

**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение  
высшего профессионального образования  
«Национальный исследовательский университет  
„Высшая школа экономики“»**

**Факультет филологии**

**Программа дисциплины**

**«Программирование (язык Python)»**

для направления 035800.62 «Фундаментальная и прикладная лингвистика»  
подготовки бакалавра

**2 курс**

Автор программы: Т. А. Архангельский, к. ф. н. (tarkhangelskiy@hse.ru)

Рекомендована секцией УМС \_\_\_\_\_ «\_\_\_» \_\_\_\_ 201\_ г.  
Председатель \_\_\_\_\_

Утверждена УС факультета филологии «\_\_\_» \_\_\_\_ 201\_ г.  
Ученый секретарь \_\_\_\_\_ [подпись]

**Москва**

*Настоящая программа не может быть использована другими подразделениями  
университета и другими вузами без разрешения кафедры-разработчика программы.*



## 1   Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчётности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления подготовки 035800.62 «Фундаментальная и прикладная лингвистика», обучающихся по программе подготовки бакалавров «Фундаментальная и прикладная лингвистика», изучающих дисциплину «Программирование (язык Python)».

Программа разработана в соответствии с:

- Образовательным стандартом государственного образовательного бюджетного учреждения высшего профессионального образования Высшей школы экономики, в отношении которого установлена категория «национальный исследовательский университет» (ГОБУ ВПО НИУ—ВШЭ), протокол от 02.07.2010
- Образовательной программой направления «Фундаментальная и прикладная лингвистика» подготовки бакалавра;
- Рабочим учебным планом НФ НИУ—ВШЭ на 2014/2015 гг. по направлению подготовки «Фундаментальная и прикладная лингвистика», утверждённым в 2014 году.

## 2   Цели освоения дисциплины

Цели курса:

1. познакомить слушателей, знакомых с основами программирования на языке Python, с более сложными конструкциями функциями и модулями этого языка;
2. научить слушателей создавать веб-приложения с помощью веб-фреймворка для языка Python;
3. научить слушателей реализовывать некоторые стандартные алгоритмы, связанные с обработкой языковых данных, и решать с помощью языка Python возникающие на практике задачи;
4. дать слушателям общее представление о структуре и возможностях других популярных языков и парадигмах программирования, показав их в сравнении с языком Python.

## 3   Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

- освоить ряд конструкций и модулей языка Python;
- научиться создавать веб-приложения с использованием веб-фреймворков для языка Python;
- уметь строить алгоритмы для решения практических задач среднего уровня сложности, связанных с обработкой языковых данных;
- знать ряд стандартных алгоритмов и уметь реализовывать их на языке Python;
- иметь представление о других языках и парадигмах программирования и понимать их возможности и отличия от языка Python;
- знать английские эквиваленты всех используемых в курсе терминов и понятий, уметь пользоваться документацией на английском языке.



## 4 Место дисциплины в структуре образовательной программы

Настоящая дисциплина является факультативом для студентов 2 курса бакалавриата.

При изучении дисциплины используются знания и навыки, полученные в результате освоения дисциплины «Компьютерные инструменты лингвистического исследования» (1 курс) и «Программирование (язык Python)» (1 курс). Часть тем синхронизирована с дисциплиной «Компьютерные инструменты лингвистического исследования» (2 курс).

Прохождение курса является желательным, хотя и не строго обязательным, для студентов, планирующих прослушать курсы «Программирование (язык Python)» (3 курс) и «Теория алгоритмов» (3 курс). Основные положения дисциплины и приобретённые навыки могут быть использованы в дальнейшем при изучении следующих дисциплин: Базы данных, Теория автоматов и формальных языков, Автоматическая обработка естественного языка, Информационный поиск и извлечение данных, Анализ и синтез звучащей речи, Морфологические и синтаксические парсеры, Онтологии и семантические технологии, Машинный перевод, а также при работе над курсовыми и дипломными работами.

