

Белорусский государственный университет

УТВЕРЖДАЮ

Проректор по учебной работе и
образовательным инновациям

«16» 02 2018 г.

Регистрационный № УД- 6603уч.

Программирование

Учебная программа учреждения высшего образования
по учебной дисциплине для специальностей:

- 1-21 06 01 Современные иностранные языки (по направлениям)
направления специальности
- 1-21 06 01-01 Современные иностранные языки (преподавание)

2018 г.

Учебная программа составлена на основе образовательного стандарта высшего образования первой ступени специальности 1-21 06 01 «Современные иностранные языки (по направлениям)», утвержденного постановлением Министерства образования РБ от 30.08.2013 № 88, и учебного плана БГУ № D21-086/уч. 2013 г. по специальности 1-21 06 01 «Современные иностранные языки (по направлениям)», направлению 1- 21 06 01 01 «Современные иностранные языки (преподавание)».

СОСТАВИТЕЛЬ:

Дубровина О.В., старший преподаватель

РЕКОМЕНДОВАНА К УТВЕРЖДЕНИЮ:

Кафедрой информационных технологий
(протокол № 10 от 27.06.2018);

Научно-методическим Советом БГУ(протокол № 7 от 13.07.2018)



ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Цели и задачи учебной дисциплины

Цель учебной дисциплины – обучение студентов основам программирования

В рамках поставленной цели **задачи учебной дисциплины** состоят в следующем:

1. дать введение в основные идеи и методы теории программирования;
2. изучить алгоритмы ветвления, циклов, элементы модульного программирования;
3. изучить средства языка Python для реализации простейших алгоритмов;
4. изучить среду для запуска и отладки программ на языке Python;
5. изучить основы работы со списками, строками, файлами, массивами, множествами, кортежами и словарями;
6. ознакомить студентов с основами объектно-ориентированного программирования;
7. возможности использования языка Python для решения задач компьютерной лингвистики.

Учебная дисциплина относится к циклу общенаучных и общепрофессиональных дисциплин (компонент учреждения высшего образования).

«Программирование» является базовым курсом, обеспечивающем в дальнейшем успешное освоение дисциплин «Компьютерная лингвистика», «Компьютерное изучение иностранного языка», «Интернет-лингвистика», «Компьютерные технологии в лингвистике».

В результате освоения учебной дисциплины студент должен:

знать: типы данных и операции над ними;

– операторы для организации линейных, разветвляющихся и повторяющихся вычислений;

– основные приемы обработки массивов, строк и других структурных типов данных;

– принципы чтения-записи информации в файл.;

уметь:

– реализовывать линейные, разветвляющихся и повторяющиеся вычисления в программе;

– обрабатывать одномерные и двумерные массивы и списки;

– обрабатывать текстовую информацию;

– осуществлять чтение информации из текстового файла и запись информации в файл;

– применять структурные типы данных для решения прикладных задач.

владеть: навыками работы в среде программирования языка Python, средствами запуска и отладки программы.

Требования к компетенциям

Освоение учебной дисциплины «Программирование» должно обеспечить формирование следующих академических, социально-личностных и профессиональных компетенций:

академические компетенции:

АК-1. Уметь применять базовые научно-теоретические знания для решения теоретических и практических задач.

АК-2. Владеть системным и сравнительным анализом.

АК-3. Обладать исследовательскими навыками в сфере решения задач программирования

АК-4. Уметь работать самостоятельно.

АК-5. Быть способным породить новые идеи (обладать креативностью).

социально-личностные компетенции:

СЛК-2. Быть способным к социальному взаимодействию.

СЛК-6. Уметь работать в команде.

профессиональные компетенции:

ПК-7. Заниматься научно-исследовательской деятельностью в сфере современной защиты информации.

ПК-9. Работать с научно-технической информацией с использованием современных информационных технологий.

ПК-10. Формулировать выводы и рекомендации по применению результатов научно-исследовательской работы.

ПК-11. Пользоваться глобальными информационными ресурсами.

