

**Учреждение образования
«Гомельский государственный университет
имени Франциска Скорины»**

Факультет математики и технологий программирования
Кафедра вычислительной математики и программирования

СОГЛАСОВАНО

Заведующий кафедрой

Кузьменков Д.С. Кузьменков
23.02. 2022 г.

СОГЛАСОВАНО

Декан факультета

Жогаль С.П. Жогаль
23.02. 2022 г.

**УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС
ПО ДИСЦИПЛИНЕ**

**ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON
И СОВРЕМЕННЫЕ WEB-ТЕХНОЛОГИИ**

для специальности

**1-40 01 01 Программное обеспечение информационных технологий
специализации**

**1-40 01 01 03 Базы данных и программное обеспечение информационных
систем**

Составитель:

Д.С. Кузьменков, кандидат физико-математических наук, доцент

Рассмотрено и утверждено
на заседании кафедры
вычислительной математики
и программирования

23.02.2022
Протокол № 8

Рассмотрено и утверждено
на заседании научно-методического совета университета
02.03. 2022 г.,
протокол № 3

02 Содержание учебно-методического комплекса по дисциплине «Язык программирования Python и современные web-технологии»

- 01 Титульный лист.
- 02 Содержание.
- 03 Пояснительная записка.
- 1 Теоретический раздел.
 - 1.1 Перечень теоретического материала.
 - 1.2 Материалы к лекциям.
- 2 Практический раздел.
 - 2.1 Перечень лабораторных работ.
 - 2.2 Задания для лабораторных работ.
- 3 Контроль знаний.
 - 3.1 Перечень вопросов к экзамену.
 - 3.2 Образец тестовых заданий по дисциплине.
- 4 Вспомогательный раздел.
 - 4.1 Учебная программа дисциплины.
 - 4.2 Перечень рекомендуемой литературы.

03 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к учебно-методическому комплексу
по дисциплине
«Язык программирования Python и современные web-технологии»
для специальности
1-40 01 01 программное обеспечение информационных технологий
специализации
1-40 01 01 03 Базы данных и программное обеспечение
информационных систем

Учебно-методический комплекс дисциплины специализации «Язык программирования Python и современные web-технологии» составлен в соответствии с образовательным стандартом высшего образования ОСВО 1-40 01 01-2013 для специальности 1-40 01 01 Программное обеспечение информационных технологий, утвержденным и введенным в действие постановлением Министерства образования Республики Беларусь от 30.08.2013, № 88; учебным планом Учреждения образования «Гомельский государственный университет имени Франциска Скорины» для специальности 1-40 01 01 Программное обеспечение информационных технологий, утверждённым 26.03.2014 г. (регистрационный номер № I-40-01-14), учебной программой учреждения высшего образования, утвержденной 22.05.2019 (регистрационный номер УД-13-2019-39/уч.).

Python – интерпретируемый объектно-ориентированный язык программирования с динамической семантикой. Язык является привлекательным для быстрой разработки приложений, так как в нем присутствуют высокоуровневые структуры данных и динамическая типизация. Python также используется в качестве сценарного языка для связи программных компонентов. В поставку в Python обширная стандартная библиотека для решения широкого круга задач. В интернете в свободном доступе находятся библиотеки для Python по различным предметным областям: интернет технологии, средства обработки текста, обработка изображений, механизмы доступа к БД, искусственный интеллект, пакеты для проведения сложных вычислений и т.д. Python имеет средство для интеграции с языками C, C++ и Java как путем встраивания (Embedding) интерпретатора программы на этих языках, так и на оборот – через использования библиотек, написанных на этих языках, в Python программах. Python поддерживает императивное, объектно-ориентированное и функциональное программирование. В настоящее время Python является одним из самых популярных языков программирования в мире.

Дисциплина специализации «Язык программирования Python и современные web-технологии» и соответствующий учебно-методический комплекс также предназначены для изучения современных языков и технологий программирования клиентских и серверных приложений. Основой программирования документов для web является специализированный язык разметки гипер-

текста HTML 5. Инструментом манипулирования страницами является язык сценариев JavaScript (его современный стандарт ECMAScript 6). Благодаря технологии CSS 3 средства форматирования документов можно отделить от содержания самого документа. Также в рамках изучения дисциплины студенты будут работать с LAMP (Linux Apache MySQL PHP) – набором популярного серверного программного обеспечения, широко используемым в web-разработке любой сложности. Всё вышеперечисленное говорит о необходимости изучения дисциплины «Язык программирования Python и современные web-технологии» специалистами в области программного обеспечения информационных технологий, прикладной математики и информатики.

Целью учебно-методического комплекса «Язык программирования Python и современные web-технологии» является овладение основами программирования на языке Python, изучение возможностей проектирования и разработки web-приложений с использованием современных web-технологий.

Задачами учебно-методического комплекса являются:

- овладение навыками программирования на языке Python;
- овладение современными Internet-технологиями;
- формирование умений и навыков в области работы с LAMP;
- формирование навыков программирования web-сайтов с использованием HTML 5, CSS 3, ECMAScript 6.

Материал дисциплины специализации и соответствующего учебно-методического комплекса базируется на ранее полученных студентами знаниях по таким дисциплинам, как «Основы алгоритмизации и программирования», «Основы конструирования программ», «Основы web-технологий», «Компьютерные системы управления документооборотом и web-технологии».

В результате изучения дисциплины специализации «Язык программирования Python и современные web-технологии» и соответствующего учебно-методического комплекса:

Студент должен знать:

- методы сертификации разрабатываемого программного продукта;
- основы настройки и разработки приложений в LAMP;
- основы программирования на языке Python;
- объектную модель браузера, методы управления окнами браузера, диалоговыми окнами;
- язык программирования JavaScript.

Студент должен уметь:

- проектировать приложения в LAMP;
- конструировать программы на языке программирования Python;
- использовать существующие прикладные программы (текстовые и графические редакторы, базы данных, средства коммуникации, и другое программное обеспечение);
- использовать современные средства программирования для создания оригинальных программных продуктов;

- обрабатывать события, которые генерируются в результате действий пользователя (щелчков мыши, нажатия клавиш и т.д.) или при изменении состояния документа;
- управлять окнами браузера, работать и диалоговыми окнами;
- создавать клиент-серверные приложения.

Студент должен владеть:

- навыками в области настройки и разработки приложений в LAMP
- новыми современными языками программирования;
- методами обеспечения заданного уровня качества программ;
- методами отладки программ;
- навыками программирования на языке программирования Python;
- умениями и навыками использования объектной модели браузера;
- умениями и навыками программирования на JavaScript.

Требования к академическим компетенциям специалиста

Специалист должен:

- АК-9. Уметь учиться, повышать свою квалификацию в течении всей жизни.

Требования к профессиональным компетенциям специалиста

Специалист должен быть способен:

Проектно-производственная и эксплуатационная деятельность

- ПК-1. Владеть современными технологиями анализа предметной области и разработки требований к создаваемым системам и программным средствам.

- ПК-4. Программировать на профессиональном уровне с учётом ресурсов и возможностей конкретного компьютера, требований стандартов, ограничений проекта.

- ПК-6. Владеть современными технологиями тестирования, отладки, верификации, аттестации и оценки качества программных средств.

- ПК-10. Владеть компьютерными методами сбора, хранения и обработки информации в сфере своей профессиональной деятельности.

- ПК-7. Владеть современными технологиями тестирования, верификации и управления качеством разрабатываемого программного обеспечения, методами сопровождения и эксплуатации программных средств.

Научно-исследовательская и образовательная деятельность

- ПК-15. Выполнять теоретические и экспериментальные исследования, различные виды моделирования автоматизируемых предметных областей.

Организационно-управленческая деятельность

- ПК-22. Взаимодействовать со специалистами смежных профилей.

В структурном отношении учебно-методический комплекс «Язык программирования Python и современные web-технологии» включает в себя четыре раздела: теоретический, практический, раздел контроля знаний, вспомогательный.

Теоретический раздел содержит основные положения, выносимые на лекции, и включающий в себя 13 тем, предназначенных для аудиторной работы со студентами (лекции преподавателя – 24 часа) и самостоятельного изучения тем, вынесенных за рамки аудиторных часов (УСР – 12 часов). Посредством этих тем студенты смогут изучить основы разработки приложений на современном объектно-ориентированном языке программирования Python, создавать современные web-приложения с использованием HTML5, CSS3, JavaScript.

Практический раздел включает в себя в соответствии с учебным планом дисциплины 20 часов лабораторных занятий. Практический раздел состоит из перечня лабораторных работ и заданий для лабораторных работ.

Предполагаются различные формы работы со студентами на лабораторных занятиях: устный опрос, тестовые задания, групповая дискуссия, подготовка презентаций и другие.

Указанные формы работы способствуют не только усвоению знаний и их репродуктивному воспроизведению, но и способности самостоятельно разрабатывать приложения на языке программирования Python, создавать современные web-приложения.

В разделе контроля знаний приводятся вопросы к экзамену по дисциплине «Язык программирования Python и современные web-технологии» для специальности 1-40 01 01 Программное обеспечение информационных технологий специализации 1-40 01 01 03 Базы данных и программное обеспечение информационных систем, и образец тестовых заданий, предназначенных для проверки уровня академических компетенций студентов по дисциплине. Они составлены в логической последовательности и охватывают практически все темы учебной дисциплины.

Вспомогательный раздел содержит необходимые элементы учебно-программной документации: учебную программу по дисциплине специализации «Язык программирования Python и современные web-технологии» с пояснительной запиской и содержанием учебного материала. Кроме того в данном разделе имеется перечень рекомендуемой литературы.