## 5 Тематический план учебной дисциплины

№	Название темы	Всего часов по дисциплине	Аудиторные часы		Самостоятельная работа
			Лекции	Сем. и практ. занятия	
1	Углублённое изучение функций и синтаксиса языка Python	36	0	20	16
2	Создание веб-приложений с помощью языка Python	28	0	16	12
3	Специализированные модули, алгоритмы и их реализация	28	0	16	12
4	Знакомство с другими языками и парадигмами программирования и их сравнение с языком Python	16	0	8	8
<b>Итого:</b>		<b>108</b>	<b>0</b>	<b>60</b>	<b>48</b>



## 6     Формы контроля знаний студентов

Тип контроля	Форма контроля	1 курс			
		1	2	3	4
Текущий	Контрольная работа		*		
Итоговый	Экзамен				*

### 6.1    Критерии оценки знаний, навыков

- Домашние задания, если явно не указано иное, необходимо присыдать на корпоративную почту преподавателя до 12:00 дня, предшествующего следующему семинару.
- При оценивании программы в первую очередь обращается внимание на то, насколько её работа соответствует требованиям, описанным в задании. Программа, не запускающаяся из-за синтаксических ошибок, не может получить оценку выше 4 баллов. Баллы могут снижаться, в частности, за неточное выполнение задания и отсутствие разбора случаев, из-за которых при исполнении программы может произойти ошибка. Во вторую очередь могут оцениваться оптимальность решения (в смысле времени работы программы и количества строк кода) и стиль.
- На контрольной работе во 2 модуле проверяется знание синтаксиса и функций языка Python, изученных в 1-2 модулях, и умение пользоваться фреймворком flask для создания веб-приложений.
- На экзамене проверяются все знания и умения, приобретённые во время изучения настоящей дисциплины.
- Основной частью задания контрольной работы и экзамена является задача, состоящая из 2-3 частей разного уровня сложности. Для получения положительной оценки необходимо решить задачу, написав программу на языке Python. Во время контрольных мероприятий разрешается пользоваться любыми источниками информации (если явным образом не оговорено иное).
- При обнаружении плагиата в домашнем или контрольном задании это задание получает оценку 0 баллов.



## 7 Содержание дисциплины

### 1. Обработка данных без использования языков программирования.

1. Функции, возвращающие несколько значений. Аргументы по умолчанию.
2. Множества и операции над ними (добавление, удаление, поиск элементов; сравнение; пересечение, объединение и разность). Различия между множествами и массивами. Различие между типами и токенами в тексте, получение списка типов с помощью множеств. Функция sorted для множеств и массивов.
3. Использование list comprehensions для обработки массивов. Выражения, возвращающие логические значения True/False. Функции max, min, sum, any, all.
4. Разница в передаче аргументов по ссылке и по значению. Копирование значений с помощью deepcopy. Сериализация и загрузка данных в формате json.
5. Итераторы, оператор yield, преимущества использования итераторов. Функция zip, модуль itertools.
6. Функции как объекты первого класса: передача функций в качестве аргументов, использование функций в качестве ключей и значений словарей, примеры.
7. Определение функций внутри других функций. Лямбда-функции, их использование при сортировке.
8. Функции map и reduce, примеры применения.
9. Скорость работы программы и понятие о сложности вычислений. Оптимизация кода. Использование множеств и словарей для поиска; использование итераторов вместо функций, возвращающих массив; предварительная компиляция регулярных выражений; ленивые вычисления.
10. Разбиение большой задачи на модули, способы импорта функций из модулей, переменная \_\_name\_\_. Установка модулей из репозиториев с помощью setuptools и pip.

### 2. Создание веб-приложений с помощью веб-фреймворков для языка Python.

1. Передача данных в Интернете: сервер, клиент, IP-адреса, порты, протокол http, GET- и POST-запросы. Использование пакета urllib для загрузки страниц из Интернета. Защита от скачивания, файл robots.txt и поле user-agent. Альтернативные пути доступа к веб-ресурсам (дампы википедии, API).
2. HTML и CSS, разница между серверными и клиентскими скриптах. Обработка percent encoding и парсинг/порождение URL с GET-запросами в Python. Преобразование HTML-entities в текст. Краулеры (пауки), поиск в ширину и в глубину. «Подводные камни»: динамически порождаемые страницы, циклы.
3. Создание веб-приложений с помощью Python: обзор веб-фреймворков (django, flask). Принципы работы flask. Использование flask-приложения в режиме веб-сервера на локальном компьютере.
4. Принципы использования декораторов в flask, адресация веб-страниц, использование переменных в адресах, перенаправления.
5. HTML-формы, обработка GET- и POST-запросов с помощью flask.
6. Использование шаблонов страниц в flask, основы языка шаблонов Jinja, функция render\_template и её аргументы.
7. Работа с cookies и авторизация пользователей в flask.
8. Работа с MySQL в Python.
9. Удалённая работа с сервером: ssh, sftp. Размещение flask-проекта на сервере в виде отдельного веб-сервера и с помощью веб-сервера apache.



