

# **БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**УТВЕРЖДАЮ**

Проректор по учебной работе

\_\_\_\_\_ А.Л. Толстик

\_\_\_\_\_

Регистрационный № УД-\_\_\_\_\_/ уч.

## **СКРИПТОВЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ (PYTHON)**

**Учебная программа учреждения высшего образования  
по учебной дисциплине для специальности второй ступени высшего  
образования (магистратуры) с углубленной подготовкой специалиста:**

**1-31 81 09 «Алгоритмы и системы обработки больших объемов  
информации».**

Учебная программа составлена на основе образовательного стандарта высшего образования ОСВО 1-31 81 09-2014 и учебного плана G31-219/уч. от 30.05.2016 г.

**СОСТАВИТЕЛИ:**

**Д.Ю. Косицин**, ассистент кафедры дискретной математики и алгоритмики Белорусского государственного университета.

**РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:**

**Кафедрой дискретной математики и алгоритмики** Белорусского государственного университета (протокол № 14 от 19 мая 2016 г.);

**Научно-методическим Советом** Белорусского государственного университета (протокол № 6 от 24 мая 2016 г.).

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Учебная программа по учебной дисциплине «Скриптовые языки программирования (Python)» разработана в соответствии с типовыми учебными планами и образовательными стандартами второй ступени высшего образования (магистратуры) с углубленной подготовкой специалиста по специальности 1-31 81 09 «Алгоритмы и системы обработки больших объемов информации» и относится к циклу дисциплин вузовского компонента.

Учебная дисциплина «Скриптовые языки программирования (Python)» дает представление о программировании на языке высокого уровня Python, а также дает обзор различных подходов к программированию: процедурного, функционального и объектно-ориентированного. Основой для изучения учебной дисциплины являются следующие учебные дисциплины первой ступени высшего образования: «Программирование», «Операционные системы», «Архитектура компьютеров», «Компьютерные сети».

**Цель преподавания** учебной дисциплины «Скриптовые языки программирования (Python)»: создание базы для разработки и проектирования магистрантами различных приложений с использованием широкого спектра инструментов скриптового языка высокого уровня Python.

**Основные задачи**, решаемые при изучении учебной дисциплины «Скриптовые языки программирования (Python)»:

- изучение базового синтаксиса и особенностей языка Python;
- изучение преимуществ и недостатков разработки приложений на скриптовых языках.

В результате изучения дисциплины магистрант должен:

*знать:*

- базовый синтаксис языка Python;
- основные стандартные модули языка;
- основы функционального и объектно-ориентированного программирования;
- особенности реализации программ на языке Python;

*уметь:*

- писать программы на языке Python;
- выбирать наиболее подходящий способ программирования;
- писать программу с использованием любого из трех следующих способов: процедурного, функционального, объектно-ориентированного;
- учитывать особенности языка Python для эффективного написания программ;

*владеть:*

– приемами разработки и проектирования приложений с использованием широкого спектра инструментов скриптового языка высокого уровня Python.

Освоение образовательной программы магистратуры должно обеспечить формирование следующих групп компетенций:

*академических компетенций* – углубленных научно-теоретических, методологических знаний и исследовательских умений, обеспечивающих разработку научно-исследовательских, инновационной деятельности, непрерывного самообразования (АК-1. Способность к самостоятельной профессиональной деятельности (анализ, сопоставление, систематизация, абстрагирование, моделирование, проверка достоверности данных, принятие решений и др.), готовность генерировать и использовать новые идеи. АК-2. Методологические знания и исследовательские умения, обеспечивающие решение прикладных задач и инновационной деятельности. АК-3. Способность к постоянному самообразованию);