Учебно-методический комплекс учебной дисциплины «Язык программирования Python и современные web-технологии» адресуется студентам 3-го курса дневной формы обучения специальности 1-40 01 01 Программное обеспечение информационных технологий специализации 1-40 01 01 03 Базы данных и программное обеспечение информационных систем, студентам 4 курса заочной формы обучения специальности 1-40 01 01 Программное обеспечение информационных технологий специализации 1-40 01 01 03 Базы данных и программное обеспечение информационных систем, студентам 3 курса заочной сокращенной формы обучения специальности 1-40 01 01 Программное обеспечение информационных технологий специализации 1-40 01 01 03 Базы данных и программное обеспечение информационных систем.

1.1 Перечень теоретического материала по учебной дисциплине «Язык программирования Python и современные web-технологии»

1 LAMP

LAMP. Другие пакеты ПО (WAMP, SAMP). Создание таблиц. Построение запросов к БД. Экспорт и импорт таблиц. Сортировка, поиск в БД.

2 Язык программирования Python

Язык программирования Python. Версии Python. Основные реализации Python. Основные различия синтаксиса Python 2.0 и Python 3.0. Установка Python. Запуск Python.

3 Синтаксис языка программирования Python

Оператор условия и выбора. Циклы. Ключевые слова. Встроенные функции. Основные встроенные функции, выполняющие преобразование типов.

4 Работа с числовыми и строковыми данными в Python

Работа с числами. Работа со строками. Базовые операции над строками. Некоторые функции и методы обработки строк. Форматирование строк. Метод format.

5 Специфичные типы данных в Python

Списки. Основные функции и методы списков. Кортежи. Словари и работа с ними. Основные методы словарей. Множества. Операции над множествами. Операции, изменяющие множества. Отличия set и frozenset.

6 Индексы, срезы, функции и их аргументы

Индексы и срезы. Описание и работа с функциями в Python. Именованные и позиционные аргументы. Анонимные функции, инструкция lambda.

7 Обработка ошибок, работа с файлами, модули

Исключения в Python. Конструкция try-except. Байтовые строки. Работа с файлами. Работа с модулями. With...as – менеджеры контекста.

8 ООП в Python

ООП в Python. 3 принципа ООП. Одиночное и двойное подчёркивание. Перегрузка операторов. PEP 8 – руководство по написанию кода. Декораторы. Передача декоратором аргументов в функцию. Декорирование методов.

9 PyQt5

Набор библиотек PyQt5. Примеры использования. Создание окна. Подсказки. Заккрытие окна. Визуальные элементы (кнопки, надписи, строки редактирования).

10 Язык программирования JavaScript. Области видимости (Scope)

Глобальная область видимости. Функциональная область видимости. Вложенные области видимости. Экранирование (shadowing). Всплытие (hoisting). Глобальный объект. Блочная область видимости в ECMAScript 6. Замыкания. This и That.

11 Язык программирования JavaScript. Функции

Function Declarations. Function Expressions. Параметры и аргументы функций. IIFE (мгновенно выполняемые функциональные выражения). Стрелочные функции. Рекурсия. Каррирование. Мемоизация. Модули.

12 Язык программирования JavaScript. Объекты и наследование

Объекты в JavaScript. Свойство `__proto__`. Описание класса. Создание объектов различными способами. Свойство `prototype`. Геттеры и сеттеры. Псевдоклассическое наследование.

13 Информационная безопасность в сети интернет

Понятие информационной безопасности. Информационная безопасность и Интернет. Методы обеспечения информационной безопасности. Рекомендации по обеспечению безопасности информации в локальных сетях, подключенных к сети Интернет. Концепция информационной безопасности Республики Беларусь.

1.2 Материалы к лекциям по учебной дисциплине «Язык программирования Python и современные web-технологии»

5. Синтаксис языка Python. Ключевые слова. Встроенные функции.

Ключевые слова

false – ложь.
true – правда.
none – "пустой" объект.
and – логическое И.
with / as – менеджер контекста.
assert условие – возбуждает исключение, если условие ложно.
break – выход из цикла.
class – пользовательский тип, состоящий из методов и атрибутов.
continue – переход на следующую итерацию цикла.
def – определение функции.
del – удаление объекта.
elif – в противном случае, если.
else – иначе
except – перехватить исключение.
finally – вкупе с инструкцией *try*, выполняет инструкции независимо от того, было ли исключение или нет.
for – цикл *for*.
from – импорт нескольких функций из модуля.
global – позволяет сделать значение переменной, присвоенное ей внутри функции, доступным и за пределами этой функции.
if – если.
import – импорт модуля.
in – проверка на вхождение.
is – ссылаются ли 2 объекта на одно и то же место в памяти.
lambda – определение анонимной функции.
nonlocal – позволяет сделать значение переменной, присвоенное ей внутри функции, доступным в объемлющей инструкции.
not – логическое НЕ.
or – логическое ИЛИ.
pass – ничего не делающая конструкция.
raise – возбудить исключение.
return – вернуть результат.
try – выполнить инструкции, перехватывая исключения.
while – цикл *while*.
yield – определение функции–генератора.

Встроенные функции, выполняющие преобразование типов

bool(x) – преобразование к типу *bool*, использующая стандартную процедуру проверки истинности. Если *x* является ложным или опущен, возвращает

значение False, в противном случае она возвращает True.

bytearray([источник [, кодировка [ошибки]]) – преобразование к bytearray. Bytearray – изменяемая последовательность целых чисел в диапазоне $0 \leq X < 256$. Вызванная без аргументов, возвращает пустой массив байт.

bytes([источник [, кодировка [ошибки]]) – возвращает объект типа bytes, который является неизменяемой последовательностью целых чисел в диапазоне $0 \leq X < 256$. Аргументы конструктора интерпретируются как для bytearray().

complex([real[, imag]]) – преобразование к комплексному числу.

dict([object]) – преобразование к словарию.

float([X]) – преобразование к числу с плавающей точкой. Если аргумент не указан, возвращается 0.0.

frozenset([последовательность]) – возвращает неизменяемое множество.

int([object], [основание системы счисления]) – преобразование к целому числу.

list([object]) – создает список.

memoryview([object]) – создает объект memoryview.

object() – возвращает безликий объект, являющийся базовым для всех объектов.

range([start=0], stop, [step=1]) – арифметическая прогрессия от start до stop с шагом step.

set([object]) – создает множество.

slice([start=0], stop, [step=1]) – объект среза от start до stop с шагом step.

str([object], [кодировка], [ошибки]) – строковое представление объекта. Использует метод `__str__`.

tuple(obj) – преобразование к кортежу.

Основные встроенные функции

abs(x) – Возвращает абсолютную величину (модуль числа).

all(последовательность) – Возвращает True, если все элементы истинные (или, если последовательность пуста).

any(последовательность) – Возвращает True, если хотя бы один элемент – истина. Для пустой последовательности возвращает False.

bin(x) – Преобразование целого числа в двоичную строку.

chr(x) – Возвращает односимвольную строку, код символа которой равен x.

dir([object]) – Список имен объекта, а если объект не указан, список имен в текущей локальной области видимости.

divmod(a, b) – Возвращает частное и остаток от деления a на b.

exec(object[, globals[, locals]]) – Выполняет программный код на Python.

format(value[, format_spec]) – Форматирование (обычно форматирование строки).

help([object]) – Вызов встроенной справочной системы.

hex(x) – Преобразование целого числа в шестнадцатеричную строку.

input([prompt]) – Возвращает введенную пользователем строку. Prompt – подсказка пользователю.

len(x) – Возвращает число элементов в указанном объекте.

`max(iter, [args ...] * [, key])` – Максимальный элемент последовательности.
`min(iter, [args ...] * [, key])` – Минимальный элемент последовательности.
`oct(x)` – Преобразование целого числа в восьмеричную строку.
`open(file, mode='r', buffering=None, encoding=None, errors=None, new-line=None, closefd=True)` – Открывает файл и возвращает соответствующий поток.
`ord(c)` – Код символа.
`pow(x, y[, r])` – $(x ** y) \% r$ (x в степени y по модулю r).
`print([object, ...], *, sep=" ", end='\n', file=sys.stdout)` – Печать.
`round(X [, N])` – Округление до N знаков после запятой.
`sum(iter, start=0)` – Сумма членов последовательности.
`type(object)` – Возвращает тип объекта.

7. Установка и запуск Python.

Установка происходит стандартно без выбора дополнительных опций. После установки запускаем `idle`. Прямо в нём можно написать программный код и нажать клавишу «ввод». Можно сохранить программный код в файл и запускать уже файл. Для того, чтобы создать новое окно, в интерактивном режиме `idle` выбираем `File` → `New File` (или нажимаем `Ctrl+N`). В открывшемся окне вводим программный код. Нажимаем `F5` (или выберем в меню `idle` `Run` → `Run Module`) для запуска программы. Перед запуском `idle` предложит сохранить файл. Без сохранения программа не запустится.

8. Работа с числами.

Целые числа в Python 3 относятся к типу `int`. Над ними можно выполнять набор обычных математических операций:

$x + y$	Сложение
$x - y$	Вычитание
$x * y$	Умножение
x / y	Деление
$x // y$	Получение целой части от деления
$x \% y$	Остаток от деления
$-x$	Смена знака числа
$x ** y$	Возведение в степень

Целые числа в python 3 поддерживают длинную арифметику (напри-

мер, 3**150)

Над целыми числами также можно производить битовые операции

$x y$	Побитовое <i>или</i>
$x \wedge y$	Побитовое <i>исключающее или</i>
$x \& y$	Побитовое <i>и</i>
$x \ll n$	Битовый сдвиг влево
$x \gg y$	Битовый сдвиг вправо
$\sim x$	Инверсия битов

В Python 3 можно работать с числами в двоичной, восьмеричной, десятичной и шестнадцатеричной системах счисления.

Вещественные числа относятся к типу *float*. Данные типа *float* являются неточными, для более высокой точности используют типы *decimal* и *fraction*. Также вещественные числа не поддерживают длинную арифметику.

Модуль *math* содержит сложные математические функции:

Пример.

```
>>>import math
>>>math.pi          #Результат 3.141592653589793
>>>math.sqrt(9)      #Результат 3
```

Модуль *random* содержит генераторы случайных чисел и функции случайного выбора.

