

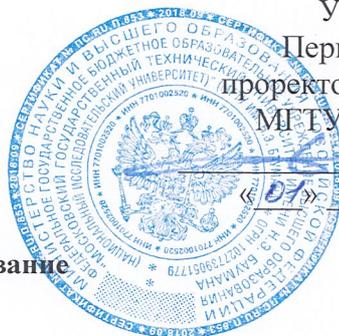
Программа курса «Программирование на Python. Уровень 2. Алгоритмы, работа с данными.»

Номер	Название темы	Количество часов	Описание темы
1	Классы и объекты.	6	Введение в Объектно-ориентированное программирование (ООП) Класс и экземпляр класса. Данные экземпляра, методы экземпляра и свойства экземпляра Создание собственного класса Инкапсуляция Атрибуты класса Чтение и изменение атрибута. Практикум: Разработка собственных классов. Определение нужных методов и свойств классов. Создание нескольких объектов
2	Наследование	5	Роль наследования в ООП, понятие иерархии наследования Принцип утиной типизации Понятие базового класса и производного класса Функция <code>isinstance</code> и ее применение. Создание производного класса Применение экземпляров базового и производного класса. Практикум: Расширение готовых классов
3	Абстрактные классы и полиморфизм	6	Полиморфизм. Принцип DRY и WET Понятие абстракции Знакомство с абстрактными классами Python Подмена методов в производном классе. Полиморфные классы Контейнерные типы. Библиотечные модули <code>collections</code> и <code>collections.abc</code> Применение контейнерных типов Практикум: Имплементация новых методов
4	Алгоритмы сортировки и поиска в Python	6	Сложность алгоритмов, $O(N)$ нотация. Типы сортировки. Основные алгоритмы сортировки и поиска Этапы разработки алгоритма Разработка на Python оптимальных алгоритмов поиска Оптимизация алгоритма Практикум: Реализация на Python алгоритма решателя Судоку. Решение 100 сложнейших Судоку
5	Алгоритмы поиска на графах	5	Введение в теорию графов, основные алгоритмы на графах Теория графов. Представление графов в Python Поиск в ширину Поиск в глубину Лабораторная работа: Имплементация графа на Python Практикум: Нахождение кратчайшего пути проезда на примере графа станций московского метрополитена 6 ак.ч.
6	Записи и данных. Работа с данными SQLite	5	Разработка структуры данных Сохранение данных. Список и запись. Записи и таблицы Чтение и запись Понятие об объектно-реляционном соответствии Хранение данных пользователей в СУБД SQLite Использование СУБД SQLite для хранения данных графа московского метрополитена Практикум: Хранение данных пользователей в СУБД SQLite
7	Хранение данных вне программы	5	Понятие о структуре данных Использование структуры для хранения данных Хранение данных вне Python Сохранение и восстановление данных. Библиотечный модуль <code>pickle</code> и <code>shelve</code> Практикум Сохранение данных графа и использованием модуля <code>pickle</code>

8	Работа с данными JSON в Python	6	Знакомство с форматом JSON. Работа с форматом JSON в Python. Сериализация и десериализация файлов JSON в Python. Хранение данных вне Python Сохранение данных в файл JSON Практикум: Составление графа московского метрополитена на основе файла JSON
9	Итоговая аттестация	1	Итоговая аттестация

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Межотраслевой инжиниринговый центр «Композиты России»



УТВЕРЖДАЮ
Первый проректор –
проректор по учебной работе
МГТУ им. Н.Э. Баумана
Б.В.Падалкин
2020 г.

Дополнительное профессиональное образование

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«Программирование на Python. Уровень 2. Алгоритмы, работа с данными»