*социально-личностных компетенций* – личностных качеств и умений следовать социально-культурным и нравственным ценностям; способностей к социальному, межкультурному взаимодействию, критическому мышлению; социальной ответственности, позволяющих решать социально-профессиональные, организационно-управленческие, воспитательные задачи (Магистр должен: СЛК-1. Учитывать социальные и нравственно-этические нормы в социально-профессиональной деятельности. СЛК-2. Быть способным к сотрудничеству и работе в команде. СЛК-3. Владеть коммуникативными способностями для работы в междисциплинарной и международной среде. СЛК-4. Совершенствовать и развивать свой интеллектуальный и общекультурный уровень, добиваться нравственного и физического совершенствования своей личности. СЛК-5. Формировать и аргументировать собственные суждения и профессиональную позицию. СЛК-6. Логично, аргументированно и ясно строить устную и письменную речь, использовать навыки публичной речи, ведения дискуссии и полемики. СЛК-7. Проявлять инициативу и креативность, в том числе в нестандартных ситуациях);

*профессиональных компетенций* – углубленных знаний по специальным дисциплинам и способностей решать сложные профессиональные задачи, задачи научно-исследовательской и научно-педагогической деятельности, разрабатывать и внедрять инновационные проекты, осуществлять непрерывное профессиональное самосовершенствование (Магистр должен быть способен: ПК-1. Квалифицированно использовать современные достижения по разработке и анализу алгоритмов и современные информационные технологии. ПК-2. Строить математические модели для

прикладных задач и проводить теоретическое и экспериментальное их исследование. ПК-3. Разрабатывать эффективные численные алгоритмы и интегрировать их в компьютерные системы. ПК-4. Обосновывать выбор методов и инструментов для решения прикладных задач. ПК-5. Обосновывать достоверность полученных результатов. ПК-6. Осваивать и реализовывать управленческие инновации в профессиональной деятельности. ПК-7. Формулировать выводы и рекомендации по применению современных достижений науки в инновационной деятельности).

Форма получения образования на второй ступени – дневная (очная).

В соответствии с образовательным стандартом специальности: 1-31 81 09 «Алгоритмы и системы обработки больших объемов информации» учебная программа предусматривает для изучения дисциплины всего 112 часов, из них 40 аудиторных часов, в том числе лекций – 20 часов, лабораторных занятий – 20 часов.

Форма текущей аттестации по учебной дисциплине – зачет.

## СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

### Раздел I. Введение в программирование на языке Python

#### Тема 1.1. Введение. Стандартные типы данных и конструкции языка.

Основные сведения о языке. Версии языка. Базовые типы и операции с ними. Условные конструкции и ветвления. Функции. Основные используемые среды разработки. Виртуальные окружения.

#### Тема 1.2. Стандартные контейнеры и коллекции.

Импортирование стандартных библиотек и модулей.

Контейнеры: списки, кортежи, словари, множества. Сокращенный синтаксис создания контейнеров. Коллекции: именованные кортежи, упорядоченные словари, очередь. Индексирование и диапазоны индексирования.

Некоторые алгоритмы, реализованные в стандартной библиотеке.

### Раздел II. Элементы функционального программирования

#### Тема 2.1. Синтаксис функций.

Функции и аргументы: позиционные и именованные аргументы, аргументы со значением по умолчанию. Комментарии и документация.

#### Тема 2.2. Обработка аргументов функций и упакованных данных.

Множественное присваивание. Упаковка и распаковка. Отличия в разных версиях языка.

Обработка аргументов командной строки.

Понятие модуля и пакета. Зарезервированные имена.

#### Тема 2.3. Организация работы со строками и файловой системой.

Кодировки и строки. Методы работы со строками.

Файлы. Организация ввода и вывода данных. Бинарный формат данных. Сериализация. Буферизация. Работа с файловой системой. Отличия в разных версиях языка.

Регулярные выражения.

Логирование.

#### Тема 2.4. Тестирование программ.

Организация тестирования программы. Юнит-тесты. Профилирование. Основные библиотеки для тестирования и профилирования.

## **Раздел III. Объектно-ориентированное программирование**

### **Тема 3.1. Объектно-ориентированная модель языка.**

Объектно-ориентированное программирование. Классы и методы. Атрибуты классов и экземпляров классов. Наследование. Способ разрешения имен при наследовании. Просмотр иерархии. Подмешиваемые классы. Специальные методы. Статические методы и методы классов. Контроль доступа к атрибутам. Отличия в разных версиях языка.