Пример.

```
>>>import random
>>>random.random()   #Результат 0.1516
```

В Python также встроены комплексные числа.

Пример.

```
>>>x=complex(1,2)
>>>print(x)          #Результат (1+2j)
```

Также для работы с комплексными числами используется модуль *cmath*.

9. Некоторые функции и методы обработки строк

- 1) **s.find(str,[start],[end])** – поиск подстроки в строке, возвращает номер первого вхождения или -1;
- 2) **s.rfind(str,[start],[end])** – поиск подстроки в строке, возвращает номер последнего вхождения или -1;
- 3) **s.split(символ)** – разбиение строки по разделителю;
- 4) **s.isdigit()** – проверяет, состоит ли строка из цифр;
- 5) **s.isalpha()** – проверяет, состоит ли строка из букв;
- 6) **s.islower()** – проверяет, состоит ли строка из символов в нижнем регистре;

- 7) **s.isupper()** – проверяет, состоит ли строка из символов в верхнем регистре;
- 8) **s.istitle()** – проверяет, начинаются ли слова в строке с заглавной буквы;
- 9) **s.upper()** – преобразование строки к верхнему регистру;
- 10) **s.lower()** – преобразование строки к нижнему регистру;
- 11) **ord(символ)** – перевод символа в его код ASCII;
- 12) **chr(число)** – перевод кода ASCII в символ;
- 13) **s.lstrip([chars])** – удаление пробельных символов в начале строки;
- 14) **s.rstrip([chars])** – удаление пробельных символов в конце строки;
- 15) **s.strip([chars])** – удаление пробельных символов в начале и в конце строки;
- 16) **s.swapcase()** – переводит символы нижнего регистра в верхний, а верхнего – в нижний;
- 17) **s.title()** – первую букву каждого слова переводит в верхний регистр, а верхнего – в нижний;
- 18) **s.format(*args,**kwargs)** – форматирование строки.

2.1 Перечень лабораторных работ по учебной дисциплине «Язык программирования Python и современные web-технологии»

1. Построение SQL-запросов в LAMP.
2. Язык программирования Python. Индексы, срезы, функции и их аргументы.
3. Язык программирования Python. Обработка ошибок, работа с файлами, модули.
4. Работа со строковой информацией в Python.
5. Работа со множествами в Python.
6. Работа с декораторами в Python.
7. PyQt5.
8. Язык программирования JavaScript. Работа с областями видимости (использование массивов).
9. Язык программирования JavaScript. Описание пользовательских функций.

2.2 Задания для лабораторных работ по учебной дисциплине «Язык программирования Python и современные web-технологии»

Задание 1. Создать, используя `phrmyadmin`, базу данных в соответствии с заданной предметной областью. Заполнить таблицы БД осмысленными записями (должно быть, как минимум, 4 записи в каждой таблице БД), написать запросы к БД (как минимум, 4 `select`, `insert`, `update`). Экспортировать созданную БД (в файл с расширением `sql`).

Задание 2. На языке Python 3 разработать 2 программы (модули) для обработки одномерных массивов (векторов), **используя списки**. Одна программа должна работать с целочисленным вектором, а вторая с вещественным вектором.

Задание 3. Разработанные во втором задании программы оформить, с использованием пользовательских функций. Обработать все возможные (по возможности) ошибки пользовательского ввода.

Задание 4. Написать программу на языке Python, реализующую решение задачи с использованием строкового типа данных. Программу просчитать для различных исходных данных.

Задание 5. Написать программу на языке Python, реализующую решение задачи с использованием структурированного типа данных: множества (**обязательно!**) согласно варианту. Программу просчитать для различных исходных данных.

Задание 6. Написать программу на языке Python с использованием **декораторов**. Должен быть описан один общий декоратор, который будет использоваться для декорирования нескольких функций (**как минимум, двух!**) и одного метода класса, также необходимо написать программу, в которой будет использовано несколько декораторов для одной функции или метода. Функции и их содержание (например, вывод текста на экран) должны соответствовать заданной предметной области.

Задание 7. Написать программу на языке Python (в виде небольшого окна) с использованием GUI (графического интерфейса пользователя). Программа должна быть создана **без использования дизайнера (т.е. весь код пишется вручную)**. Программа должна выполнять сложение двух чисел, которые пользователь вводит в два `QLineEdit`, результат помещается в `Label`. Еще один `Label` содержит пояснение, что делает программа (например, сложение двух чисел и т.д.). В окне программы должны присутствовать 3 кнопки: вычислить, очистить (очистка `QLineEdit` и `Label`, перевод курсора в первый `QLineEdit`) и выход (закрывает текущее окно). Обработать все возможные

ошибки пользовательского ввода (недопустимые символы в окнах редактирования, пустоту и т.д.).

Задание 8. На языке JS разработать 2 программы (модули) для обработки одномерных массивов (векторов), используя функции. Для ввода-вывода можно использовать сообщения и всплывающие окна (Alert, Prompt) или осуществлять ввод-вывод в Edit на HTML-странице.

Задание 9. На языке JS разработать 5 программ, используя функции.

1. Напишите функцию, которая возвращает последнюю цифру заданного целого числа в виде английского слова. Примеры: 512 → "two", 1024 → "four", 12309 → "nine"

2. Напишите функцию, которая переворачивает цифры заданного десятичного числа. Например: 256 → 652

3. Напишите функцию, которая находит все вхождения слова в текст

* Поиск с учетом регистра или без учета регистра

4. Напишите функцию для подсчета количества тегов <div> на веб-странице.

5. Напишите функцию, которая подсчитывает, сколько раз встречается заданное число в заданном массиве. Напишите тестовую функцию, чтобы проверить правильность работы функции.

Варианты для заданий 2, 3, 8

1. 1) Дан вещественный вектор $A(n)$. Найти число ненулевых элементов в векторе. На печать выдавать исходный вектор, число ненулевых элементов.
2) Если у вещественного вектора $A(n)$ хотя бы один элемент меньше, чем -2 , то все отрицательные компоненты заменить их квадратами, оставив все остальные без изменения. В противном случае вектор A умножить на 0.1 . На печать выдавать исходный вектор и полученный вектора.
3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом линейного выбора.
2. 1) Дан вещественный вектор $A(n)$. Найти количество элементов в векторе, абсолютная величина которых больше 7 . На печать выдавать исходный вектор, количество элементов, абсолютная величина которых больше 7 .
2) Дан вещественный вектор $A(n)$. Получить новый вектор путем умножения элементов стоящих за максимальным элементом на минимальный элемент вектора. На печать выдавать исходный вектор, минимальный элемент и полученный вектор.
3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с обменом.
3. 1) Дан вещественный вектор $A(n)$. Найти количество элементов этого вектора, больше среднего арифметического всех его элементов.
2) Дан вещественный вектор $A(n)$. Все элементы вектора предшествующие первому минимальному элементу умножить на 10 , если элемент минимальный по величине встречается в векторе более чем один раз. В противном случае вектор оставить без изменения. На печать выдавать исходный вектор, минимальный элемент, индекс первого и их количество, преобразованный вектор.
3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом стандартного обмена.
4. 1) Дан вещественный вектор $A(n)$. Подсчитать количество таких i , что $A[i]$ не меньше всех предыдущих элементов вектора ($A[1], A[2], \dots, A[i-1]$).
2) Дан вещественный вектор $A(n)$. Поменять местами минимальный и последний элементы вектора.
3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом парного обмена.
5. 1) Дан целочисленный вектор $A(n)$. Построить вектор $B(n)$, i -ый элемент которого равен среднему арифметическому первых i -элементов вектора A : $B[i] = (A[1] + \dots + A[i]) / i$.
2) Дан целочисленный вектор $A(n)$. Поменять местами максимальный и минимальный элементы вектора.
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом просеивания.
6. 1) Дан целочисленный вектор $A(n)$. Подсчитать сколько раз встречается в этом векторе максимальное по величине число.
2) Дан целочисленный вектор $A(n)$. Найти наибольшее из четных и количество нечетных чисел вектора. На печать выдавать исходный вектор и полученный результат.
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с подсчётом.
7. 1) Дан целочисленный вектор $A(n)$. Проверьте, есть ли в нем элементы равные 0 . Если есть, найдите номер первого из них, то есть наименьшее i , при котором $A[i] = 0$.
2) Дан целочисленный вектор $A(2n)$. Все четные числа, стоящие за максимальным элементом, домножить на минимальный элемент. На печать выдавать исходный, полученный вектора, максимальный элемент и его индекс, минимальный элемент.
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейной вставки.