Регистрац. № 05.22.20.09.2

Москва, 2020

Оглавление

1.	ОБЩАЯ ХАРАКТЕРИСТИКА ДПП.....	4
1.1.	Цель ДПП.....	4
1.2.	Планируемые результаты обучения.....	4
1.3.	Дополнительные характеристики ДПП.....	4
1.4.	Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения.....	5
1.5.	Соответствие видов деятельности профессиональным компетенциям и их составляющих ...	5
2.	УЧЕБНЫЙ ПЛАН ДПП.....	5
2.1.	Категория слушателей ДПП.....	5
2.2.	Общая трудоёмкость программы, аудиторная и самостоятельная работа.....	6
2.3.	Форма обучения.....	6
2.4.	Учебный план.....	6
3.	КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК.....	6
4.	РАБОЧАЯ ПРОГРАММА ДПП.....	7
5.	УСЛОВИЯ РЕАЛИЗАЦИИ ДПП.....	13
5.1.	Организационные условия реализации ДПП.....	13
5.2.	Педагогические условия реализации ДПП.....	14
5.3.	Учебно-методическое обеспечение ДПП.....	14
5.4.	Методические рекомендации.....	15
6.	ФОРМЫ ИТОГОВОЙ АТТЕСТАЦИИ ДПП.....	15
7.	ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ИТОГОВОЙ АТТЕСТАЦИИ.....	16
7.1.	Паспорт комплекта оценочных средств.....	16
7.2.	Комплект оценочных средств.....	16

1. ОБЩАЯ ХАРАКТЕРИСТИКА ДПП

Программа подготовлена на основе:

- Федерального закона от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
- требований Приказа Минобрнауки России от 01.07.2013 года № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- методических рекомендаций-разъяснений Минобрнауки России по разработке дополнительных профессиональных программ на основе профессиональных стандартов от 22 апреля 2015 года № ВК-1030/06.

Реализация программы ДПП направлена на совершенствование компетенции, необходимой для профессиональной деятельности.

1.1. Цель ДПП

Совершенствование у обучающихся навыков разработки, отладки, проверки работоспособности, а также модификации программного обеспечения.

1.2. Планируемые результаты обучения

Планируемые результаты обучения по ДПП:

- освоение профессиональных компетенций в процессе изучения перечисленных тем в учебном плане;
- успешное освоение программы повышения квалификации;
- успешное прохождение итоговой аттестации (зачет).

Обучающимся, успешно прошедшим обучение, выполнившим текущие контрольные задания и выдержавшим предусмотренное учебным планом зачет, выдается удостоверение о повышении квалификации по ДПП «Программирование на Python. Уровень 2. Алгоритмы, работа с данными».

1.3. Дополнительные характеристики ДПП

Характеристики новой квалификации определены в приказе Минтруда России от 18 ноября 2013 г. № 679н «Об утверждении профессионального стандарта «Программист». Регистрационный номер 4.

Вид профессиональной деятельности:

- разработка программного обеспечения (Код 06.001).

Трудовые функции:

- проектирование программного обеспечения (D/03.6).

1.4. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения

Формируемые компетенции базируются на основании Приказа Минобрнауки России от 23 августа 2017 г. № 808 «Об утверждении федерального государственного образовательного стандарта высшего образования по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии (уровень бакалавриата)».

Перечень компетенций:

ОПК-4 - Способен участвовать в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла.

1.5. Соответствие видов деятельности профессиональным компетенциям и их составляющих

Профессиональные компетенции	Умения	Знания
Проектирование программного обеспечения (D/03.6)		
ОПК-4	Использовать существующие типовые решения и шаблоны проектирования программного обеспечения.	Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения Методы и средства проектирования программного обеспечения Методы и средства проектирования баз данных Методы и средства проектирования программных интерфейсов

2. УЧЕБНЫЙ ПЛАН ДПП

2.1. Категория слушателей ДПП

Имеющаяся квалификация (требования к слушателям) – к освоению ДПП допускаются лица имеющие среднее профессиональное и/или высшее образование.

2.2. Общая трудоёмкость программы, аудиторная и самостоятельная работа

Общая трудоёмкость программы 45 академических часов, из них 39 академических часов аудиторной работы, 5 академических часов самостоятельной работы и 1 академический час итоговой аттестации.

2.3. Форма обучения

Форма обучения по ДПП - очная.

2.4. Учебный план

ДПП «Программирование на Python. Уровень 2. Алгоритмы, работа с данными» реализуется одним модулем.

