуль стандартной библиотеки venv
* менеджер пакетов pip, установка и удаление пакетов из репозитория PyPi

Тема 2.3. Объектная структура в Python (2,7 часа)

* ссылки на объект, счетчик ссылок, сборщик мусора
* синтаксис: функция dir, оператор сравнения is
* понятие изменяемых и неизменяемых типов данных

**Модуль 3.** Коллекции (5 часов)

Тема 3.1. Списки и кортежи (2 часа)

* списки, класс list, элементы списка, обращение по индексу, срезы
* итерация по элементам списка, функция enumerate
* методы строк: append, extend, index, reverse, sort
* синтаксис: функции sum, min, max, sorted, reversed, hash, оператор del,
* кортежи и их отличия от списков, класс tuple

Тема 3.2. Словари (1,2 часа)

* словари, класс dict, способы обращения к элементам словаря
* методы словарей: update, pop, setdefault, keys, values, items
* модуль collections, тип данных orderedDict

Тема 3.3.Множества (1,8 часа)

* множества, класс set, способы создания множеств
* методы множества: set, методы множеств remove, pop
* операции над множествами, операторы &, |, ^, -
* тип данных frozenset

**Модуль 4.** Функции (9 часов)

Тема 4.1. Функции (1,5 часа)

* функции, как объекты, атрибуты функции: \_\_doc\_\_, \_\_name\_\_, \_\_defaults\_\_
* аргументы и параметры функции (позиционные, именованные), значение параметра по умолчанию
* распаковка аргументов \*args, \*\*kwargs
* возврат значений из функции, return
* области видимости, функции: global, local
* модуль стандартной библиотеки datetime

Тема 4.2. Файлы (2,1 часа)

* работа с файлами из python кода
* функция open, режимы открытия файла
* файловый объект, методы read. readline, write, close, tell
* использования контекстного менеджера для работы с файловым объектом, оператор with

Тема 4.3. Функциональное программирование (1,6 часа)

* функции - объекты первого класса
* анонимные функции, lambda
* замыкания
* синтаксис: функции map, filter, zip
* модуль стандартной библиотеки functools, функции: reduce, partial
* списочные вложения (List Comprehension)

Тема 4.4. Декораторы (1,8 часа)

* понятие декоратора и круг проблем, которые он решает
* синтаксис декораторов
* декораторы с параметрами
* правила применения нескольких декораторов

Тема 4.5. Генераторы (2 часа)

* определение генератора, оператор yield
* синтаксис: функция next
* передача значений в генератор, метод send

**Модуль 5.** Классы и объекты (6 часов)

Тема 5.1. Классы и экземпляры (3,3 часа)

* введение в ооп, понятие класса и экземпляра класса, функция isinstance
* способы объявления класса, понятие атрибутов
* обращение к атрибутам, различие между атрибутами класса и атрибутами экземпляра класса
* этапы создания объекта, магические методы \_\_new\_\_, \_\_init\_\_, \_\_repr\_\_, \_\_str\_\_
* хранение атрибутов, метод \_\_dict\_\_, функция dir

Тема 5.2.Методы (2,7 часа)

* понятие методов, методы класса и экземпляра класса
* вызов методов внутри класса через ссылку на экземпляр - self
* методы класса, декоратор @classmethod
* “приватность” методов, соглашение об именовании методов
* статические методы, декоратор @staticmethod
* доступ к атрибутам, свойства, декоратор @property
* модуль requests

**Модуль 6.** Наследование (5 часов)

Тема 6.1. Наследование в Python (2,5 часа)

* принцип ооп - наследование, описание концепции, базовые классы и классы наследники
* множественное наследование, функции: super, issubclass
* проблемы множественного наследования, “ромб смерти”, линеаризация, метод \_\_mro\_\_
* особенности наследования атрибутов и методов
* переопределение методов в классах наследниках
* классы примеси, mixins

Тема 6.2. Композиция классов, пример (2,5 часа)

* понятие композиции классов
* выбор - композиция классов или наследование, области применения

**Модуль 7.** Работа с ошибками(9 часов)

Тема 7.1.Классы исключений и их обработка (4 часа)

* понятие исключения, получение информации из сообщении об ошибке, traceback
* типы исключений, иерархия встроенных исключений стандартной библиотеки Python
* обработка исключений, синтаксическая конструкция try-except
* обработка нескольких исключений, блоки finally, else

Тема 7.2. Генерация исключений (5 часов)

* выброс исключения, функция raise
* модуль traceback
* проброс исключений, конструкция “raise .. from err”
* создание пользовательских исключений

**Модуль 8.** Особые методы классов(7 часов)

Тема 8.1. Магические методы (2,5 часа)

* внутреннее устройство объектов Python
* “магические” методы \_\_setattr\_\_, \_\_setattribute\_\_, \_\_getattr\_\_, \_\_getattribute\_\_, \_\_call\_\_
* пример реализации декоратора с помощью класса

Тема 8.2. Итераторы (2,3 часа)

* протокол итератора
* итератор и итерируемый объект, функция iter
* реализация протокола итератора в классе, “магические” методы \_\_iter\_\_, \_\_next\_\_