8. 1) Дан вещественный вектор $A(n)$. Найти количество элементов вектора, меньших среднего арифметического всех его элементов.
2) Дан вещественный вектор $A(n)$. Получить новый вектор путем умножения элементов стоящих перед минимальным элементом на максимальный элемент вектора. На печать выдавать исходный вектор, максимальный, минимальный элементы и полученный вектор.
3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом линейного выбора.
9. 1) Дан целочисленный вектор $A(n)$. Проверьте, есть ли в нем отрицательные элементы. Если есть, найдите наибольшее i , при котором $A[i] < 0$.
2) Дан целочисленный вектор $A(n)$. Построить вещественный вектор $B(n)$, i -ый элемент которой равен среднему арифметическому двух соседних элементов вектора A : $B[i] = (A[i] + A[i+1])/2$, (и $B[10] = A[10]$).
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с обменом.
10. 1) Дан целочисленный вектор $A(n)$. Построить вектор $B(n)$, который содержит те же числа, что и вектор $A(n)$, но в котором все отрицательные элементы предшествуют всем неотрицательным.
2) Дан целочисленный вектор $A(n)$. Подсчитайте сколько раз встречается в этом векторе минимальное по величине число.
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом стандартного обмена.
11. 1) Дан целочисленный вектор $A(n)$. Поменять местами максимальный и первый элементы вектора.
2) Дан целочисленный вектор $A(n)$. Подсчитать количество таких i , что $A[i]$ не больше всех предыдущих элементов вектора ($A[1], A[2], \dots, A[i-1]$).
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом парного обмена.
12. 1) Дан целочисленный вектор $A(n)$. Подсчитать количество положительных элементов вектора, предшествующих максимальному элементу.
2) Дан целочисленный вектор $A(2n)$. Если в векторе сумма $S_1 = a_1 + a_2 + \dots + a_n$ равна сумме $S_2 = a_{n+1} + a_{n+2} + \dots + a_{2n}$, то поменять местами первый и последний элементы вектора. На печать выдавать исходный вектор, суммы S_1, S_2 , преобразованный вектор.
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом просеивания.
13. 1) Дан целочисленный вектор $A(n)$. Найти сумму отрицательных элементов вектора, следующих за максимальным элементом.
2) Дан целочисленный вектор $A(2n)$. Если в последовательности сумма $S_1 = a_1 + a_2 + \dots + a_n$ равна сумме $S_2 = a_{n+1} + a_{n+2} + \dots + a_{2n}$, то поменять местами максимальный и минимальный элементы вектора. На печать выдавать исходный вектор, суммы S_1, S_2 , преобразованный вектор.
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с подсчетом.
14. 1) Дан целочисленный вектор $A(n)$. Поменять местами первый отрицательный элемент вектора с последним положительным элементом вектора.
2) Дан целочисленный вектор $A(n)$. Найти наименьшее из четных чисел входящих в вектор. Определить его индекс и поменять местами с первым элементом. На печать выдавать исходный вектор, полученный результат и преобразованный вектор.
3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейной вставки.
15. 1) Дан целочисленный вектор $A(n)$. Найти количество положительных элементов, стоящих между минимальным и максимальным элементами вектора.
2) Если у вещественного вектора $A(n)$ хотя бы один элемент меньше, чем 2, то все от-

рицательные компоненты заменить их квадратами, оставив все остальные без изменения. В противном случае вектор A умножить на 0.5. На печать выдавать исходный и полученный вектора.

3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора.

16. 1) Дан вещественный вектор $A(n)$. Найти количество элементов в векторе, абсолютная величина которых меньше 5. На печать выдавать исходный вектор, количество элементов, абсолютная величина которых меньше 5.

2) Дан вещественный вектор $A(n)$. Получить новый вектор путем умножения элементов стоящих за минимальным элементом на максимальный элемент вектора. На печать выдавать исходный вектор, максимальный элемент и полученный вектор.

3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с обменом.

17. 1) Дан вещественный вектор $A(n)$. Найти количество элементов этого вектора, больше среднего арифметического всех его элементов.

2) Дан вещественный вектор $A(n)$. Все элементы вектора предшествующие первому максимальному элементу разделить на 2, если элемент максимальный по величине встречается в векторе более чем один раз. В противном случае вектор оставить без изменения. На печать выдавать исходный вектор, максимальный элемент, индекс первого и их количество, преобразованный вектор.

3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом стандартного обмена.

18. 1) Дан вещественный вектор $A(n)$. Поменять местами минимальный и первый элементы вектора.

2) Дан вещественный вектор $A(n)$. Подсчитать количество таких i , что $A[i]$ не меньше всех предыдущих элементов вектора ($A[1], A[2], \dots, A[i-1]$).

3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом парного обмена.

19. 1) Дан целочисленный вектор $A(n)$. Поменять местами первый и максимальный элементы вектора.

2) Дан целочисленный вектор $A(n)$. Построить вектор $B(n)$, i -ый элемент которого равен среднему арифметическому первых i -элементов вектора A : $B[i] = (A[1] + \dots + A[i]) / i$.

3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом просеивания.

20. 1) Дан целочисленный вектор $A(n)$. Подсчитать сколько раз встречается в этом векторе минимальное по величине число.

2) Дан целочисленный вектор $A(n)$. Найти наибольшее из нечетных и количество четных чисел вектора. На печать выдавать исходный вектор и полученный результат.

3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с подсчетом.

21. 1) Дан целочисленный вектор $A(2n)$. Все нечетные числа, стоящие за минимальным элементом, умножить на максимальный элемент. На печать выдавать исходный, полученный вектора, минимальный элемент и его индекс, максимальный элемент.

2) Дан целочисленный вектор $A(n)$. Проверьте, есть ли в нем элементы равные 0. Если есть, найдите номер первого из них, то есть наименьшее i , при котором $A[i] = 0$.

3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейной вставки.

22. 1) Дан вещественный вектор $A(n)$. Найти количество элементов вектора, больших среднего арифметического всех его элементов.

2) Дан вещественный вектор $A(n)$. Получить новый вектор путем умножения элементов стоящих перед максимальным элементом на минимальный элемент вектора. На печать выдавать исходный вектор, максимальный, минимальный элементы и получен-

- ный вектор.
- 3) Дан вещественный вектор $A(n)$. Отсортировать его элементы методом линейного выбора.
23. 1) Дан целочисленный вектор $A(n)$. Проверьте, есть ли в нем положительные элементы. Если есть, найдите наибольшее i , при котором $A[i] > 0$.
- 2) Дан целочисленный вектор $A(n)$. Подсчитать количество таких i , что $A[i]$ не больше всех предыдущих элементов вектора ($A[1], A[2], \dots, A[i-1]$).
- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с обменом.
24. 1) Дан целочисленный вектор $A(n)$. Построить вектор $B(n)$, который содержит те же числа, что и вектор $A(n)$, но в котором все неотрицательные элементы предшествуют всем отрицательным.
- 2) Дан целочисленный вектор $A(n)$. Подсчитайте, сколько раз встречается в этом векторе максимальное по величине число.
- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом стандартного обмена.
25. 1) Дан целочисленный вектор $A(n)$. Поменять местами максимальный и последний элементы вектора.
- 2) Дан целочисленный вектор $A(n)$. Построить вещественный вектор $B(n)$, i -ый элемент которой равен среднему арифметическому двух соседних элементов вектора A : $B[i] = (A[i] + A[i+1])/2$, (и $B[10] = A[10]$).
- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом парного обмена.
26. 1) Дан целочисленный вектор $A(n)$. Подсчитать количество положительных элементов вектора, предшествующих минимальному элементу.
- 2) Дан целочисленный вектор $A(2n)$. Если в векторе сумма $S_1 = a_1 + a_2 + \dots + a_n$ равна сумме $S_2 = a_{n+1} + a_{n+2} + \dots + a_{2n}$, то поменять местами максимальный и последний элементы вектора. На печать выдавать исходный вектор, суммы S_1, S_2 , преобразованный вектор.
- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом просеивания.
27. 1) Дан целочисленный вектор $A(n)$. Найти сумму отрицательных элементов вектора, следующих за минимальным элементом.
- 2) Дан целочисленный вектор $A(2n)$. Если в последовательности сумма $S_1 = a_1 + a_2 + \dots + a_n$ равна сумме $S_2 = a_{n+1} + a_{n+2} + \dots + a_{2n}$, то поменять местами первый и минимальный элементы вектора. На печать выдавать исходный вектор, суммы S_1, S_2 , преобразованный вектор.
- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с подсчетом.
28. 1) Дан целочисленный вектор $A(n)$. Поменять местами первый положительный элемент вектора с последним отрицательным элементом вектора.
- 2) Дан целочисленный вектор $A(n)$. Найти наименьшее из четных чисел входящих в вектор. Определить его индекс и поменять местами с последним элементом. На печать выдавать исходный вектор, полученный результат и преобразованный вектор.
- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейной вставки.
29. 1) Дан целочисленный вектор $A(n)$. Все четные числа, стоящие за минимальным элементом, умножить на максимальный элемент. На печать выдавать исходный, полученный вектора, минимальный элемент и его индекс, максимальный элемент.
- 2) Если у целочисленного вектора $A(n)$ хотя бы один элемент меньше, чем -3, то все положительные компоненты заменить их квадратами, оставив все остальные без изменения. В противном случае вектор A умножить на 2. На печать выдавать исходный и полученный вектора.

- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора.
30. 1) Дан целочисленный вектор $A(n)$. Проверьте, есть ли в нем нулевые элементы. Если есть, найдите наибольшее i , при котором $A[i]=0$.
- 2) Дан целочисленный вектор $A(n)$. Построить вектор $B(n)$, i -ый элемент которого равен среднему арифметическому первых i -элементов вектора A : $B[i] = (A[1] + \dots + A[i]) / i$.
- 3) Дан целочисленный вектор $A(n)$. Отсортировать его элементы методом линейного выбора с обменом.

3.1 Перечень вопросов к экзаменам по учебной дисциплине «Язык программирования Python и современные web-технологии»

1. LAMP (Linux, Apache, MySQL, PHP).
2. Введение в Python.
3. Синтаксис языка Python.
4. Оператор условия и выбор. Циклы.
5. Ключевые слова. Встроенные функции.
6. Основные различия синтаксиса Python 2 и Python 3.
7. Установка и запуск Python.
8. Работа с числами.
9. Работа со строками.
10. Форматирование строк. Метод format.
11. Списки. Основные функции и методы списков.
12. Индексы и срезы.
13. Кортежи.
14. Словари и работа с ними. Основные методы словарей.
15. Множества (set и frozenset).
16. Функции, их аргументы.
17. Исключения в Python. Конструкция try-except.
18. Байтовые строки. Ввод данных с клавиатуры.
19. Работа с файлами в Python.
20. Работа с модулями в Python.
21. With...as – менеджеры контекста.
22. ООП в Python. 3 принципа ООП.
23. Перегрузка операторов в Python.
24. PEP 8 – руководство по написанию кода на Python.
25. Декораторы.
26. PyQt5. Примеры использования.