№ п/п	Наименование темы, модуля	Форма Контроля	Всего, час	В том числе			
				Лекции	Практические занятия	Самостоятельная работа	Итоговая аттестация
1.	Классы и объекты	устный опрос	6	4	1	1	-
2.	Наследование	устный опрос	5	3	1	1	-
3.	Абстрактные классы и полиморфизм	устный опрос	6	4	2	-	-
4.	Алгоритмы сортировки и поиска в Python	устный опрос	6	4	1	1	-
5.	Алгоритмы поиска на графах	устный опрос	5	4	1	-	-
6.	Записи и данных. Работа с данными SQLite	устный опрос	5	3	1	1	-
7.	Хранение данных вне программы	устный опрос	5	4	1	-	-
8.	Работа с данными JSON в Python	устный опрос	6	4	1	1	-
9.	Итоговая аттестация	Зачет	1	-	-	-	1
	ИТОГО	-	45	30	9	5	1

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование темы	1 день	2 день	3 день	4 день	5 день	6 день
1.	Классы и объекты						
2.	Наследование						
3.	Абстрактные классы и полиморфизм						
4.	Алгоритмы сортировки и поиска в Python						

5.	Алгоритмы поиска на графах						
6.	Записи и данных. Работа с данными SQLite						
7.	Хранение данных вне программы						
8.	Работа с данными JSON в Python						
9.	Итоговая аттестация						

Минимальный срок освоения ДПП — 6 дней.

4. РАБОЧАЯ ПРОГРАММА ДПП

4.1. Рабочая программа модуля «Программирование на языке Python. Уровень 2. Продвинутый курс»

4.1.1. Цель изучения модуля: совершенствование у обучающихся навыков разработки, отладки, проверки работоспособности, модификации программного обеспечения

4.1.2. Задача изучения модуля:

- создавать собственные классы;
- оценивать сложность алгоритма;
- использовать важнейшие стандартные структуры данных;
- создавать собственные структуры данных на основе стандартных.

4.1.3. Планируемые результаты обучения

Процесс изучения раздела направлен на формирование следующих компетенций

Код компетенции	Перечень планируемых результатов обучения по модулю	Формы и методы обучения, способствующие формированию и развитию компетенции
ОПК-4	<p>Знать: Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения</p> <p>Уметь: Использовать методы и средства проектирования программных интерфейсов</p>	<p>Формы обучения: Фронтальная. Методы обучения: Лекция; Практические Работы; Самостоятельная работа.</p>

4.1.4 Содержание курса

Тема 1. Классы и объекты (6 часов).

Лекция (4 часа). Введение в Объектно-ориентированное программирование (ООП).

Класс и экземпляр класса. Данные экземпляра, методы экземпляра и свойства экземпляра.

Создание собственного класса. Инкапсуляция. Атрибуты класса.

Практические занятия (1 час). Разработка собственных классов. Определение нужных методов и свойств классов. Создание нескольких объектов.

Самостоятельная работа (1 час). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Классы и объекты	Чтение и изменение атрибута.	Проработка дополнительной литературы	Фарафонов, А. С. Программирование на языке высокого уровня: методические указания к проведению лабораторных работ по курсу «Программирование» / А. С. Фарафонов. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2013. — 32 с. — ISBN 2227-8397.	устный опрос

Тема 2. Наследование (5 часов).

Лекция (3 часа). Роль наследования в ООП, понятие иерархии наследования. Принцип утиной типизации. Понятие базового класса и производного класса. Функция `isinstance` и ее применение. Создание производного класса

Практические занятия (1 час). Расширение готовых классов.

Самостоятельная работа (1 час). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
-------------------	---	------------------------------	---------------------------------	----------------

Наследование	Применение экземпляров базового и производного класса.	Проработка дополнительной литературы	Фарафонов, А. С. Программирование на языке высокого уровня: методические указания к проведению лабораторных работ по курсу «Программирование» / А. С. Фарафонов. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2013. — 32 с. — ISBN 2227-8397.	устный опрос
--------------	--	--------------------------------------	--	--------------

Тема 3. Абстрактные классы и полиморфизм (6 часов)

Лекция (4 часа). Полиморфизм. Принцип DRY и WET. Понятие абстракции. Знакомство с абстрактными классами Python. Подмена методов в производном классе. Полиморфные классы. Контейнерные типы. Библиотечные модули collections и collections.abc. Применение контейнерных типов.

Практические занятия (2 часа). Имплементация новых методов.

Тема 4. Алгоритмы сортировки и поиска в Python (6 часов)

Лекция (4 часа). Сложность алгоритмов, O(N) нотация. Типы сортировки. Основные алгоритмы сортировки и поиска. Этапы разработки алгоритма. Разработка на Python оптимальных алгоритмов поиска.