### **Тема 3.2. Модель памяти.**

Модель памяти. Изменяемые и неизменяемые объекты. Копирование объектов. Синглтоны.

### **Тема 3.3. Подходы к обработке ошибок.**

Подходы к обработке ошибок. Исключения и предупреждения. Менеджеры контекста.

### **Тема 3.4. Расширенный синтаксис функций и итераторов.**

Области видимости переменных. Замыкания. Анонимные функции. Декораторы.

Итераторы и генераторы. Дополнительные способы итерирования. Отличия в разных версиях языка.

### **Тема 3.5. Углубленная работа с классами.**

Дескрипторы. Работа с атрибутами классов. Метаклассы. Базовые классы для коллекций.

## **Раздел IV. Работа с математическими библиотеками**

### **Тема 4.1. Основные библиотеки для математических вычислений.**

Математические вычисления. Работа с библиотекой NumPy. Использование специальных типов данных. Векторные операции. Использование Cython.

### **Тема 4.2. Работа с массивами данных.**

Работа с данными. Визуализация данных. Библиотеки Pandas и Matplotlib.

Использование алгоритмов машинного обучения. Библиотека Scikit Learn.

## **Раздел V. Дополнительные сведения о языке**

### **Тема 5.1. Организация процесса выполнения программы.**

Параллельное программирование. Понятие GPL. Процессы. Потоки. Отличия в разных версиях языка.

Сборщик мусора. Интроспекция. Рефлексия.

### **Тема 5.2. Статическая модель языка.**

Организация языка. Абстрактные синтаксические деревья. Байткод. Расширения на языке C.



# УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

№ п/п	Название раздела, темы	Количество часов				Коли- чество часов УСР	Форма кон- троля знаний
		Аудиторные					
		Лек- ции	Пр. и сем. зан.	Лаб. зан.	Иное		
1	Введение в программирование на языке Python	4		4			
1.1	Введение. Стандартные типы данных и конструкции языка.	2					Устный опрос
	Лабораторная работа 1 «Базовый синтаксис языка программирования Python»			2			Защита лабораторной работы 1
1.2	Стандартные контейнеры и коллекции.	2					Устный опрос
	Лабораторная работа 2 «Работа с коллекциями и контейнерами языка Python»			2			Защита лабораторной работы 2
2	Элементы функционального программирования	4		4			
2.1	Синтаксис функций.	1					Устный опрос
2.2	Обработка аргументов функций и упакованных данных.	1					Устный опрос
2.3	Организация работы со строками и файловой системой.	1					Устный опрос
2.4	Тестирование программ.	1					Устный опрос
	Лабораторная работа 3 «Функциональный подход написания программ на языке Python»			4			Защита лабораторной работы 3
3	Объектно-ориентированное программирование	6		6			
3.1	Объектно-ориентированная модель языка.	2					Устный опрос
	Лабораторная работа 4 «Классы в языке Python»			2			Защита лабораторной работы 4
3.2	Модель памяти.	1					Устный опрос
3.3	Подходы к обработке ошибок.	1					Устный опрос
3.4	Расширенный синтаксис функций и итераторов.	1					Устный опрос
3.5	Углубленная работа с классами.	1					Устный опрос

	<i>Лабораторная работа 5 «Объектно-ориентированный подход написания программ на языке Python»</i>			4			Защита лабораторной работы 5
<b>4</b>	<b>Работа с математическими библиотеками</b>	<b>4</b>		<b>4</b>			
4.1	Основные библиотеки для математических вычислений.	2					Устный опрос
4.2	Работа с массивами данных.	2					Устный опрос
	<i>Лабораторная работа 6 «Обработка, анализ и визуализация данных с помощью Python»</i>			4			Защита лабораторной работы 6
<b>5</b>	<b>Дополнительные сведения о языке</b>	<b>2</b>		<b>2</b>			
5.1	Организация процесса выполнения программы.	1					Устный опрос
5.2	Статическая модель языка.	1					Устный опрос
	<i>Лабораторная работа 7 «Использование особенностей языка Python для написания эффективных программ»</i>			2			Защита лабораторной работы 7
<b>ИТОГО</b>		<b>20</b>		<b>20</b>			

## ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

### Рекомендуемая литература

#### *Основная*

1. Лутц, М. Программирование на Python: в 2 т. / М. Лутц // Символ-Плюс, 2011 – 2 т.
2. Pilgrim, M. Dive Into Python 3 / M. Pilgrim // Apress, 2009 – 412 p.