3.2 Образец тестовых заданий по учебной дисциплине «Язык программирования Python и современные web-технологии»

1. Тип `int` в Python имеет ограничение по длине:
 - а) не имеет ограничения
 - б) 32768
 - в) $2 \cdot 10^{17}$
 - г) $2 \cdot 10^{38}$
 - д) $2 \cdot 10^{53}$

2. Для чего служит ключевое слово `yield`?
 - а) для определения анонимной функции
 - б) для определения функции–генератора
 - в) для работы в отладочном режиме
 - г) для перехвата ошибок, возникающих в программе
 - д) для создания исключения

3. Для чего служит ключевое слово `raise`?
 - а) для определения анонимной функции
 - б) для определения функции–генератора
 - в) для работы в отладочном режиме
 - г) для перехвата ошибок, возникающих в программе
 - д) для создания исключения

4. В Python 3 можно работать с числами:
 - а) в двоичной, восьмеричной и шестнадцатеричной системах счисления
 - б) в двоичной, восьмеричной, десятичной и шестнадцатеричной системах счисления
 - в) в двоичной, восьмеричной и десятичной системах счисления
 - г) в восьмеричной, десятичной и шестнадцатеричной системах счисления
 - д) в двоичной, десятичной и шестнадцатеричной системах счисления

5. Для работы с комплексными числами в Python используется модуль:
 - а) `cp1`
 - б) `cpu`
 - в) `math`
 - г) `cmath`
 - д) `compll`

6. При работе со строками `\t` осуществляет:

- а) перевод страницы.
- б) вертикальную табуляцию
- в) перевод строки
- г) добавление пробела
- д) горизонтальную табуляцию

7. При работе со строками `\n` осуществляет:

- а) перевод страницы.
- б) вертикальную табуляцию
- в) перевод строки
- г) добавление пробела
- д) горизонтальную табуляцию

8. Функция `s.split(символ)`:

- а) удаляет указанный символ в начале строки
- б) удаляет указанный символ в начале и конце строки
- в) удаляет указанный символ в конце строки
- г) удаляет указанный символ во всей строке
- д) разбивает строку по разделителю

9. Функция `s.swapcase()`:

- а) переводит символы верхнего регистра в нижний, а нижнего - в верхний
- б) переводит символы к нижнему регистру
- в) переводит символы к верхнему регистру
- г) переводит символы верхнего регистра в нижний
- д) переводит символы нижнего регистра в верхний

10. `a=[1,2,3]` - это описание:

- а) словаря
- б) функции
- в) множества
- г) кортежа
- д) списка

Учреждение образования
«Гомельский государственный университет имени Франциска Скорины»

УТВЕРЖДАЮ

Проректор по учебной работе
УО «ГГУ имени Ф. Скорины»



И.В. Семченко

(подпись)

в день утверждения)

Регистрационный № УД-13-2019-39/уч.

**ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON
И СОВРЕМЕННЫЕ WEB-ТЕХНОЛОГИИ**

Учебная программа учреждения высшего образования
по учебной дисциплине для специальности

1-40 01 01 «Программное обеспечение информационных технологий»
(код специальности) (наименование специальности)

специализации 1-40 01 01 03 «Базы данных и программное обеспечение
(код специализации) (наименование специализации)
информационных систем»

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-40 01 01-2013. Высшее образование. Первая ступень. Специальность 1-40 01 01 Программное обеспечение информационных технологий, утвержденного и введенного в действие постановлением Министерства образования Республики Беларусь от 30.08.2013, № 88; учебного плана Учреждения образования «Гомельский государственный университет имени Франциска Скорины». Специальность 1-40 01 01 Программное обеспечение информационных технологий, рег. № I-40-01-14, от 26.03.2014 г.

СОСТАВИТЕЛЬ:

Д.С. Кузьменков — заведующий кафедрой вычислительной математики и программирования учреждения образования «Гомельский государственный университет имени Франциска Скорины», к.ф.-м.н., доцент.

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой вычислительной математики и программирования учреждения образования «Гомельский государственный университет имени Франциска Скорины»

(протокол № 11 от 14 мая 2019 г.);

Научно-методическим советом учреждения образования «Гомельский государственный университет имени Франциска Скорины»

(протокол № 8 от 14.05 2019 г.)

Зам. директора "ЦБА - Гомель - Парк"
Архивisto



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Python – интерпретируемый объектно-ориентированный язык программирования с динамической семантикой. Язык является привлекательным для быстрой разработки приложений, так как в нем присутствуют высокоуровневые структуры данных и динамическая типизация. Python также используется в качестве сценарного языка для связи программных компонентов. В поставку в Python обширная стандартная библиотека для решения широкого круга задач. В интернете в свободном доступе находятся библиотеки для Python по различным предметным областям: интернет технологии, средства обработки текста, обработка изображений, механизмы доступа к БД, искусственный интеллект, пакеты для проведения сложных вычислений и т.д. Python имеет средство для интеграции с языками C, C++ и Java как путем встраивания (Embedding) интерпретатора программы на этих языках, так и на оборот – через использования библиотек, написанных на этих языках, в Python программах. Python поддерживает императивное, объектно-ориентированное и функциональное программирование. В настоящее время Python является одним из самых популярных языков программирования в мире.

Дисциплина специализации «Язык программирования Python и современные web-технологии» также предназначена для изучения современных языков и технологий программирования клиентских и серверных приложений. Основой программирования документов для web является специализированный язык разметки гипертекста HTML 5. Инструментом манипулирования страницами является язык сценариев JavaScript (его современный стандарт ECMAScript 6). Благодаря технологии CSS 3 средства форматирования документов можно отделить от содержания самого документа. Также в рамках изучения дисциплины студенты будут работать с LAMP (Linux Apache MySQL PHP) – набором популярного серверного программного обеспечения, широко используемым в web-разработке любой сложности. Всё вышеперечисленное говорит о необходимости изучения дисциплины «Язык программирования Python и современные web-технологии» специалистами в области программного обеспечения информационных технологий, прикладной математики и информатики.

Целью дисциплины специализации «Язык программирования Python и современные web-технологии» является овладение основами программирования на языке Python, изучение возможностей проектирования и разработки web-приложений с использованием современных web-технологий.

Задачами дисциплины специализации являются:

- овладение навыками программирования на языке Python;
- овладение современными Internet-технологиями;
- формирование умений и навыков в области работы с LAMP;
- формирование навыков программирования web-сайтов с использованием HTML 5, CSS 3, ECMAScript 6.

Материал дисциплины специализации базируется на ранее полученных студентами знаниях по таким дисциплинам, как «Основы алгоритмизации и программирования», «Основы конструирования программ», «Основы web-технологий», «Компьютерные системы управления документооборотом и web-технологии».

В результате изучения дисциплины специализации «Язык программирования Python и современные web-технологии»:

Студент должен знать:

- методы сертификации разрабатываемого программного продукта;
- основы настройки и разработки приложений в LAMP;
- основы программирования на языке Python;
- объектную модель браузера, методы управления окнами браузера, диалоговыми окнами;
- язык программирования JavaScript.

Студент должен уметь:

- проектировать приложения в LAMP;
- конструировать программы на языке программирования Python;
- использовать существующие прикладные программы (текстовые и графические редакторы, базы данных, средства коммуникации, и другое программное обеспечение);
- использовать современные средства программирования для создания оригинальных программных продуктов;
- обрабатывать события, которые генерируются в результате действий пользователя (щелчков мыши, нажатия клавиш и т.д.) или при изменении состояния документа;
- управлять окнами браузера, работать с диалоговыми окнами;
- создавать клиент-серверные приложения.

Студент должен владеть:

- навыками в области настройки и разработки приложений в LAMP
- новыми современными языками программирования;
- методами обеспечения заданного уровня качества программ;
- методами отладки программ;
- навыками программирования на языке программирования Python;
- умениями и навыками использования объектной модели браузера;
- умениями и навыками программирования на JavaScript.

Требования к академическим компетенциям специалиста

Специалист должен:

- АК-9. Уметь учиться, повышать свою квалификацию в течении всей жизни.

Требования к профессиональным компетенциям специалиста

Специалист должен быть способен:

Проектно-производственная и эксплуатационная деятельность

– ПК-1. Владеть современными технологиями анализа предметной области и разработки требований к создаваемым системам и программным средствам.

– ПК-4. Программировать на профессиональном уровне с учётом ресурсов и возможностей конкретного компьютера, требований стандартов, ограничений проекта.

– ПК-6. Владеть современными технологиями тестирования, отладки, верификации, аттестации и оценки качества программных средств.

– ПК-10. Владеть компьютерными методами сбора, хранения и обработки информации в сфере своей профессиональной деятельности.

– ПК-7. Владеть современными технологиями тестирования, верификации и управления качеством разрабатываемого программного обеспечения, методами сопровождения и эксплуатации программных средств.

Научно-исследовательская и образовательная деятельность

– ПК-15. Выполнять теоретические и экспериментальные исследования, различные виды моделирования автоматизируемых предметных областей.

Организационно-управленческая деятельность

– ПК-22. Взаимодействовать со специалистами смежных профилей.

Форма получения образования – дневная. Дисциплина изучается студентами 3 курса специальности 1-40 01 01 «Программное обеспечение информационных технологий» в 6 семестре. Общее количество часов – 126 (3.5 зач. ед.); аудиторное количество часов – 56, из них: лекции – 24, лабораторные занятия – 20, управляемая самостоятельная работа (УСР) – 12. Форма отчётности – экзамен (6 семестр).

Форма получения образования – заочная. Дисциплина изучается студентами 4 курса специальности 1-40 01 01 «Специальность 1-40 01 01 Программное обеспечение информационных технологий» в 7–8 семестрах. Общее количество часов – 216 – 126 (3.5 зач.ед.); аудиторное количество часов – 14, из них: лекции – 10, лабораторные занятия – 4. Форма отчётности – экзамен (8 семестр).