Тема 8.3. Контекстные менеджеры (2,2 часа)

* понятие контекстного менеджера, оператор with
* пример реализации контекстного менеджера в классе, “магические” методы \_\_enter\_\_, \_\_exit\_\_

**Модуль 9.** Структурное программирование(4 часа)

Тема 9.1. Культура программирования(4 часа)

* почему важна читабельность кода, Дзен Python
* code style, соглашения принятые в сообществе Python, PEP8
* инструменты разработчика - линтеры кода
* понятие структурного программирования
* пример проектирования приложения «сверху-вниз»

**Модуль 10.** Отладка и тестирование (9 часов)

Тема 10.1. Зачем нужно тестировать программы (0,9 часа)

* необходимость обеспечения качества программного продукта
* виды тестирования (модульное, интеграционное, приемочное)

Тема 10.2.Модульное тестирование и Test-Driven Development (1 часа)

* техника разработки программного обеспечения Test-Driven Development
* цикл разработки TDD
* плюсы и минус Test-Driven Development
* пример разработки через тестирование

Тема 10.3.Библиотека doctest (2,2 часа)

* основы работы с модулем стандартной библиотеки doctest

Тема 10.4.Библиотека unittest (3,6 часа)

* история создания модуль unittest, “наследие” от JUnit
* структура модульного теста, класс TestCase
* утверждения assert в модуле unittest
* фикстуры, методы setUp, tearDown, setUpClass, tearDownClass
* параметризация тестовых случаев, subTest
* проверка выброса исключений assertRaises

Тема 10.5.Отладка (1,3 часа)