ПК-29. Взаимодействовать со специалистами смежных профилей.

Структура учебной дисциплины

Структура содержания учебной дисциплины включает такие дидактические единицы, как темы, в соответствии с которыми разрабатываются и реализуются соответствующие лекционные и практические занятия. Примерная тематика практических занятий приведена в информационно-методической части.

Дисциплина изучается в третьем и четвертом семестрах. Всего на изучение учебной дисциплины «Программирование» отведено 136 часов, в том числе 66 аудиторных часов, из них: лекции – 30 часов, практические занятия – 28 часов, управляемая самостоятельная работа – 8 часов.

Трудоемкость учебной дисциплины составляет 3,5 зачетных единиц.

Форма текущей аттестации – зачет (третий и четвертый семестры).

СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Раздел 1 Основы теории программирования

Исторический обзор развития алгоритмизации и программирования. Понятие алгоритма. Примеры алгоритмов. Блок-схема. Алгоритмические языки низкого и высокого уровня. Компиляторы и интерпретаторы. Среды проектирования.

Раздел 2 Введение в язык Python. Реализация простейших алгоритмов.

Тема 2.1 Введение в язык Python

Исторический обзор развития языка Python. Возможности Python. Различия версий. Синтаксис языка. Типы данных. Операторы. Литералы. Переменные. Пространство имен. Операции.

Тема 2.2 Введение в среду программирования Python. Линейные алгоритмы.

Знакомство со средой проектирования и ее функциями. Операторы print и input, оператор присваивания. Приведение типов

Тема 2.3 Разветвляющиеся алгоритмы

Условный оператор. Синтаксис, формы записи. Примеры задач, использующих оператор ветвления.

Тема 2.4 Организация повторяющихся вычислений.

Операторы цикла for и while. Синтаксис. Примеры задач, использующих операторы цикла.

Тема 2.5 Функции и модули.

Назначение и синтаксис функции. Вызов функции в программе. Передача параметров в функцию. Локальные и глобальные переменные. Вывод результата функцией. Модули. Подключение и использование модуля.

Тема 2.6 Обработка исключений.

Типы ошибок в программе. Исключения. Оператор try ... except. Получение информации об исключении. Примеры исключений, генерируемых программой. Некоторые системные исключения Python. Генерация исключений.

Раздел 3 Структурные типы данных

Тема 3.1 Одномерные списки

Списки. Функции и методы для работы со списками. Алгоритмы обработки одномерных массивов. Примеры. Ввод-вывод списка. Индексы и срезы.

Тема 3.2 Двумерные списки

Алгоритмы обработки двумерных массивов. Особенности ввода-вывода двумерного списка. Примеры. Создание и обработка двумерного списка.

Тема 3.3 Строки

Строковый тип данных. Ввод-вывод строки. Форматирование строк. Функции и методы строк. Базовые операции. Доступ по индексу. Срезы. Примеры обработки строк.

Тема 3.4 Файлы

Общие принципы работы с файлами. Функция open и ее параметры. Чтение из файла. Запись в файл. Получение текстовой информации из файла.

Тема 3.5 Кортежи

Назначение кортежей. Операции с кортежами. Примеры.

Тема 3.6 Множества

Множества set и frozenset. Назначение и отличия. Операции над множествами. Примеры.

Тема 3.7 Словари.

Тип данных «словарь». Словари dict и основы работы с ними. Методы словарей.

Раздел 4. Основы объектно-ориентированного программирования

Классы и объекты. Поля, методы, свойства. Инкапсуляция, наследование, полиморфизм. Перегрузка операторов. Создание графического интерфейса пользователя.