Форма получения образования – заочная (сокращённая). Дисциплина изучается студентами 3, 4 курса специальности 1-40 01 01 «Специальность 1-40 01 01 Программное обеспечение информационных технологий» в 6–7 семестрах. Общее количество часов – 126 (3.5 зач.ед.); аудиторное количество часов – 14, из них: лекции – 10, лабораторные занятия – 4. Форма отчётности – экзамен (7 семестр).

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

1 LAMP

LAMP. Другие пакеты ПО (WAMP, SAMP). Создание таблиц. Построение запросов к БД. Экспорт и импорт таблиц. Сортировка, поиск в БД.

2 Язык программирования Python

Язык программирования Python. Версии Python. Основные реализации Python. Основные различия синтаксиса Python 2.0 и Python 3.0. Установка Python. Запуск Python.

3 Синтаксис языка программирования Python

Оператор условия и выбора. Циклы. Ключевые слова. Встроенные функции. Основные встроенные функции, выполняющие преобразование типов.

4 Работа с числовыми и строковыми данными в Python

Работа с числами. Работа со строками. Базовые операции над строками. Некоторые функции и методы обработки строк. Форматирование строк. Метод format.

5 Специфичные типы данных в Python

Списки. Основные функции и методы списков. Кортежи. Словари и работа с ними. Основные методы словарей. Множества. Операции над множествами. Операции, изменяющие множества. Отличия set и frozenset.

6 Индексы, срезы, функции и их аргументы

Индексы и срезы. Описание и работа с функциями в Python. Именованные и позиционные аргументы. Анонимные функции, инструкция lambda.

7 Обработка ошибок, работа с файлами, модули

Исключения в Python. Конструкция try-except. Байтовые строки. Работа с файлами. Работа с модулями. With...as – менеджеры контекста.

8 ООП в Python

ООП в Python. 3 принципа ООП. Одиночное и двойное подчёркивание. Перегрузка операторов. PEP 8 – руководство по написанию кода. Декораторы. Передача декоратором аргументов в функцию. Декорирование методов.

9 PyQt5

Набор библиотек PyQt5. Примеры использования. Создание окна. Подсказки. Заккрытие окна. Визуальные элементы (кнопки, надписи, строки редактирования).

10 Язык программирования JavaScript. Области видимости (Scope)

Глобальная область видимости. Функциональная область видимости. Вложенные области видимости. Экранирование (shadowing). Всплытие (hoisting). Глобальный объект. Блочная область видимости в ECMAScript 6. Замыкания. This и That.

11 Язык программирования JavaScript. Функции

Function Declarations. Function Expressions. Параметры и аргументы функций. IIFE (мгновенно выполняемые функциональные выражения). Стрелочные функции. Рекурсия. Каррирование. Мемоизация. Модули.

12 Язык программирования JavaScript. Объекты и наследование

Объекты в JavaScript. Свойство `__proto__`. Описание класса. Создание объектов различными способами. Свойство `prototype`. Геттеры и сеттеры. Псевдоклассическое наследование.

13 Информационная безопасность в сети интернет

Понятие информационной безопасности. Информационная безопасность и Интернет. Методы обеспечения информационной безопасности. Рекомендации по обеспечению безопасности информации в локальных сетях, подключенных к сети Интернет. Концепция информационной безопасности Республики Беларусь.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
1	LAMP (4 ч) 1 Создание таблиц. 2 Построение запросов к БД. 3 Экспорт и импорт таблиц. 4 Сортировка, поиск в БД.	2	–	2	–	Цифровой проектор	[1] [2]	Защита отчетов по лабораторным работам
2	Язык программирования Python (2 ч) 1 Основные реализации Python. 2 Основные различия синтаксиса Python 2.0 и Python 3.0. 3 Установка Python. 4 Запуск Python.	2	–	–	–	Цифровой проектор	[3] [4] [5] [6] [7] [15]	
3	Синтаксис языка программирования Python (4 ч) 1 Оператор условия и выбора. 2 Циклы. 3 Ключевые слова. 4 Встроенные функции.	2	–	2	–	Цифровой проектор	[3] [4] [5] [6] [15]	Защита отчетов по лабораторным работам
4	Работа с числовыми и строковыми данными в Python (4 ч) 1 Работа с числами. 2 Работа со строками. 3 Базовые операции над строками. 4 Форматирование строк.	2	–	2	–	Цифровой проектор	[3] [4] [5] [6] [7] [15]	Защита отчетов по лабораторным работам
5	Специфичные типы данных в Python (4 ч) 1 Списки. 2 Кортежи. 3 Словари и работа с ними. 4 Множества.	2	–	2	–	Цифровой проектор	[3] [4] [5] [6] [7]	Защита отчетов по лабораторным работам

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
6	Индексы, срезы, функции и их аргументы (6 ч) 1 Индексы и срезы. 2 Описание и работа с функциями в Python. 3 Именованные и позиционные аргументы. 4 Анонимные функции, инструкция lambda.	2	–	2	2	Цифровой проектор	[3] [4] [5] [6] [7] [15] [16]	Защита отчетов по лабораторным работам Групповая консультация
7	Обработка ошибок, работа с файлами, модули (4ч) 1 Исключения в Python. 2 Работа с файлами. 3 Работа с модулями. 4 With...as – менеджеры контекста.	–	–	2	2	–	[3] [4] [5] [6] [7] [15]	Защита отчетов по лабораторным работам Групповая консультация
8	ООП в Python (4 ч) 1 ООП в Python. 2 Перегрузка операторов. 3 PEP 8 – руководство по написанию кода. 4 Декораторы.	2	–	2	–	Цифровой проектор	[3] [4] [5] [6] [7]	Защита отчетов по лабораторным работам
9	PyQt5 (4 ч) 1 Набор библиотек PyQt5. 2 Примеры использования. 3 Визуальные элементы (кнопки, надписи, строки редактирования).	–	–	2	2	–	[3] [4] [5] [6] [7] [16]	Защита отчетов по лабораторным работам Групповая консультация
10	Язык программирования JavaScript. Области видимости (Scope) (6 ч.) 1 Глобальная область видимости. 2 Функциональная область видимости. 3. Всплытие (hoisting). 4 Блочная область видимости в ECMAScript 6.	2	–	2	2	Цифровой проектор	[8] [9] [10] [11] [12] [17]	Тестовые задания Групповая консультация

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
11	Язык программирования JavaScript. Функции (6 ч.) 1 Function Declarations. 2 Function Expressions. 3 Параметры и аргументы функций. 4 Стрелочные функции.	2	–	2	2	Цифровой проектор	[8] [9] [10] [11] [12] [17] [18]	Тестовые задания Групповая консультация
12	Язык программирования JavaScript. Объекты и наследование (4 ч.) 1 Объекты в JavaScript. 2 Свойство __proto__. 3 Создание объектов различными способами. 4 Свойство prototype.	2	–	–	2	Цифровой проектор	[8] [9] [10] [11] [12] [17] [18]	Тестовые задания Групповая консультация
13	Информационная безопасность в сети интернет (4 ч) 1 Понятие информационной безопасности. 2 Информационная безопасность и Интернет. 3 Методы обеспечения информационной безопасности. 4 Концепция информационной безопасности Республики Беларусь	4	–	–	–	Цифровой проектор	[13] [14]	
	ВСЕГО ЧАСОВ (56 ч.)	24		20	12			экзамен

преподаватель

Д.С. Кузьменков

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА

(заочная форма получения образования)

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
1	LAMP (2 ч) 1 Создание таблиц. 2 Построение запросов к БД. 3 Экспорт и импорт таблиц. 4 Сортировка, поиск в БД.	2	–	–	–	Цифровой проектор	[1] [2]	Групповая консультация
2	Язык программирования Python 1 Основные реализации Python. 2 Основные различия синтаксиса Python 2.0 и Python 3.0. 3 Установка Python. 4 Запуск Python.	самостоятельное изучение				Цифровой проектор	[3] [4] [5] [6] [7] [15]	Групповая консультация
3	Синтаксис языка программирования Python (2 ч) 1 Оператор условия и выбора. 2 Циклы. 3 Ключевые слова. 4 Встроенные функции.	2	–	–	–	Цифровой проектор	[3] [4] [5] [6] [15]	Групповая консультация
4	Работа с числовыми и строковыми данными в Python 1 Работа с числами. 2 Работа со строками. 3 Базовые операции над строками. 4 Форматирование строк.	самостоятельное изучение					[3] [4] [5] [6] [7] [15]	Групповая консультация

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
5	Специфичные типы данных в Python (2 ч) 1 Списки. 2 Кортежи. 3 Словари и работа с ними. 4 Множества.	2	–	–	–	Цифровой проектор	[3] [4] [5] [6] [7]	Групповая консультация
6	Индексы, срезы, функции и их аргументы (2 ч) 1 Индексы и срезы. 2 Описание и работа с функциями в Python. 3 Именованные и позиционные аргументы. 4 Анонимные функции, инструкция lambda.	–	–	2	–		[3] [4] [5] [6] [7] [15] [16]	Защита отчетов по лабораторным работам Групповая консультация
7	Обработка ошибок, работа с файлами, модули (2 ч) 1 Исключения в Python. 2 Работа с файлами. 3 Работа с модулями. 4 With...as – менеджеры контекста.	2	–	–	–	Цифровой проектор	[3] [4] [5] [6] [7] [15]	Групповая консультация
8	ООП в Python 1 ООП в Python. 2 Перегрузка операторов. 3 PEP 8 – руководство по написанию кода. 4 Декораторы.	самостоятельное изучение					[3] [4] [5] [6] [7]	Групповая консультация
9	PyQt5 1 Набор библиотек PyQt5. 2 Примеры использования. 3 Визуальные элементы (кнопки, надписи, строки редактирования).	самостоятельное изучение					[3] [4] [5] [6] [7] [16]	Групповая консультация