* модуль pdb, основы использования отладчика pdb

**Итоговая аттестация.** Выходное тестирование (1 час)

**Описание практико-ориентированных заданий и кейсов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Номер темы/модуля** | **Наименование практического занятия** | **Описание** |
| **1** | Модуль 1 Тема 1.1 | Выбор среды разработки (IDE) | Практика по учебному материалу |
| **2** | Модуль 1  Тема 1.2 | Работа в терминале | Практика по учебному материалу |
| **3** | Модуль 1  Тема 1.3 | Проверка установки Python | В материалах темы “Установка Python 3” изучить инструкции по установке Python для своей операционной системы. Произвести установку Python версии 3.6. Запустить интерпретатор Python из командной строки. Убедиться, что: 1. интерпретатор запускается 2. в приглашении интерпретатора указана версия 3.6. |
| **4** | Модуль 1 Тема 1.4 | Базовые типы: числовые типы; Работа с документацией - type, float, int, Numeric Types, decimal, fractions,”Floating Point Arithmetic: Issues and Limitations” | Практика по учебному материалу; Работа с документацией |
| **5** | Модуль 1 Тема 1.5 | Базовые типы: логический тип; Работа с документацией - Класс bool, Built-in Constants (True, False),  Boolean Operations, Comparisons,  Operator precedence, Modules | Практика по учебному материалу; Работа с документацией |
| **6** | Модуль 1 Тема 1.6 | Базовые типы: строки и байтовые строки; Работа с документацией - id, input, print, Text Sequence Type, str, “The while statement” | Практика по учебному материалу; Работа с документацией |
| **7** | Модуль 1 Тема 1.7 | Базовые типы: объект None;Работа с документацией - None | Практика по учебному материалу; Работа с документацией |
| **8** | Модуль 1 Тема 1.8 | Конструкции управления потоком;Работа с документацией - “The if statement”, “More Control Flow Tools”, random | Практика по учебному материалу; Работа с документацией |
| **9** | Модуль 2 Тема 2.1 | Модули и пакеты; Работа с документацией - sys, os, inspect, “The import system”, “How do I create a .pyc file?” | Практика по учебному материалу, Работа с документацией |
| **10** | Модуль 2 Тема 2.2 | Виртуальное окружение (Virtualenv); Работа с документацией - “venv - Creation of virtual environments”, “Virtual Environments and Packages”, “pip - The Python Package Installer” | Практика по учебному материалу, Работа с документацией |
| **11** | Модуль 2 Тема 2.3 | Объектная структура в Python;Работа с документацией - dir, “Identity comparisons”, dis | Практика по учебному материалу, работа с документацией |
| **12** | Модуль 1,2 | Сумма цифр в строке,Рисуем лестницу,Корни квадратного уравнения | Задания на программирование |
| **13** | Модуль 3 Тема 3.1 | Списки и кортежи,Работа с документацией - lists,  “Data Structures”, ranges, sum, hash, len, max, min, reversed, sorted | Практика по учебному материалу, работа с документацией |
| **14** | Модуль 3 Тема 3.2 | Словари, Работа с документацией - “Mapping Types — dict”, “OrderedDict objects”, dict | Практика по учебному материалу, работа с документацией |
| **15** | Модуль 3 Тема 3.3 | Множества,Работа с документацией - set,  “Set Types - set, frozenset” | Практика по учебному материалу, работа с документацией |
| **16** | Модуль 4 Тема 4.1 | Работа с документацией - “Function definitions, “What is the difference between arguments and parameters?”, “How do I write a function with output parameters (call by reference)?, “How do you make a higher order function in Python?”, datetime, globals, locals, “How can I pass optional or keyword parameters from one function to another?”, “Callable types” | Работа с документацией |
| **17** | Модуль 4 Тема 4.2 | Файлы, Работа с документацией - open,  file object, ”Reading and Writing Files”, ”Methods of File Objects” | Практика по учебному материалу, работа с документацией |
| **18** | Модуль 4 Тема 4.3 | Функциональное программирование, Работа с документацией - “Functional Programming HOWTO”, map, filter, functools, functools.partial, functools.reduce, “List Comprehensions”, zip, “Why can’t lambda expressions contain statements?” | Практика по учебному материалу, работа с документацией |
| **19** | Модуль 4 Тема 4.4 | Декораторы, Работа с документацией - decorator, @functools.wraps | Практика по учебному материалу, работа с документацией |
| **20** | Модуль 4 Тема 4.5 | Генераторы,Работа с документацией - “Generators”, generators, “Generator Expressions”, next,“The yield statement”, json, “Saving structured data with json” | Практика по учебному материалу, работа с документацией |
| **21** | Модуль 3,4 | Key-value хранилище,  Декоратор to\_json | Задание на программирование |
| **22** | Модуль 5 Тема 5.1 | Классы и экземпляры, Работа с документацией - “Classes”, “A First Look at Classes”, isinstance, “Private Variables”, “Why must ‘self’ be used explicitly in method definitions and calls?”, Objects | Практика по учебному материалу, работа с документацией |
| **23** | Модуль 5 Тема 5.2 | Методы, Работа с документацией - “Special method names”, @staticmethod, “Static Methods and Class Methods”, “Properties”,  class, property | Практика по учебному материалу, работа с документацией |
| **24** | Модуль 6 Тема 6.1 | Наследование в Python,Работа с документацией - inheritance, “Multiple Inheritance”,  super, issubclass | Практика по учебному материалу, работа с документацией |
| **25** | Модуль 6 Тема 6.2 | Композиция классов, пример | Практика по учебному материалу |
| **26** | Модуль 7 Тема 7.1 | Классы исключений и их обработка,Работа с документацией - “Errors and Exceptions”, “Built-in Exceptions” | Практика по учебному материалу, работа с документацией |
| **27** | Модуль 7 Тема 7.2 | Генерация исключений, Работа с документацией - timeit, “Raising Exceptions”, “User-defined Exceptions”, traceback, The assert statement” | Практика по учебному материалу, работа с документацией |
| **28** | Модуль 5,6,7 | “Реализация простого класса для чтения из файла” | Задание на программирование |
| **29** | Модуль 8 Тема 8.1 | Магические методы,Работа с документацией - “Special method names” | Практика по учебному материалу, работа с документацией |
| **30** | Модуль 8 Тема 8.2 | Итераторы, Работа с документацией - “Iterator Types”, next, “Iterators”, “Data Types That Support Iterators” | Практика по учебному материалу, работа с документацией |
| **31** | Модуль 8 Тема 8.3 | Контекстные менеджеры, Работа с документацией - “Context Manager Types”, “The with statement”, “With Statement Context Managers” | Практика по учебному материалу, работа с документацией |
| **32** | Модуль 8 | Файл с магическими методами | Задание на программирование |
| **33** | Модуль 9 Тема 9.1 | Работа с документацией - “The Zen of Python”, “PEP 8” | Работа с документацией |
| **34** | Модуль 10 Тема 10.3 | Практика по учебному материалу, работа с документацией | Практика по учебному материалу, работа с документацией |
| **35** | Модуль 10 Тема 10.4 | Библиотека unittest,Работа с документацией - unittest, asserts, assertRaises, “Distinguishing test iterations using subtests”,  Задача по созданию модульного теста функции factorize | Практика по учебному материалу, работа с документацией, задание на программирование |
| **36** | Модуль 10 Тема 10.5 | Отладка,Работа с документацией - “pdb - The Python Debugger” | Практика по учебному материалу, работа с документацией |