УЧЕБНО-МЕТОДИЧЕСКАЯ КАРТА УЧЕБНОЙ ДИСЦИПЛИНЫ

Номер раздела, темы	Название раздела, темы	Количество аудиторных часов					Количество часов - УСП	Форма контроля знаний
		Лекции	Практические занятия	Семинарские занятия	Лабораторные занятия	Иное		
	2	3	4	5	6	7	8	9
1.	Основы теории программирования	2						
2.	Введение в язык Python. Реализация простейших алгоритмов.	12	10				2	Тест, защита лабораторных работ
2.1	Введение в язык Python	2						
2.2	Введение в среду программирования Python. Линейные алгоритмы	2	2					
2.3	Разветвляющиеся алгоритмы	2	2					
2.4	Организация повторяющихся вычислений	2	2					
2.5	Функции	2	2				1	
2.6	Обработка исключений	2	2				1	
3	Структурные типы данных	14	14				4	Тест, защита лабораторных работ
3.1	Одномерные списки	2	2					
3.2	Двумерные списки	2	2				2	
3.3	Строки	4	2				2	
3.4	Файлы	2	2					
3.5	Множества. Кортежи	2	2					
3.6	Словари	2	2					
4.	Основы объектно-ориентированного программирования	4	2				2	Устный опрос

ИНФОРМАЦИОННО-МЕТОДИЧЕСКАЯ ЧАСТЬ

Основная литература

1. Доусон, М. Программируем на Python / М. Доусон – СПб.: Питер, 2014. – 416 с.
2. Лутц, М. Python. Карманный справочник / М. Лутц – М.: Вильямс, 2017 – 320 с.
3. Златопольский, Д. Основы программирования на языке Python / Д. Златопольский – М.: ДМК, 2018. – 396 с.

Дополнительная литература

1. МакГрат, М. Программирование на Python для начинающих / М. МакГрат. – М.: Эксмо, 2015 – 192 с.
2. Шоу, З. Легкий способ выучить Python / З.Шоу. – М.: Эксмо, 2017. – 352 с.
3. Гэддис, Т. Начинаем программировать на Python / Т. Гэддис. – СПб.: БХВ-Петербург, 2019. – 768 с.
4. Хеллман, Д. Стандартная библиотека Python 3. Справочник с примерами / Д. Хеллман. – М.: Вильямс, 2018. – 1376 с.

Электронные ресурсы

1. Самоучитель Python. [Электронный ресурс]. – Режим доступа: <https://pythonworld.ru/samouchitel-python> – Дата доступа: 17.01.2019.
2. Учебник по языку Python 3.1. [Электронный ресурс]. – Режим доступа: <https://per8.ru/doc/tutorial-3.1/> – Дата доступа: 17.01.2019.
3. Python Tutor. [Электронный ресурс]. – Режим доступа: <http://www.pythontutor.com> – Дата доступа: 17.01.2019.

Перечень рекомендуемых средств диагностики

Оценка промежуточных учебных достижений студента производится по десятибалльной шкале в соответствии с критериями, утвержденными Министерством образования Республики Беларусь.

Для текущего контроля качества усвоения знаний студентами используется следующий диагностический инструментарий:

- тесты (в том числе электронные),
- устный опрос,
- письменные отчеты по лабораторным работам,
- защита лабораторных работ,
- проектный метод.

Формой текущей аттестации по дисциплине «Программирование» учебным планом предусмотрен зачет.

Используется оценка знаний студента по лабораторным работам, а также оценка по письменному тесту, для получения положительной оценки на который студент должен дать не менее, чем 75% правильных ответов.

Примерный перечень тем практических занятий

1. Введение в язык и среду программирования Python. Реализация линейных алгоритмов.
2. Организация разветвляющихся вычислений. Условный оператор.
3. Организация повторяющихся вычислений. Операторы цикла.
4. Списки. Работа с одномерными массивами.
5. Двумерные списки.
6. Функции.
7. Обработка исключений.
8. Строки.
9. Преобразование строк.
10. Файлы.
11. Кортежи.
12. Множества.
13. Словари.