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
10	Язык программирования JavaScript. Области видимости (Scope) (2 ч.) 1 Глобальная область видимости. 2 Функциональная область видимости. 3 Всплытие (hoisting). 4 Блочная область видимости в ECMAScript 6.	–	–	2	–	Цифровой проектор	[8] [9] [10] [11] [12] [17]	Защита отчетов по лабораторным работам Групповая консультация
11	Язык программирования JavaScript. Функции (2 ч.) 1 Function Declarations. 2 Function Expressions. 3 Параметры и аргументы функций. 4 Стрелочные функции.	2	–	–	–	Цифровой проектор	[8] [9] [10] [11] [12] [17] [18]	Групповая консультация
12	Язык программирования JavaScript. Объекты и наследование 1 Объекты в JavaScript. 2 Свойство __proto__. 3 Создание объектов различными способами. 4 Свойство prototype.	самостоятельное изучение				Цифровой проектор	[8] [9] [10] [11] [12] [17] [18]	Групповая консультация
13	Информационная безопасность в сети интернет 1 Понятие информационной безопасности. 2 Информационная безопасность и Интернет. 3 Методы обеспечения информационной безопасности. 4 Концепция информационной безопасности Республики Беларусь	самостоятельное изучение				Цифровой проектор	[13] [14]	Групповая консультация
	ВСЕГО ЧАСОВ (14 ч.)	10		4				экзамен

преподаватель

Д.С. Кузьменков

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА

(заочная сокращённая форма получения образования)

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
1	LAMP (2 ч) 1 Создание таблиц. 2 Построение запросов к БД. 3 Экспорт и импорт таблиц. 4 Сортировка, поиск в БД.	2	–	–	–	Цифровой проектор	[1] [2]	Групповая консультация
2	Язык программирования Python 1 Основные реализации Python. 2 Основные различия синтаксиса Python 2.0 и Python 3.0. 3 Установка Python. 4 Запуск Python.	самостоятельное изучение				Цифровой проектор	[3] [4] [5] [6] [7] [15]	Групповая консультация
3	Синтаксис языка программирования Python (2 ч) 1 Оператор условия и выбора. 2 Циклы. 3 Ключевые слова. 4 Встроенные функции.	2	–	–	–	Цифровой проектор	[3] [4] [5] [6] [15]	Групповая консультация
4	Работа с числовыми и строковыми данными в Python 1 Работа с числами. 2 Работа со строками. 3 Базовые операции над строками. 4 Форматирование строк.	самостоятельное изучение					[3] [4] [5] [6] [7] [15]	Групповая консультация

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
5	Специфичные типы данных в Python (2 ч) 1 Списки. 2 Кортежи. 3 Словари и работа с ними. 4 Множества.	2	–	–	–	Цифровой проектор	[3] [4] [5] [6] [7]	Групповая консультация
6	Индексы, срезы, функции и их аргументы (2 ч) 1 Индексы и срезы. 2 Описание и работа с функциями в Python. 3 Именованные и позиционные аргументы. 4 Анонимные функции, инструкция lambda.	–	–	2	–		[3] [4] [5] [6] [7] [15] [16]	Защита отчетов по лабораторным работам Групповая консультация
7	Обработка ошибок, работа с файлами, модули (2 ч) 1 Исключения в Python. 2 Работа с файлами. 3 Работа с модулями. 4 With...as – менеджеры контекста.	2	–	–	–	Цифровой проектор	[3] [4] [5] [6] [7] [15]	Групповая консультация
8	ООП в Python 1 ООП в Python. 2 Перегрузка операторов. 3 PEP 8 – руководство по написанию кода. 4 Декораторы.	самостоятельное изучение					[3] [4] [5] [6] [7]	Групповая консультация
9	PyQt5 1 Набор библиотек PyQt5. 2 Примеры использования. 3 Визуальные элементы (кнопки, надписи, строки редактирования).	самостоятельное изучение					[3] [4] [5] [6] [7] [16]	Групповая консультация

Номер раздела, темы, занятия	Название раздела, темы, занятия; перечень изучаемых вопросов	Количество аудиторных часов				Материальное обеспечение занятия (наглядные, методические пособия и др.)	Литература	Формы контроля знаний
		Лекции	практические (семинарские) занятия	лабораторные занятия	Контролируемая самостоятельная работа студента			
10	Язык программирования JavaScript. Области видимости (Scope) (2 ч.) 1 Глобальная область видимости. 2 Функциональная область видимости. 3 Всплытие (hoisting). 4 Блочная область видимости в ECMAScript 6.	–	–	2	–	Цифровой проектор	[8] [9] [10] [11] [12] [17]	Защита отчетов по лабораторным работам Групповая консультация
11	Язык программирования JavaScript. Функции (2 ч.) 1 Function Declarations. 2 Function Expressions. 3 Параметры и аргументы функций. 4 Стрелочные функции.	2	–	–	–	Цифровой проектор	[8] [9] [10] [11] [12] [17] [18]	Групповая консультация
12	Язык программирования JavaScript. Объекты и наследование 1 Объекты в JavaScript. 2 Свойство __proto__. 3 Создание объектов различными способами. 4 Свойство prototype.	самостоятельное изучение				Цифровой проектор	[8] [9] [10] [11] [12] [17] [18]	Групповая консультация
13	Информационная безопасность в сети интернет 1 Понятие информационной безопасности. 2 Информационная безопасность и Интернет. 3 Методы обеспечения информационной безопасности. 4 Концепция информационной безопасности Республики Беларусь	самостоятельное изучение				Цифровой проектор	[13] [14]	Групповая консультация
	ВСЕГО ЧАСОВ (14 ч.)	10		4				экзамен

преподаватель

Д.С. Кузьменков

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Примерный перечень лабораторных работ

1. «Построение SQL-запросов» в LAMP.
2. «Индексы, срезы, функции и их аргументы».
3. «Обработка ошибок, работа с файлами, модули».
4. Работа со строковой информацией в Python.
5. Работа со множествами в Python.
6. Работа с декораторами в Python.
7. «PyQt5».
8. JavaScript. Работа с областями видимости (использование массивов).
9. JavaScript. Описание пользовательских функций.

Рекомендуемые формы контроля знаний

1. Тестовые задания (дневная форма обучения).
2. Контрольные работы (заочная форма обучения).

Рекомендуемые темы тестовых заданий

1. Построение SQL-запросов в LAMP.
2. Синтаксис языка программирования Python.
3. Специфичные типы данных в Python.
4. Индексы, срезы, функции и их аргументы в Python.
5. Объектно-ориентированно программирование в Python.
6. Язык программирования JavaScript. Области видимости.
7. Язык программирования JavaScript. Функции.
8. Язык программирования JavaScript. Объекты и наследование.

Рекомендуемые темы контрольных работ

1. Построение SQL-запросов в LAMP.
2. Синтаксис языка программирования Python.
3. Специфичные типы данных в Python.
4. Индексы, срезы, функции и их аргументы в Python.
5. Объектно-ориентированно программирование в Python.
6. Язык программирования JavaScript. Функции.

Рекомендуемая литература

Основная

1. Кауфман, Д. SQL. Программирование / Д. Кауфман, Б. Матсик, К. Спенсер. – М.: Бином, 2012. – 744 с.
2. Аллен, Г. Тейлор. SQL для чайников / Аллен Г. Тейлор. – М.: Диалектика, Вильямс, 2015. – 416 с.

- 3 Бизли, Д.М. Python. Подробный справочник / Д.М. Бизли, пер. с. англ. – М. : Символ-Плюс, 2010. – 864 с.
- 4 Прохоренок, Н.А. PyQt. Создание оконных приложений на Python 3 / Н.А. Прохоренок. – – СПб.: БХВ-Петербург, 2011. – 243 с.
- 5 Лутц, М. Изучаем Python. 4-е издание / М. Лутц, пер. с. англ. – М. : Символ-Плюс, 2011. – 1271 с.
- 6 Форсье, Д. Django. Разработка веб-приложений на Python / Д. Форсье, П. Биссекс, У. Чан; пер. с англ. – М. : Символ-Плюс, 2010. – 456 с.
- 7 Хахаев, И. А. Java. Практикум по алгоритмизации и программированию на Python / И. А. Хахаев. – М. : ALT Linux, 2010. – 126 с.
- 8 Дмитриева, М. В. Самоучитель JavaScript / М. В. Дмитриева. – СПб.: БХВ-Петербург, 2001. – 512 с.
- 9 Флэнаган, Д. JavaScript. Подробное руководство. / Д. Флэнаган. – М.: Символ-Плюс, 2008. – 984 с.
- 10 Дмитриева, М. JavaScript: быстрый старт / М. Дмитриева. – СПб.: БХВ-Петербург, 2002. – 328 с.
- 11 Дмитриева, М. В. JavaScript. Экспресс-курс / М. В. Дмитриева. – СПб.: БХВ-Петербург, 2004. – 336 с.
- 12 Флэнаган, Д. ECMAScript 6. Подробное руководство. / Д. Флэнаган. – М.: Символ-Плюс, 2018. – 984 с.
- 13 Sb.by Концепция информационной безопасности Республики Беларусь [Электронный ресурс]. – Режим доступа: <https://www.sb.by/articles/kontseptsiya-informatsionnoy-bezopasnosti-respubliki-belarus.html> /– Дата доступа – 02-05.2019.
- 14 Родичев, Ю.А. Нормативная база и стандарты в области информационной безопасности» / Ю.А. Родичев. – СПб: Питер, 2017. – 254 с.

Дополнительная

- 15 Чаплыгин, А. Н. Учимся программировать вместе с Питоном / А. Н. Чаплыгин. – М. : Интертраст, 2004. – 110 с.
- 16 Головатый, А. Django. Подробное руководство / А. Головатый, Дж Каплан-Мосс, пер. с. англ. – М. : Символ-Плюс, 2010. – 560 с.
- 17 Бранденбау, Д. JavaScript: сборник рецептов / Д. Бранденбау; пер. с англ. Е. Матвеева. – СПб.: Питер, 2000. – 416 с.
- 18 Дарнелл, Р. JavaScript: справочник / Р. Дарнелл. – СПб.: Питер, 2000. 192 с.

