

СОВРЕМЕННЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ

1. Цель освоения дисциплины

Сформировать систему компетенций бакалавра на основе изучения наиболее современных языков программирования и практики использования полученных теоретических знаний для решения задач профессиональной деятельности.

2. Место дисциплины в структуре ОПОП

Дисциплина «Современные языки программирования» относится к вариативной части блока дисциплин и является дисциплиной по выбору.

Для освоения дисциплины «Современные языки программирования» обучающиеся используют знания, умения, способы деятельности и установки, сформированные в ходе изучения дисциплин «Алгебра», «Архитектура компьютера», «Вариативные методические системы обучения математике», «Вводный курс математики», «Высокоуровневые методы программирования», «Геометрия», «Дидактика математики с практикумом решения математических задач», «Дискретная математика», «ИКТ и медиаинформационная грамотность», «Информационные технологии», «Математическая логика и теория алгоритмов», «Математический анализ», «Методика обучения информатике», «Практикум решения задач по элементарной математике», «Программирование», «Теория вероятностей и математическая статистика», «Теория чисел», «Технологии обучения решению задач по математике повышенной сложности», «Философия», «Частная методика обучения математике», «3D-моделирование и печать», «Веб-дизайн и разработка интернет-приложений», «Естественнонаучная картина мира», «Инструментальные учебные среды», «Информационные системы», «Компьютерная графика и мультимедиа технологии», «Компьютерные сети», «Методика обучения математике на углубленном уровне», «Образовательная робототехника», прохождения практик «Производственная (исследовательская) практика», «Учебная (технологическая) практика».

Освоение данной дисциплины является необходимой основой для последующего изучения дисциплин «Дифференциальные уравнения», «Исследование операций», «Методика обучения информатике», «Теоретические основы информатики», «Численные методы», «Числовые системы», «Графы и их приложения», «Дополнительные главы математического анализа», «История математики», «Методика обучения информатике на углубленном уровне», «Основные алгебраические системы», «Основы теории решеток», «Пропедевтический курс обучения информатике», «Расширения полей», «Социальная информатика», «Теория функций комплексного переменного», «Физика», прохождения практик «Производственная (педагогическая) практика (Информатика)», «Производственная (педагогическая) практика (Математика)», «Производственная (преддипломная) практика», «Учебная (методическая) практика».

3. Планируемые результаты обучения

В результате освоения дисциплины выпускник должен обладать следующими компетенциями:

- способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач (УК-1);
- способен применять предметные знания в образовательном процессе (ПК-3).

В результате изучения дисциплины обучающийся должен:

знать

- особенности реализации объектно-ориентированного подхода в языке Python;
- концепцию, основные понятия и конструкции функционального программирования;
- особенности реализации функционального подхода в языке Python;

уметь

- разрабатывать объектно-ориентированные приложения на языке Python;
- разрабатывать приложения на языке Python в соответствии с функциональным подходом;

владеть

- навыками объектно-ориентированного программирования на языке Python;
- опытом разработки приложений на языке Python в соответствии с функциональным подходом.

4. Общая трудоёмкость дисциплины и её распределение

количество зачётных единиц – 2,

общая трудоёмкость дисциплины в часах – 72 ч. (в т.ч. аудиторных часов – 10 ч., СРС – 62 ч.),

распределение по семестрам – 5 курс, зима,

форма и место отчётности – зачёт (5 курс, зима).

5. Краткое содержание дисциплины

Объектно-ориентированное программирование на языке Python.

Классы. Экземпляры классов. Атрибуты. Методы. Скрытие данных. Свойства.

Полиморфизм. Перегрузка операций. Наследование. Множественное наследование.

Абстрактные методы. Метаклассы.

Функциональное программирование на языке Python.

Парадигма функционального программирования. Функции в Python, типы аргументов.

Способы вызова функций. Рекурсия. Лямбда-выражения (анонимные функции). Функции

как параметры и как результат. Функции для обработки последовательностей. Списковые

включения. Итераторы. Генераторы. Генераторные выражения.

6. Разработчик

Маркович Ольга Сергеевна, кандидат педагогических наук, доцент кафедры информатики и методики преподавания информатики ФГБОУ ВО «ВГСПУ»,

Усольцев Вадим Леонидович, кандидат физико-математических наук, доцент кафедры информатики и методики преподавания информатики ФГБОУ ВО «ВГСПУ».