**8.Оценочные материалы по образовательной программе**

**8.1. Вопросы тестирования по модулям**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ модуля** | **Вопросы входного тестирования** | **Вопросы промежуточного тестирования** | **Вопросы итогового тестирования** |
| **Входное тестирование** | Тест содержит 20 вопросов, выявляющих до начала обучения соответствие уровня знаний обучаемого,предъявляемым требованиям (п 3.4) |  |  |
| **1** |  | Тест №1 Модуля 1 содержит 2 вопроса:  один - на определение версии Python, установленного на компьютере студента, второй - на проверку правильности выполнения команды в интерпретаторе Python.  Тест №2 Модуля 1 содержит 16 вопросов:  - Вопросы 1, 2, 10 на проверку особенностей приведения типов), выбирается один правильный вариант ответа из предложенных  *-* Вопросы 5, 6 о представлении строк, выбирается несколько правильных вариантов ответа из предложенных  Вопросы 3, 14, 15, 16 об управляющих конструкциях и использовании логических операторов, выбирается один правильный вариант ответа из предложенных  - Вопросы 11, 12, 13 про байтовые строки, выбирается один правильный вариант ответа из предложенных  - Вопросы 8, 9 про срезы, выбирается один правильный вариант ответа из предложенных  - Вопросы 7, 4 про работу со стоками, выбирается один правильный вариант ответа из предложенных |  |
|
| **2** |  | Тест содержит 15 вопросов:  - В вопросах 4,5, выбирается несколько правильных вариантов ответа из предложенных, в остальных вопросах выбирается один правильный вариант ответа из предложенных.  - Вопрос 1 по назначению языка программирования Python  - Вопросы 2, 3, 15 (по особенностям реализации интерпретатора Python)  - Вопросы 4, 5 по базовому синтаксису языка Python  - Вопросы 6, 7, 9, 10 по базовым типам данных: int, float, str, bool  - Вопросы 11, 12, 14 по системе импорта, модулям и пакетам  - Вопрос 13 (по созданию изолированного окружения (модуль venv)) |  |
| **3** |  | Тест содержит 10 вопросов:  - Вопросы 1, 2 по особенностям создания объектов типа list, tuple, set, dict  - Вопросы 3, 5, 10 об особенностях реализации коллекций, доступе к элементам коллекций, методах коллекций  - Вопросы 4, 6, 8 по отличиям между типами коллекций и областями применения каждого типа  - Вопрос 7 по функциям и методам для сортировки коллекций |  |
| **4** |  | Тест содержит 10 вопросов:  - Вопросы 1, 6 по синтаксису объявления функции  - Вопрос 2 по областям видимости в Python  Вопросы 3, 4, 5 по параметрам функции и передаче аргументов в функцию  - Вопросы 7, 9 по правильным способам реализации параметров со значением по умолчанию  - Вопрос 8 по способам возврата результата работы функции  - Вопрос 10 по особенностям функций в Python |  |
| **5** |  | Тест содержит 6 вопросов:  - Вопросы 1, 2, 5 по концепции наследования в ооп  - Вопрос 3 по особенностям применения классов примесей в Python  - Вопросы 4, 6 по проблемам множественного наследования |  |
| **6** |  | Тест содержит 10 вопросов:  - Вопросы 1, 8 по “приватности” методов в Python и соглашении об именовании методов  - Вопросы 2, 9 по типам исключений и иерархии встроенных исключений стандартной библиотеки Python  - Вопрос 3, 4, 5 по переопределение методов в классах наследниках, правильном использовании функции super  Вопросы 6, 7 по синтаксису генерации и обработке исключений  Вопрос 10 по использованию композиции в классах |  |
| **7** |  | Тест содержит 6 вопросов:  - Вопросы 1, 2 по именам “магических методов”, входящих в стандартную библиотеку Python  - Вопросы 3, 4, 5, 6 по функционалу реализуемом с помощью конкретного “магического” метода |  |
| **8** |  | Тест содержит 8 вопросов:  - Вопросы 1, 2 по протоколу итератора и механизме работы итератора в Python  Вопросы 3, 4 по способам создания итераторов в Python  Вопросы 5, 6 по синтаксису реализации декоратора с помощью класса  Вопросы 7, 8 по способу реализации контекстных менеджеров в Python |  |
| **9** |  | Тест содержит 6 вопросов:  - Вопросы 1, 2, 3 по правилам именования переменных, методов  - Вопросы 4, 5 по правилам оформления кода с помощью отступов и пробелов  - Вопрос 6 по правилам оформления длинных строк |  |
| **10** |  | Тест содержит 6 вопросов:  - Вопросы 1, 2 по возможностям фреймворка unittest для создания модульных тестов  - Вопросы 3, 4, 5, 6 - по синтаксическим конструкциям для создания тестового окружения, тестовых наборов, параметризации тестов) |  |
| **Выходное тестирование** |  |  | Тест содержит 20 вопросов (по 2 вопроса на каждый модуль) |

**8.2.**  **описание показателей и критериев оценивания, шкалы оценивания**

Входная/выходная диагностика участников представляет собой измерение соответствующих программе компетенций участников перед началом и по завершении обучения:

1. Входная/выходная диагностика представляет собой тест, который направлен на выявление начального уровня подготовки и уровня подготовки после завершения программы

2. Критерии и система оценки результатов входной/выходной диагностики:

**Шкала оценивания для входной диагностики**

|  |  |  |
| --- | --- | --- |
| Критерии | Шкала | Описание |
| зачтено | Обучающийся успешно выполнил входной тест: 50%-100% верных ответов на вопросы теста | Потенциальный обучающийся успешно выполнил входной тест и может быть допущен к обучению по программе |
| не зачтено | Обучающийся не выполнил успешно входной тест: 0%-49% верных ответов на вопросы теста | Потенциальный обучающийся не выполнил успешно входной тест и не может быть допущен к обучению по программе |

**Шкала промежуточного тестирования по темам/модулям**

**Шкала оценивания для промежуточного контроля по темам/модулям**

|  |  |  |
| --- | --- | --- |
| Критерии | Шкала | Описание |
| зачтено | Обучающийся проходит тест для самоконтроля до успешного завершения теста: 55%-100% верных ответов на вопросы теста | Данный вид тестов не оценивается |