**Методические рекомендации по организации и выполнению УСП
по дисциплине «Язык программирования Python
и современные web-технологии»**

Для самостоятельного изучения выделяются следующие темы дисциплины «Язык программирования Python и современные web-технологии»:

- «Индексы, срезы, функции и их аргументы»;
- «Обработка ошибок, работа с файлами, модули»;
- «PyQt5»;
- «Язык программирования JavaScript. Области видимости (Scope)»;
- «Язык программирования JavaScript. Функции»;
- «Язык программирования JavaScript. Объекты и наследование».

Самостоятельное изучение данных тем преследует следующие цели:

- овладение навыками практической работы с индексами и срезами в языке программирования Python;
- изучение информации об обработке ошибок в языке программирования Python;
- овладение навыками практической работы с файлами в языке программирования Python;
- получение навыков в создании графического интерфейса пользователя с использованием библиотеки PyQt5;
- изучение информации о вложенных областях видимости, экранирование (shadowing) и замыканиях в JavaScript;
- изучение информации о рекурсии, каррировании и мемоизации в JavaScript;
- изучение информации о геттерах и сеттерах, псевдоклассическом наследовании в JavaScript.

Учебная программа УСП

1 Тема «Индексы, срезы, функции и их аргументы» – 2 часа

Цели: 1) овладеть знаниями по данной теме, овладение навыками практической работы с индексами и срезами в языке программирования Python; 2) подготовиться к выполнению лабораторной работы по теме «Индексы, срезы, функции и их аргументы».

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тестирование в составе теста по окончании курса «Язык программирования Python и современные web-технологии», защита лабораторной работы.

Учебно-методическое обеспечение – электронный вариант текстов лекций, а также [3], [4], [5], [6], [7], [15], [16] из списка рекомендуемой литературы.

2 Тема «Обработка ошибок, работа с файлами, модули» – 2 часа

Цели: 1) овладеть знаниями по данной теме, овладение навыками практической работы с файлами в языке программирования Python; 2) подготовиться к выполнению лабораторной работы по теме «Обработка ошибок, работа с файлами, модули».

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тестирование в составе теста по окончании курса «Язык программирования Python и современные web-технологии», защита лабораторной работы.

Учебно-методическое обеспечение – электронный вариант текстов лекций, а также [[3], [4], [5], [6], [7], [15] из списка рекомендуемой литературы.

3 Тема «PyQt5» – 2 часа

Цели: 1) овладеть знаниями по данной теме, овладение навыками практической работы в создании графического интерфейса пользователя с использованием библиотеки PyQt5; 2) подготовиться к выполнению лабораторной работы по теме «PyQt5».

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тестирование в составе теста по окончании курса «Язык программирования Python и современные web-технологии», защита лабораторной работы.

Учебно-методическое обеспечение – электронный вариант текстов лекций, а также [3], [4], [5], [6], [7], [16] из списка рекомендуемой литературы.

4 Тема «Язык программирования JavaScript. Области видимости (Scope)» – 2 часа

Цели: 1) овладеть знаниями по данной теме, овладеть навыками работы с областями видимости в языке программирования JavaScript; 2) подготовиться к выполнению лабораторной работы по теме «Язык программирования JavaScript. Области видимости (Scope)».

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тестирование в составе теста по окончании курса «Язык программирования Python и современные web-технологии», защита лабораторной работы.

Учебно-методическое обеспечение – электронный вариант текстов лекций, а также [8], [9], [10], [11], [12], [17] из списка рекомендуемой литературы.

5 Тема «Язык программирования JavaScript. Функции» – 2 часа

Цели: 1) овладеть знаниями по данной теме, овладеть навыками описания пользовательских функций различными способами на языке программирования JavaScript; 2) подготовиться к выполнению лабораторной работы по теме «Язык программирования JavaScript. Функции».

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тестирование в составе теста по окончании курса «Язык программирования Python и современные web-технологии», защита лабораторной работы.

Учебно-методическое обеспечение – электронный вариант текстов лекций, а также [8], [9], [10], [11], [12], [17], [18] из списка рекомендуемой литературы.

6 Тема «Язык программирования JavaScript. Объекты и наследование» – 2 часа

Цели: 1) овладеть знаниями по данной теме, изучить информацию о геттерах и сеттерах, псевдоклассическом наследовании в JavaScript; 2) подго-

товиться к выполнению лабораторной работы по теме «Язык программирования JavaScript. Объекты и наследование».

Форма выполнения заданий – индивидуальная.

Форма контроля выполнения заданий – тестирование в составе теста по окончании курса «Язык программирования Python и современные web-технологии», защита лабораторной работы.

Учебно-методическое обеспечение – электронный вариант текстов лекций, а также [8], [9], [10], [11], [12], [17], [18] из списка рекомендуемой литературы.

**ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ
ПО ДИСЦИПЛИНЕ
«ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON
И СОВРЕМЕННЫЕ WEB-ТЕХНОЛОГИИ»
С ДРУГИМИ ДИСЦИПЛИНАМИ СПЕЦИАЛЬНОСТИ**

Название дисциплины, с которой требуется согласование	Название кафедры	Предложения об изменениях в содержании учебной программы по изучаемой учебной дисциплине	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)
Языки программирования	Кафедра Вычислительной математики и программирования	нет	Рекомендовать к утверждению учебную программу в представленном варианте, протокол № ____ от ____ 2019 г.
Основы алгоритмизации и программирования	Кафедра Вычислительной математики и программирования	нет	Рекомендовать к утверждению учебную программу в представленном варианте, протокол № ____ от ____ 2019 г.
Основы конструирования программ	Кафедра Вычислительной математики и программирования	нет	Рекомендовать к утверждению учебную программу в представленном варианте, протокол № ____ от ____ 2019 г.
Основы web-технологий	Кафедра Вычислительной математики и программирования	нет	Рекомендовать к утверждению учебную программу в представленном варианте, протокол № ____ от ____ 2019 г.
Компьютерные системы управления документооборотом и web-технологии	Кафедра Вычислительной математики и программирования	нет	Рекомендовать к утверждению учебную программу в представленном варианте, протокол № ____ от ____ 2019 г.

ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ
ПО ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ
НА ____/____ УЧЕБНЫЙ ГОД

№ /п	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры вычислительной математики и программирования (протокол № ____ от _____ 20__ г.)

Заведующий кафедрой
к.ф.-м.н., доцент

_____ Д.С. Кузьменков

УТВЕРЖДАЮ
Декан факультета математики и
Технологий программирования
УО «ГГУ им. Ф. Скорины»
к.ф.-м.н., доцент

_____ С.П. Жогаль

4.2 Перечень рекомендуемой литературы по учебной дисциплине «Язык программирования Python и современные web-технологии»

Основная

1. Кауфман, Д. SQL. Программирование / Д. Кауфман, Б. Матсик, К. Спенсер. – М.: Бином, 2012. – 744 с.
2. Аллен, Г. Тейлор. SQL для чайников / Аллен Г. Тейлор. – М.: Диалектика, Вильямс, 2015. – 416 с.
3. Бизли, Д.М. Python. Подробный справочник / Д.М. Бизли, пер. с англ. – М.: Символ-Плюс, 2010. – 864 с.
4. Прохоренок, Н.А. PyQt. Создание оконных приложений на Python 3 / Н.А. Прохоренок. – СПб.: БХВ-Петербург, 2011. – 243 с.
5. Лутц, М. Изучаем Python. 4-е издание / М. Лутц, пер. с англ. – М.: Символ-Плюс, 2011. – 1271 с.
6. Форсье, Д. Django. Разработка веб-приложений на Python / Д. Форсье, П. Биссекс, У. Чан; пер. с англ. – М.: Символ-Плюс, 2010. – 456 с.
7. Хахаев, И. А. Java. Практикум по алгоритмизации и программированию на Python / И. А. Хахаев. – М.: ALT Linux, 2010. – 126 с.
8. Дмитриева, М. В. Самоучитель JavaScript / М. В. Дмитриева. – СПб.: БХВ-Петербург, 2001. – 512 с.
9. Флэнаган, Д. JavaScript. Подробное руководство. / Д. Флэнаган. – М.: Символ-Плюс, 2008. – 984 с.
10. Дмитриева, М. JavaScript: быстрый старт / М. Дмитриева. – СПб.: БХВ-Петербург, 2002. – 328 с.
11. Дмитриева, М. В. JavaScript. Экспресс-курс / М. В. Дмитриева. – СПб.: БХВ-Петербург, 2004. – 336 с.
12. Флэнаган, Д. ECMAScript 6. Подробное руководство. / Д. Флэнаган. – М.: Символ-Плюс, 2018. – 984 с.
13. Sb.by Концепция информационной безопасности Республики Беларусь [Электронный ресурс]. – Режим доступа: <https://www.sb.by/articles/kontseptsiya-informatsionnoy-bezopasnosti-respubliki-belarus.html> / – Дата доступа – 02-05.2019.
14. Родичев, Ю.А. Нормативная база и стандарты в области информационной безопасности» / Ю.А. Родичев. – СПб: Питер, 2017. – 254 с.

Дополнительная

15. Чаплыгин, А. Н. Учимся программировать вместе с Питоном / А. Н. Чаплыгин. – М.: Интертраст, 2004. – 110 с.
16. Головатый, А. Django. Подробное руководство / А. Головатый, Дж. Каплан-Мосс, пер. с англ. – М.: Символ-Плюс, 2010. – 560 с.
17. Бранденбау, Д. JavaScript: сборник рецептов / Д. Бранденбау; пер. с англ. Е. Матвеева. – СПб.: Питер, 2000. – 416 с.
18. Дарнелл, Р. JavaScript: справочник / Р. Дарнелл. – СПб.: Питер, 2000. 192 с.