Примерный перечень заданий для управляемой самостоятельной работы студентов

1. *Одномерные и двумерные списки.* Сформировать двумерный список. Отсортировать список по заданному критерию. Произвести форматированный вывод двумерного списка. Произвести выбор элемента по заданному критерию.
2. *Функции. Обработка исключений.* Разработать функцию, позволяющую проверять корректность вводимых пользователем данных.
3. *Строки. Файлы.* Считать из текстового файла информацию, произвести ее преобразование по заданному критерию. Записать результат в текстовый файл.
4. *Основы объектно-ориентированного программирования. Создание графического интерфейса пользователя.* Разработать графической интерфейс пользователя для одной из ранее реализованных задач.

Методы и технологии обучения

В процессе обучения используются: метод проектного обучения, метод портфолио, элементы, элементы метода эвристического обучения, методы развития критического мышления, методы группового обучения.

При чтении дисциплины используются мультимедийные технические средства, практические занятия проводятся в компьютерном классе.

Примерный перечень вопросов к экзамену/зачету

1. Алгоритм. Блок-схема. Программа. Определения и свойства.
2. Этапы разработки программ. Компиляторы и интерпретаторы.
3. Синтаксис языка Python. Комментарии. Примеры.
4. Пространство имен. Создание переменных. Оператор присваивания.
5. Типы данных языка Python. Динамическая типизация. Приведение типов.
6. Числовые типы данных. Операции над числами. Примеры.
7. Логические операции. Примеры.
8. Простейшие операции над строками. Примеры.
9. Условный оператор. Примеры.
10. Оператор цикла `for`. Примеры.
11. Оператор цикла `while`. Примеры.
12. Библиотека `math`.
13. Тип список. Операции над списками.
14. Одномерные списки. Примеры.
15. Двумерные списки. Особенности работы.
16. Функции.
17. Модули.
18. Обработка исключений.
19. Строковый тип данных. Операции над строками. Примеры.
20. Методы строк. Форматирование строк. Примеры.
21. Текстовые файлы. Операции чтения-записи. Примеры.
22. Бинарные файлы. Особенности работы. Примеры.
23. Тип множество. Определение множества. Добавление и удаление элементов. Примеры.
24. Перебор элементов множества. Операции над множествами.
25. Тип кортеж. Использование кортежей.
26. Тип словарь. Преобразование других типов в словарь. Примеры.
27. Получение, перебор и изменение элементов словаря. Примеры.
28. Особенности проектирования графического интерфейса пользователя.
29. Основы объектно-ориентированного программирования: наследование, инкапсуляция, полиморфизм.
30. Определение объектов в языке Python.
31. Компоненты для ввода-вывода текстовой информации.
32. Компоненты для организации действий пользователя.

ПРОТОКОЛ СОГЛАСОВАНИЯ УЧЕБНОЙ ПРОГРАММЫ УВО

Название учебной дисциплины, изучение которой связано с дисциплиной учебной программы	Кафедра, обеспечивающая изучение дисциплины	Предложения кафедры об изменениях в содержании учебной программы	Решение, принятое кафедрой, разработавшей учебную программу (с указанием даты и номера протокола) ¹
Основы информационных технологий	Кафедра информационных технологий	нет	Рекомендована к утверждению на заседании кафедры инф. технологий (пр. № 1 от 31.08.2018 г.)

¹При наличии предложений об изменениях в содержании учебной программы учреждения высшего образования по учебной дисциплине.

**ДОПОЛНЕНИЯ И ИЗМЕНЕНИЯ К УЧЕБНОЙ ПРОГРАММЕ ПО
ИЗУЧАЕМОЙ УЧЕБНОЙ ДИСЦИПЛИНЕ**

на ____ / ____ учебный год

№ п/п	Дополнения и изменения	Основание

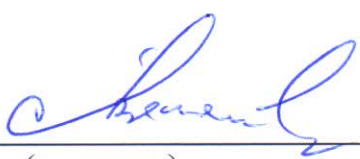
Учебная программа пересмотрена и одобрена на заседании кафедры
информационных технологий (протокол № _____ от _____ г.).

Заведующий кафедрой

кандидат физ.-мат. наук, доцент  В.А. Нифагин
(подпись)

УТВЕРЖДАЮ

Декан факультета

кандидат филол. наук, доцент  С.А. Важник
(подпись)