**Шкала оценивания для выходной диагностики/итогового тестирования**

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии | Шкала | Описание | Уровень сформированности компетенции |
| зачтено | 91%-100% верных ответов на вопросы итогового теста  Успешно выполнено не менее 80% практических заданий курса | Обучающийся успешно выполнил итоговый тест | Профессиональный уровень |
| зачтено | 76%-90% верных ответов на вопросы итогового теста | Обучающийся успешно выполнил итоговый тест | Продвинутый уровень |
| зачтено | 60%-75% верных ответов на вопросы итогового теста | Обучающийся успешно выполнил итоговый тест | Базовый уровень |
| не зачтено | 50%-59% верных ответов на вопросы итогового теста | Обучающийся не выполнил успешно итоговый тест необходимо повторное назначение теста и консультация преподавателя | Начальный уровень |
| не зачтено | 0%-49% верных ответов на вопросы теста | Обучающийся не выполнил успешно итоговый тест | Не владеет, компетенция не сформирована |

**Шкала оценивания практических заданий**

|  |  |  |  |
| --- | --- | --- | --- |
| Критерии | Шкала | Описание | Допуск к итоговой аттестации/выходной диагностики |
| зачтено | 50%-100% верных ответов по всем практическим заданиям программы | Обучающийся успешно выполнил практические задания | Да |
| не зачтено | 0%-49% верных ответов по всем практическим заданиям программы | Обучающийся не выполнил успешно практические задания в нужном объеме | Нет  Необходимо повторное выполнение заданий и консультация преподавателя |

**8.3.**  **примеры контрольных заданий по модулям или всей образовательной программе**

**Задание 1.**

Написать консольного клиента для работы с хранилищем данных (представление данных - ключ: список значений, формат хранения - json, данные сохраняются в файле на файловой системе). Клиент должен поддерживать две операции с хранилищем - сохранение данных по ключу и получение данных по ключу. Название операции и данные передаются, как набор, описанных в тз, параметров командной строки. Результат запросов на получение данных должен выводиться на печать в стандартный поток вывода терминала.

**Задание 2.**

Реализовать декоратор функции. После применения данного декоратора к любой функции, возвращаемый результат из нее должен иметь тип данных str, а сами данные сериализованы в формат json. Декорируемая функция должна сохранить начальное имя в атрибуте \_\_name\_\_, после применения к ней декоратора.

**8.4.**  **тесты и обучающие задачи (кейсы), иные практикоориентированные формы заданий**

В программе используются преимущественно задания с автоматизированной проверкой, которые являются обязательными для выполнения по каждой теме (см. п. 7.)

**Примеры вопросов теста к Модулю 1:**

1. Запустите интерпретатор Python, введите и выполните в нем следующую команду:

import this

Скопируйте вывод, полученный в результате выполнения команды (без приглашения интерпретатора) и вставьте его в поле ответа.

2. В интерпретаторе Python выполнили операцию 10 / 2. Объект какого типа получится в результате деления?

Варианты ответа:

* вещественное число - float
* целое число - int

3. Отметьте правдивые факты про строки

Варианты ответа:

* строки это последовательность юникодных символов
* строки в Python не изменяемы
* строки в Python это последовательность чисел от 0 до 255

4. Что выведет программа с if elif else?

x = "Москва"

if "ква" not in x:

print("1")

elif "ва" not in x:

print("2")

else:

print("3")

Варианты ответа:

* 2
* 1
* 3

**Пример вопросов теста к Модулю 2:**

1. На каком языке программирования написана основная реализация спецификации Python?

Варианты ответа:

* Python
* Java
* С

2. Как пишутся комментарии в Python?

Варианты ответа:

* # это комментарий
* /\* это комментарий \*/
* // это комментарий

3. Предположим, есть пакет foo, в котором находится модуль bar.py, внутри bar.py определена функция с именем run. Какая конструкция импорта является правильной?

Варианты ответа:

* import run from foo.bar
* from foo.bar import run
* import foo.bar.run

**Пример вопросов теста к Модулю 3:**

1. К чему приведет обращение к не пустому списку по индексу «-1»?

Варианты ответа:

* вернется первый элемент
* ошибка IndexError
* ошибка KeyError
* вернется последний элемент
* среди предложенных вариантов нет правильного ответа

2. В чем отличие стандартного метода списка sort и встроенное функции sorted?

Варианты ответа:

* отличий нет
* в стандартной библиотеки Python нет функции sorted
* функция sorted сортирует исходный список, а метод sort возвращает новый
* функция sorted возвращает новый список, а метод sort сортирует исходный список

**Пример вопросов теста к Модулю 4:**

1. Как оформляется тело функции в Python?

Варианты ответа:

* операторами BEGIN - END
* фигурными скобками
* отступом
* квадратными скобками

2. В коде программы объявлена функция:

def foo(\*args, \*\*kwargs):

pass

Что произойдет при вызове функции foo? Выберите все правильные утверждения.