#### *Дополнительная*

1. Лутц, М. Изучаем Python. / М. Лутц // Символ-Плюс, 2011 – 1280 с.
2. Маккини, У. Python и анализ данных. / У. Маккини // ДМК Пресс, 2015. – 462 с.
3. Python documentation [Electronic resource]. – Mode of access: <https://docs.python.org/>.
4. Scikit-learn Machine Learning in Python [Electronic resource]. – Mode of access: <http://scikit-learn.org/>.

### Рекомендации по контролю качества усвоения знаний и проведению аттестации

На лекционных занятиях по учебной дисциплине «Скриптовые языки программирования (Python)» рекомендуется использование элементов проблемного обучения: проблемное изложение некоторых аспектов, использование частично-поискового метода.

### Перечни используемых средств диагностики результатов учебной деятельности

Для аттестации обучающихся на соответствие их персональных достижений поэтапным и конечным требованиям образовательной программы создаются фонды оценочных средств, включающие типовые задания для лабораторных работ, а также материалы для проведения дополнительных опросов и тестов. Оценочными средствами предусматривается оценка способности обучающихся к творческой деятельности, их готовность вести поиск решения новых задач, связанных с недостаточностью конкретных специальных знаний и отсутствием общепринятых алгоритмов.

Для диагностики компетенций в рамках учебной дисциплины рекомендуется использовать следующие формы:

- устная форма: опросы, устная защита лабораторных работ;
- письменная форма: отчеты по лабораторным работам, оценивание на основе модульно-рейтинговой системы.

Контрольные мероприятия проводятся в соответствии с учебно-методической картой дисциплины. В случае неявки на контрольное мероприятие по

уважительной причине студент вправе по согласованию с преподавателем выполнить его в дополнительное время. Для студентов, получивших неудовлетворительные оценки за контрольные мероприятия, либо не явившихся по неуважительной причине, по согласованию с преподавателем и с разрешения заведующего кафедрой мероприятие может быть проведено повторно.

Оценка текущей успеваемости рассчитывается как среднее за отчеты по домашним практическим упражнениям.

Итоговая аттестация предусматривает проведение зачета. При этом рекомендуется использовать оценивание успеваемости на основе модульно-рейтинговой системы.

# **ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ**

<p>Название учебной дисциплины, с которой требуется согласование</p>	<p>Название кафедры</p>	<p>Предложения об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине</p>	<p>Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола)</p>
<p>Программирование</p>	<p>Кафедра ИСУ</p>	<p>Нет</p>	<p>Оставить содержание учебной дисциплины без изменения, протокол № 7 от 20.01.2011 г.</p>
<p>Компьютерные сети. Архитектура компьютеров. Операционные системы.</p>	<p>Кафедра МСС</p>	<p>Нет</p>	<p>Оставить содержание учебной дисциплины без изменения, протокол № 8 от 13.04.2015 г.</p>

# ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ

на \_\_\_\_ / \_\_\_\_ учебный год

№№ Пп	Дополнения и изменения	Основание

Учебная программа пересмотрена и одобрена на заседании кафедры дискретной математики и алгоритмики (протокол № \_\_\_\_ от \_\_\_\_\_ 201\_ г.)

Заведующий кафедрой

\_\_\_\_\_  
(ученая степень, звание)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(И.О. Фамилия)

УТВЕРЖДАЮ

Декан факультета

\_\_\_\_\_  
(ученая степень, звание)

\_\_\_\_\_  
(подпись)

\_\_\_\_\_  
(И.О. Фамилия)