Практические занятия (1 час). Реализация на Python алгоритма решателя Судоку. Решение 100 сложнейших Судоку.

Самостоятельная работа (1 час). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Алгоритмы сортировки и поиска в Python	Оптимизация алгоритма	Проработка дополнительной литературы	Фризен, И. Г. Офисное программирование : учебное пособие /	устный опрос

			И. Г. Фризен. — Ростов-на-Дону : Феникс, 2010. — 239 с. — ISBN 978-5-222-16500-3.	
--	--	--	---	--

Тема 5. Алгоритмы поиска на графах (5 часов)

Лекция (4 часа). Введение в теорию графов, основные алгоритмы на графах. Теория графов. Представление графов в Python. Поиск в ширину. Поиск в глубину.

Практические занятия (1 час). Имплементация графа на Python. Нахождение кратчайшего пути проезда на примере графа станций московского метрополитена.

Тема 6. Записи и данных. Работа с данными SQLite (5 часов)

Лекция (3 часа). Разработка структуры данных. Сохранение данных. Список и запись. Записи и таблицы. Чтение и запись. Понятие об объектно-реляционном соответствии.

Практические занятия (1 час). Хранение данных пользователей в СУБД Sqlite

Самостоятельная работа (1 час). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Записи и данных. Работа с данными SQLite	Использование СУБД Sqlite для хранения данных графа московского метрополитена	Проработка дополнительной литературы	Фризен, И. Г. Офисное программирование : учебное пособие / И. Г. Фризен. — Ростов-на-Дону : Феникс, 2010. — 239 с. — ISBN 978-5-222-16500-3.	устный опрос

Тема 7. Хранение данных вне программы (5 часов)

Лекция (4 часа). Понятие о структуре данных. Использование структуры для хранения данных. Хранение данных вне Python. Сохранение и восстановление данных. Библиотечный модуль pickle и shelve.

Практические занятия (1 час). Сохранение данных графа и использованием модуля pickle.

Тема 8. Работа с данными JSON в Python (6 часов)

Лекция (4 часа). Знакомство с форматом JSON. Работа с форматом JSON в Python. Сериализация и десериализация файлов JSON в Python. Хранение данных вне Python
 Практические занятия (1 час). Составление графа московского метрополитена на основе файла JSON
 Самостоятельная работа (1 час). Проработка материала дополнительной литературы по теме.

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Работа с данными JSON в Python	Сохранение данных в файл JSON	Проработка дополнительной литературы	Фризен, И. Г. Офисное программирование : учебное пособие / И. Г. Фризен. — Ростов-на-Дону : Феникс, 2010. — 239 с. — ISBN 978-5-222-16500-3.	устный опрос

4.1.5. Оценочное средство для текущего контроля (примерные вопросы для устного опроса):

Тема 1. Классы и объекты.

1. Как производится отладка программы на Python? Можно ли выполнить Python-код пошагово?
2. Рассмотрите два подхода ниже для инициализации массива и массивов. В чём разница между этими подходами и почему вам следует использовать только один из них?

```
# Инициализация массива -- метод 1
...
x = [[1,2,3,4]] * 3
x
[[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]

# Инициализация массива -- метод 2
...
y = [[1,2,3,4] for _ in range(3)]
y
[[1, 2, 3, 4], [1, 2, 3, 4], [1, 2, 3, 4]]
```

Тема 2. Наследование

2. Что такое NumPy? Почему лучше использовать массивы NumPy вместо списков Python?
2. Как можно создать массив NumPy в Python?
3. Зачем использовать декораторы функций?

Тема 3. Абстрактные классы и полиморфизм

1. Что такое `@classmethod`, `@staticmethod`, `@property`?
2. Создайте логируемый декоратор.