Варианты ответов:

* все позиционные аргументы запишутся в кортеж args
* все именованные аргументы запишутся в словарь kwargs
* все именованные аргументы запишутся в кортеж kwargs
* выбросится исключение ValueError

**Пример вопросов теста к Модулю 5:**

В коде определены - базовый класс Pet и класс наследник Dog. Отметьте варианты, которые вернут True

Варианты ответов:

* issubclass(Dog, Pet)
* issubclass(Pet, object)
* issubclass(Pet, Dog)
* issubclass(Dog, object)

**Пример вопросов теста к Модулю 6:**

1. Какие из методов генерации исключения разрешены в Python? Выберите все правильные варианты генерации исключений.

Варианты ответа:

* raise None
* raise ValueError(“erorr”)
* raise “ValueError”
* raise ValueError

2. Закончите фразу - “Если имя метода в классе начинается с символа нижнего подчеркивания, например: “\_some\_method”, то согласно соглашения о приватности методов в Python, … “

Варианты ответа:

* метод становится приватным и доступ к нему запрещен из публичного API
* метод можно использовать только в классах наследниках
* обращаться к методу не рекомендуется, так как он не считается методом публичного API
* такой способ именования является особым случаем и не имеет никакого значения в рамках данного соглашения

**Пример вопросов теста к Модулю 7:**

Какой метод отвечает за обращение к объекту по индексу?

Варианты ответа:

* \_\_get\_\_
* \_\_getattr\_\_
* \_\_getattribute\_\_
* \_\_getindex\_\_
* \_\_getitem\_\_
* \_\_getvalue\_\_

**Пример вопросов теста к Модулю 8:**

Как закончить исполнение итератора при реализации метода \_\_next\_\_?

Варианты ответа:

* вернуть None
* вызвать метод \_\_exit\_\_
* выбросить исключение StopIteration

**Пример вопросов теста к Модулю 9:**

В каком блоке кода имеются нарушения в расстановке пробелов?

Вариант ответа:

* c = (a+b) \* (a-b)
* operators = {"plus": "+", "minus": "-", "mul": "\*", "div": "/"}
* n, m = input().split()
* a = np.zeros((5, 5), dtype = int)

**Пример вопросов теста к Модулю 10:**

В коде определен тестовый класс TestFactorization с шаблоном теста test\_simple\_multipliers в нем. Вычислите ожидаемые значения a и b для случая, когда функция factorize вызывается с параметром x = 77. Напишите проверку для этого тестового случая с помощью утверждения assertEqual из класса TestCase.

Код:

class TestFactorization(unittest.TestCase):

def test\_simple\_multipliers(self):

x = 77

a, b = factorize(x)

# напишите вашу реализацию проверки здесь

Прим. factorize - функция разложения числа на простые множители

**8.5.**  **описание процедуры оценивания результатов обучения**

К итоговой аттестации допускаются слушатели, выполнившие не менее 50% всех практических заданий и сдавшие итоговые тесты по отдельным темам (60% верных ответов по каждому тесту по теме), предусмотренные учебной программой.

В ходе итогового электронного тестирования слушателю предлагается ответить на 20 вопросов. Тестирование считается успешно завершенным, если слушателем даны верные ответы на не менее чем 60% вопросов теста.

**9.Организационно-педагогические условия реализации программы**

**9.1. Кадровое обеспечение программы**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№**  **п/п** | **Фамилия, имя, отчество (при наличии)** | **Место основной работы и должность, ученая степень и ученое звание (при наличии)** | **Ссылки на веб-страницы с портфолио (при наличии)** | **Фото в формате jpeg** | **Отметка о полученном согласии на обработку персональных данных** |
| 1 | Швеенков Дмитрий Владимирович | Mail.Ru Group, руководитель команды разработки |  |  | + |
| 2 | Жебрак Александр Михайлович | CTO в Insilico Medicine |  |  | + |
| 3 | Емелин Александр Юрьевич | Программист в отделе внутренней разработки Mail.Ru Group***.*** |  |  | + |
| 4 | Максимов Егор Сергеевич | Researcher группы DL в лаборатории машинного интеллекта МФТИ |  |  | + |
| 5 | Хирьянов Тимофей Фёдорович | Преподаватель информатики в МФТИ и онлайн-школе Фоксфорд. Эксперт московской предметной комиссии ГИА-11 по информатике и ИКТ. |  |  | + |
| 6 | Герцев Михаил Николаевич | Преподаватель информатики в МФТИ |  |  | + |

**9.2.Учебно-методическое обеспечение и информационное сопровождение**

|  |  |
| --- | --- |
| **Учебно-методические материалы** | |
| **Методы, формы и технологии** | **Методические разработки,**  **материалы курса, учебная литература** |
| Проблемно-развивающие методы  Исследовательский метод  Объяснительно-иллюстративный метод  Метод проблемного изложения  Частично-поисковый метод  Метод контроля и самоконтроля  Самостоятельная работа по освоению теоретического материала  Самоконтроль через ответы на вопросы по видеолекциям и выполнение практических заданий для самоконтроля  Промежуточный контроль (тесты по темам для промежуточной аттестации)  Итоговый контроль (итоговое тестирование по курсу)  Консультативная поддержка преподавателя и менторов в режиме онлайн | 1. Саммерфилд М. "Программирование на Python 3. Подробное руководство": Книга содержит все необходимое для практического освоения языка: написания любых программ с использованием, как стандартной библиотеки, так и сторонних библиотек для языка Python 3, а также создания собственных библиотечных модулей./ изд. Символ-Плюс, 2016, 608 стр.  2. Лутц М. "Изучаем Python. 4-е изд.": учебник, написанный доступным языком, рассчитанный на индивидуальную скорость обучения и основанный на материалах учебных курсов, которые автор, Марк Лутц, ведет уже на протяжении десяти лет./ изд. Символ-Плюс, 2019, 1280 стр. |