### **3. Специализированные модули и алгоритмы для работы с языковыми данными.**

1. Модуль обработки текстов nltk, обзор функций.
2. Понятие коллокации, различные определения и метрики. Закон Ципфа и другие статистические закономерности в текстах. Построение частотных списков n-грамм и коллокаций, подсчёт статистики с помощью nltk.
3. Корпуса текстов в nltk.
4. Базовые алгоритмы на графах и модуль networkx.

### **4. Обзор популярных языков и парадигм программирования и их сравнение с языком Python.**

## **8      Образовательные технологии**

Для изучения дисциплины необходим компьютер и следующее программное обеспечение: текстовый редактор Notepad++ или любой другой, поддерживающий подсветку синтаксиса, переключение между разными кодировками и поиск с использованием регулярных выражений; интерпретатор языка Python. Домашние задания необходимо присыпать электронной почтой на адрес tarkhangelskiy@hse.ru.

## **9      Оценочные средства для текущего контроля и аттестации студента**

### **9.1    Вопросы для оценки качества освоения дисциплины**

Примерный список вопросов к контрольным и зачётам по курсу:

- Дать определение одному из встретившихся в курсе понятий (list comprehension, итератор и т. п.).
- Дано текстовое описание алгоритма и его реализация на языке Python с ошибкой. Найти и исправить ошибку.
- Решить с помощью языка Python практическую задачу, связанную с обработкой текстовых данных (предполагаемая длина менее 250 строк кода) по текстовому описанию.
- Дано техническое задание для проекта, связанного с обработкой языковых данных. Описать, какими средствами лучше воспользоваться для его реализации.

## **10     Порядок формирования оценок по дисциплине**

Преподаватель или учебный ассистент каждую неделю оценивает самостоятельную работу студентов, проверяя домашние работы. Оценки за самостоятельную работу студента выставляются в рабочую ведомость. Накопленная оценка по десятибалльной шкале за самостоятельную работу определяется перед промежуточным или итоговым контролем —  $O_{\text{сам. р.}}$ .

Накопленная оценка за текущий контроль равна оценке за контрольную работу:

$$O_{\text{текущий}} = O_{\text{к/p 1}}.$$

Результирующая оценка за итоговый контроль в форме экзамена выставляется по следующей формуле, где  $O_{\text{экзамен}}$  — оценка за работу непосредственно на экзамене:



$$O_{итоговый} = 0,4 \cdot O_{экзамен} + 0,3 \cdot O_{текущий} + 0,3 \cdot O_{сам. р.}$$

Таким образом, в процентном отношении вклад имеющихся форм контроля выглядит так:

- экзамен — 40%
- текущий контроль — 30%
- самостоятельная работа — 30%

При подсчёте итоговой оценки промежуточные оценки (среднее арифметическое оценок за контрольные работы и среднее арифметическое оценок за домашние работы) не округляются.

## 11 Учебно-методическое и информационное обеспечение дисциплины

### 11.1 Базовый учебник

Курс лекций.

### 11.2 Основная литература

Марк Лутц. Изучаем Питон (4-е издание). Символ-плюс: М., 2011

### 11.3 Дополнительная литература

Томас Кормен, Чарльз Лейзерсон, Рональд Ривест, Клиффорд Штайн. Алгоритмы: построение и анализ. Вильямс: М., 2011

### Интернет-ресурсы

Документация по языку Python: <http://docs.python.org/>

Steven Bird, Ewan Klein, Edward Loper. Natural Language Processing with Python: <http://www.nltk.org/>

### 11.4 Программные средства

- текстовый редактор Notepad++ или любой другой, поддерживающий подсветку синтаксиса, переключение между разными кодировками и поиск с использованием регулярных выражений;
- интерпретатор языка Python (<http://www.python.org/download/>).

## 12 Материально-техническое обеспечение дисциплины

Для проведения семинаров необходим компьютерный класс, оснащённый проектором.