Тема 4. Алгоритмы сортировки и поиска в Python

Посмотрите на небольшой декоратор, который можно использовать для записи названия любой функции и того, что она делает:

```
import logging

def log(func):
    """
    Логируем какая функция вызывается
    """

    def wrap_log(*args, **kwargs):
        name = func.__name__
        logger = logging.getLogger(name)
        logger.setLevel(logging.INFO)

        # Открываем файл логов для записи
        fh = logging.FileHandler("%s.log" % name)
        fmt = '%(asctime)s - %(name)s - %(levelname)s - %(message)s'
        formatter = logging.Formatter(fmt)
        fh.setFormatter(formatter)
        logger.addHandler(fh)

        logger.info("Вызов функции: %s" % name)
        result = func(*args, **kwargs)
        logger.info("Результат: %s" % result)
        return func

    return wrap_log

@log
def double_function(a):
    """
    Умножаем полученный параметр
    """
    return a*2

if __name__ == "__main__":
    value = double_function(2)
```

Тема 5. Алгоритмы поиска на графах

1. Введение в теорию графов, основные алгоритмы на графах
2. Теория графов. Представление графов в Python
3. Поиск в ширину
4. Поиск в глубину

Тема 6. Записи и данных. Работа с данными SQLite

1. Разработка структуры данных
2. Сохранение данных.
3. Список и запись. Записи и таблицы
4. Чтение и запись Понятие об объектно-реляционном соответствии

Тема 7. Хранение данных вне программы

1. Понятие о структуре данных
2. Использование структуры для хранения данных
3. Хранение данных вне Python

Тема 8. Работа с данными JSON в Python

1. Знакомство с форматом JSON. Работа с форматом JSON в Python.
2. Сериализация и десериализация файлов JSON в Python.
3. Хранение данных вне Python
4. Сохранение данных в файл JSON

5. УСЛОВИЯ РЕАЛИЗАЦИИ ДПП

5.1. Организационные условия реализации ДПП

Наименование аудитории	Вид занятия	Наименование оборудования, программного обеспечения
Компьютерный класс	Лекции	Персональные компьютеры с доступом к сети Интернет, мультимедийный проектор, экран, доска, маркер, лазерная указка, ПО Python версии не ниже 3.6
Компьютерный класс	Практические занятия	Персональные компьютеры с доступом к сети Интернет, мультимедийный проектор, экран, доска, маркер, лазерная указка, ПО Python версии не ниже 3.6
Компьютерный класс	Самостоятельная работа	Персональные компьютеры с доступом к сети Интернет, мультимедийный проектор, экран, доска, маркер, лазерная указка, ПО Python версии не ниже 3.6
Компьютерный класс	Итоговая аттестация	Персональные компьютеры с доступом к сети Интернет, мультимедийный проектор, экран, доска, маркер, лазерная указка, ПО Python версии не ниже 3.6

5.2. Педагогические условия реализации ДПП

Реализация программы обеспечивается преподавательским составом, удовлетворяющим следующим условиям:

- наличие высшего профессионального образования, соответствующее профилю программы, из числа штатных преподавателей, или привлеченных на условиях почасовой оплаты труда;
- значительный опыт практической деятельности в соответствующей сфере из числа штатных преподавателей или привлеченных на условиях почасовой оплаты труда

5.3. Учебно-методическое обеспечение ДПП

Основная литература:

1. Ковалевская, Е. В. Методы программирования : учебное пособие / Е. В. Ковалевская, Н. В. Комлева. — Москва : Евразийский открытый институт, 2011. — 320 с. — ISBN 978-5-374-00356-7.
2. Шелудько, В. М. Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули : учебное пособие / В. М. Шелудько. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2017. — 107 с. — ISBN 978-5-9275-2648-2.
3. Сузи, Р. А. Язык программирования Python : учебное пособие / Р. А. Сузи. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 350 с. — ISBN 978-5-4497-0705-5.

Дополнительная литература:

1. Визуальное программирование на основе библиотеки MFC : методические указания к лабораторным работам по курсу «Визуальное программирование» для студентов направления 09.03.02 Информационные системы и технологии / составители А. Я. Лахов, Р. Е. Борщиков. — Саратов : Вузовское образование, 2016. — 57 с. — ISBN 2227-8397.
2. Фарафонов, А. С. Программирование на языке высокого уровня : методические указания к проведению лабораторных работ по курсу «Программирование» / А. С. Фарафонов. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2013. — 32 с. — ISBN 2227-8397.
3. Фризен, И. Г. Офисное программирование : учебное пособие / И. Г. Фризен. — Ростов-на-Дону : Феникс, 2010. — 239 с. — ISBN 978-5-222-16500-3.

5.4. Методические рекомендации

ДПП построена по тематическому принципу, каждый раздел представляет собой логически завершённый материал.