|  |  |
| --- | --- |
| **Информационное сопровождение** | |
| **Электронные**  **образовательные ресурсы** | **Электронные**  **информационные ресурсы** |
| Авторская система автоматизированной проверки заданий на языке Python на платформе  Материалы программы на образовательной платформе | Официальная документация Python - <https://docs.python.org/3.6/>  Официальная документация библиотеки requests - <https://requests.readthedocs.io/en/master/>  Большая база вопросов и ответов по Python - <https://stackoverflow.com/> |

**9.3.Материально-технические условия реализации программы**

|  |  |
| --- | --- |
| Вид занятий | Наименование оборудования,  программного обеспечения |
| Теоретический компонент | Операционная система семейств Windows, Mac Os, Linux.  Браузер:  • Firefox 64  • Chrome 61  • Safari 11  • Opera 57  Рекомендуется от 128 кбит/сек исходящего потока.  Рекомендуется от 256 кбит/сек входящего потока. |
| Дополнительные теоретические материалы | Операционная система семейств Windows, Mac Os, Linux.  Браузер:  • Firefox 64  • Chrome 61  • Safari 11  • Opera 57  Рекомендуется от 128 кбит/сек исходящего потока.  Рекомендуется от 256 кбит/сек входящего потока. |
| Аттестационный компонент | Операционная система семейств Windows, Mac Os, Linux.  Браузер:  • Firefox 64  • Chrome 61  • Safari 11  • Opera 57  Рекомендуется от 128 кбит/сек исходящего потока.  Рекомендуется от 256 кбит/сек входящего потока. |
| Практический компонент | Операционная система семейств Windows, Mac Os, Linux.  Браузер:  • Firefox 64  • Chrome 61  • Safari 11  • Opera 57  Рекомендуется от 128 кбит/сек исходящего потока.  Рекомендуется от 256 кбит/сек входящего потока. |

**III.Паспорт компетенций (Приложение 2)**

Образовательная автономная некоммерческая организация

дополнительного профессионального образования

«ВЫСШАЯ ШКОЛА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И БЕЗОПАСНОСТИ»