Преподавание программы основано на личностно-ориентированной технологии образования, сочетающей два равноправных аспекта этого процесса: обучение и учение. Личностно-ориентированный подход развивается при участии слушателей в активной работе на практических занятиях. Личностно-ориентированный подход направлен, в первую очередь, на развитие индивидуальных способностей обучающихся, создание условий для развития творческой активности слушателя и разработке инновационных идей, а также на развитие самостоятельности мышления при решении учебных задач разными способами, нахождение рационального варианта решения, сравнения и оценки нескольких вариантов их решения и т.п. Это способствует формированию приемов умственной деятельности по восприятию новой информации, ее запоминанию и осознанию, созданию образов для сложных понятий и процессов, приобретению навыков поиска решений в условиях неопределенности.

Практические занятия проводятся для приобретения навыков решения практических задач в предметной области модуля. Задания, выполняемые на практических занятиях, выполняются с использованием активных и интерактивных методов обучения.

Самостоятельная работа слушателей предназначена для проработки дополнительной литературы. Результаты практических заданий слушателей учитываются на итоговой аттестации.

При изучении курса предусмотрены следующие методы организации и осуществления учебно-познавательной деятельности:

- объяснительно-иллюстративный метод;
- репродуктивный метод;
- частично-поисковый метод.

6. ФОРМЫ ИТОГОВОЙ АТТЕСТАЦИИ ДПП

Итоговая аттестация проводится в форме зачета. Зачет проводится в присутствии преподавателя. Результатом зачета служат правильные ответы на вопросы билета, состоящего из 5 (пяти) вопросов.

По результатам итоговой аттестации слушателю выставляется оценка «ЗАЧТЕНО/НЕ ЗАЧТЕНО»:

Оценка «ЗАЧТЕНО» выставляется слушателю, который:

- правильно ответил не менее чем на 60% вопросов в билете;
- продемонстрировал необходимые систематизированные знания и достаточную степень владения принципами предметной области программы, понимание их особенностей и взаимосвязь между ними в течение всего срока обучения по ДПП.

Оценка «НЕ ЗАЧТЕНО» ставятся слушателю, который:

- ответил правильно менее чем на 60% полученных вопросов в билете;
- имеет крайне слабые теоретические и практические знания, обнаруживает неспособность к построению самостоятельных заключений..

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ИТОГОВОЙ АТТЕСТАЦИИ

7.1. Паспорт комплекта оценочных средств

Предметы оценивания	Объекты оценивания	Показатели оценки
ОПК-4. Способен участвовать в разработке технической документации программных продуктов и комплексов с использованием стандартов, норм и правил, а также в управлении проектами создания информационных систем на стадиях жизненного цикла.	Тестовое задание	Количество правильных ответов

7.2. Комплект оценочных средств

7.2.1. Темы для подготовки к зачету:

1. Генератор-функция
2. Генератор списков, генератор словарей, генератор множеств
3. Создание модуля
4. Импорт пакета
5. Создание собственного пакета
6. Важнейшие стандартные пакеты
7. Подсистема `pip`
8. Установка стороннего модуля
9. Работа с файлами и каталогами
10. Основные операции с файлами.
11. Основные операции с путями к файлам.
12. Рекурсивный обход каталога.
13. Обработка параметров командной строки
14. Чтение файла

15. Запись в файл
16. Понятие об исключении
17. Выброс исключения
18. Перехват исключения
19. Стандартные исключения
20. Понятие о регулярном выражении
21. Синтаксис регулярных выражений
22. Применение регулярных выражений
23. Каррирование
24. Замыкание
25. Понятие о декораторе

7.2.2. Примерные варианты билетов для проведения зачета:

Билет №1

1. Принцип динамической типизации.
2. Основные операции с кортежем.
3. Приведите пример кода на использование перебора (for).
4. Приведите пример кода с использованием функции filter.
5. Словарь. Основные операции со словарем.

Билет №2

1. Именованные и неименованные аргументы функций.
2. Создание функции (привести пример кода).
3. Операторы break и continue.
4. Приведите пример кода на использование перебора (for).
5. Понятие коллекции.

Билет №3

1. Множества. Основные операции с множеством.
2. Именованные и неименованные аргументы функций.
3. Приведите пример кода на использование лямбда-функции.
4. Приведите пример кода на использование перебора (for).
5. Типы данных: числа, строки, списки, логический тип, None