## ПАСПОРТ КОМПЕТЕНЦИИ

**Дополнительная профессиональная программа повышения квалификации**

**«Основы программирования на Python»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1. | Наименование компетенции | | Разработка и отладка программного кода | |
| 2. | Указание типа компетенции | общекультурная/  универсальная |  | |
| общепрофессиональная |  | |
| профессиональная |  | |
| профессионально-специализированная | Да | |
| 3. | Определение, содержание и основные сущностные характеристики компетенции | | Под компетенцией понимается способность постановки и нахождения путей решения прикладных задач по созданию и совершенствованию программного кода с использованием языка программирования Python.  **Знать:**  - особенности языка программирования Python и область его применения  - синтаксис основных конструкций языка программирования Python (управлением потокам выполнения программы, объявление переменных, функций, классов )  - основные типы данных, используемые в стандартной библиотеке Python (числа, строки, списки, кортежи, словари, множества, функции, классы)  - основные возможности, синтаксис и области применения модулей стандартной библиотеки Python (os, sys, json, csv, tempfile, unittest, decimal)  - особенности функционального и объектно-ориентированного подходов в программировании, их различия и области применения  - цели и задачи тестирования программ, методику построения наборов тестов и способы их реализации в коде с помощью фреймворка unittest  - основные возможности и синтаксис библиотеки requests по работе с API  **Уметь:**  - устанавливать, настраивать среду разработки Python и работать в ней  - выбирать архитектурный стиль приложения исходя из технического задания по созданию программного продукта  - производить декомпозицию задачи и разрабатывать алгоритм простых приложений  - создавать несложные скрипты и консольные приложения, автоматизирующие повседневные задачи  - использовать при разработке программного приложения библиотеки requests  - работать с данными, представленными в форматах json и csv (получать, обрабатывать, сохранять)  - проводить отладку приложения с помощью модуля pdb  - с помощью фреймворка unittest создавать наборы тестов для приложения, написанного на языке Python  - использовать при разработке программного приложения модули стандартной библиотеки Python  - делать запросы к API интернет сервисов и обрабатывать полученные данные  - работать с документацией Python  - читать и анализировать код написанный на языке Python  **Владеть:**  - навыком работы в командной строке  - навыком работы в интерпретаторе Python  - навыком работы в среде разработки Python  - навыками работы с официальной документаций по языку програмирования Python  - навыком чтения кода, написанном на языке программирования Python  - навыками создания консольных приложений и модулей на языке программирования Python  - навыками создания тестов для python-приложений с использованием фреймворка unittest  - навыком отладки кода с помощью модуля pdb | |
| 4. | Дескриптор знаний, умений и навыков по уровням | | Уровни  сформированности компетенции  обучающегося | Индикаторы |
|  | | Начальный уровень    (Компетенция недостаточно развита. Частично проявляет навыки, входящие в состав компетенции. Пытается, стремится проявлять нужные навыки, понимает их необходимость, но у него не всегда получается.) | Знает: основные методы модулей стандартной библиотеки Python; синтаксис основных конструкция языка;  Умеет: устанавливать, настраивать среду разработки Python и работать в ней; создавать несложные скрипты и консольные приложения; создавать наборы тестов для приложения;  Владеет: навыком работы в командной строке; навыком работы в среде разработки Python; |
|  | | Базовый уровень    (Уверенно владеет навыками, способен, проявлять соответствующие навыки в ситуациях с элементами неопределённости, сложности.) | Знает: о типах данных из модуля стандартной библиотеки collections и способы их применения при написании кода; методику построения наборов тестов и способы их реализации в коде с помощью фреймворка unittest;  Умеет: писать параметризированные тесты использованием фреймворка unittest; писать код в объектно-ориентированном стиле с применением наследования; писать код с соблюдением требований pep8; делать запросы к API с помощью библиотеки requests;  Владеет: навыком анализа кода программы; навыком работы с документацией к модулям стандартной библиотеки Python; навыком выбора архитектурного стиля программы на основе анализа поставленной задачи; |
|  | | Продвинутый    (Владеет сложными навыками, способен активно влиять на происходящее, проявлять соответствующие навыки в ситуациях повышенной сложности.) | Знает: альтернативные способы написания кода при решении типовых задач;  Умеет: использовать линтеры для проверки стиля кода; пользоваться сторонними утилитами для отладки кода; находить и использовать модули и пакеты с необходимым функционалом на PyPi;  Владеет:  навыком написания собственных модулей; |
|  | | Профессиональный    (Владеет сложными навыками, создает новые решения для сложных проблем со многими взаимодействующими факторами, предлагает новые идеи и процессы, способен активно влиять на происходящее, проявлять соответствующие навыки  в ситуациях повышенной сложности.) | Знает: о способах оценки сложности алгоритмов;  Умеет: определить «слабые» места программы и провести рефакторинг кода; предложить и обосновать новые идеи для решения сложных задач;  Владеет:  навыком принятия решения в условиях неполного описания поставленной задачи; . |
| 5. | Характеристика взаимосвязи данной компетенции с другими компетенциями/ необходимость владения другими компетенциями для формирования данной компетенции | | Для формирования данной компетенции слушатель должен владеть:  - компетенциями цифровой грамотности, в том числе навыками владения ПК на уровне среднего или продвинутого пользователя;  - компетенции в области алгоритмизации, в том числе знаниями и умениями в области алгоритмизации (знание основных алгоритмических структур, умения выражать их на псевдокоде и в виде блок-схем) и программирования, базовыми знаниями в области языка Python | |
| 6. | Средства и технологии оценки | | Входное, промежуточное, итоговое тестирование | |

**VI.Иная информация о качестве и востребованности образовательной программы (результаты профессионально-общественной аккредитации образовательной программы, включение в системы рейтингования, призовые места по результатам проведения конкурсов образовательных программ и др.) (при наличии)**

**V.Рекомендаций к программе от работодателей: наличие не менее двух писем и/или подтверждения на цифровой платформе Государственной системы предоставления ПЦС от работодателей о рекомендации образовательной программы для реализации в рамках Государственной системы предоставления ПЦС на формирование у трудоспособного населения компетенций цифровой экономики с указанием востребованности результатов освоения программы в сфере деятельности соответствующих компаний и готовности к рассмотрению заявок наиболее успешно освоивших образовательную программу граждан на прохождение стажировки и (или) собеседования на предмет трудоустройства путем проставления отметки в профиле программы**

К паспорту прилагаются рекомендации от работодателей в формате .pdf

**VI.Указание на возможные сценарии профессиональной траектории граждан по итогам освоения образовательной программы (в соответствии с приложением)**

|  |  |
| --- | --- |
| **Цели получения персонального цифрового сертификата** | |
| **текущий статус** | **цель** |
| **Развитие компетенций в текущей сфере занятости** | |
| работающий по найму в организации, на предприятии | сохранение текущего рабочего места |
| работающий по найму в организации, на предприятии | развитие профессиональных качеств |
| работающий по найму в организации, на предприятии | повышение заработной платы |
| работающий по найму в организации, на предприятии | смена работы без изменения сферы профессиональной деятельности |

**VII.Дополнительная информация**

**VIII.Приложенные Скан-копии**

1. Рекомендательные письма

2. Скан утвержденной образовательной программы

3. Скан утвержденного паспорта образовательной программы

4. Паспорт образовательной программы в формате word

**Исполнительный директор**

**ОАНО ДПО «ВЫШТЕХ» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О.И. Поддубный-Тумаларян**

М.П.