

**Список тем к устной части зачета по курсу**  
**«Программирование на языке Python»**  
**Осенний семестр. Учебный год 2024-2025.**

Основы языка:

1. Структура языка. Физические и логические строки, простые и составные инструкции. Токены: литералы, идентификаторы, разделители, операторы, ключевые слова.
2. Объекты Python и их основные характеристики: идентичность, тип данных и значение. Встроенная функция `id()` и оператор `is`.
3. Понятия изменяемых и неизменяемых типов данных. Оператор связывания (`=`). Переменные в Python и ссылочная модель памяти.
4. Булев тип данных. Операции над объектами булева типа данных. Синглтоны `True`, `False` и `None`.
5. Оператор `not`. Ленивые операторы `or` и `and`.
6. Встроенные функции `all()` и `any()`. Примеры использования. Ленивость вычисления.
7. Управляющие конструкции: различные формы ветвления и цикла `while`. Ключевые слова `break` и `continue`. Примеры использования.
8. Числовые типы данных. Основные операции, допустимые значения, численные литералы.
9. Кортежи и операции над ними. Обход ограничения на неизменяемость элементов. Основные методы.
10. Списки и операции над ними. Основные методы. Ошибки при построении списков через повторение с пояснением сути ошибки.
11. Цикл `for` и его различные формы. Инструкции `break`, `continue` и `else` в контексте цикла `for`.

12. Строки, виды строк. Основные операции над строками: манипуляция с регистром, проверка наличия символов из определенной группы, проверка регистра, очистка строки, разбиение и объединение строк, работа с подстроками.
13. Словари. Понятие хешируемости и ограничения на ключи словаря. Упорядоченность словарей. Основные операции над словарями и методы словарей.
14. Исключения. Механизм распространения исключений. try-except-else-finally блок. Ключевое слово raise, конструкция raise from. Мотивация и примеры использования.
15. Модули. Мотивация использования. Доступ к содержимому модуля. Конструкции import, from ... import ..., from ... import as ..., from ... import \*. Порядок поиска модулей.
16. Исполняемые модули. Конструкция if \_\_name\_\_ == «\_\_main\_\_». Примеры и мотивация использования. Пакеты. Мотивация использования.

## Функции

1. Понятие функции в программировании и мотивация использования.
2. Некоторые встроенные функции: min(), max(), sum(), sorted(), print(), input(). Примеры использования.
3. Пользовательские функции: определение пользовательских функций. Символические таблицы. Инструкция return и возвращаемое значение.
4. Передача аргументов в функции. Изменение объектов изменяемых типов данных в теле функции.
5. Параметры по умолчанию. Время вычисления параметров по умолчанию. Ошибка использования изменяемых объектов в качестве параметров по умолчанию. Правильные подходы для

использования изменяемых объектов в качестве параметров по умолчанию.

6. Именованные и позиционные аргументы. Строго именованные и строго позиционные аргументы. Мотивация использования.
7. Синтаксис `*args`, `**kwargs`. Мотивация и примеры использования.
8. Метаинформация о функциях: `__name__`, `docstring` и аннотации типов. Мотивация использования.
9. Функции, как объекты высшего порядка: создание функций во время исполнений, присвоение функции переменным, использование функций в качестве аргументов и возвращаемых значений других функций. Примеры.
10. Ключевые слова `global` и `nonlocal`. Замыкания. Мотивация и примеры использования.
11. Декораторы. Мотивация и примеры использования. Декоратор `wraps`. Мотивация использования. Параметрические декораторы. Примеры и мотивация использования.
12. Анонимные функции. Мотивация и примеры использования. Примеры замыкания анонимных функций на локальный контекст.

#### ООП:

1. Классы. Объект класса. Определение пользовательского класса. Объект класса как объект высшего порядка. Атрибуты класса.
2. Экземпляры классов. Методы экземпляра. Специальный метод `__init__`. Смысл аргумента `self`. Статические методы.
3. Атрибуты классов и атрибуты экземпляров. Порядок поиска атрибутов. Динамичное добавление атрибутов.
4. Модуль стандартной библиотеки `dataclasses`. Декоратор `dataclass` и мотивация использования.

5. Инкапсуляция в Python. Реализация геттеров и сеттеров с помощью декоратора `property`.
6. Наследование в Python. Множественное наследование в Python. Суть проблемы ромба и MRO, как решение этой проблемы. Специальный объект `super()`.
7. Полиморфизм в Python.
8. Абстракция в Python. Модуль стандартной библиотеки `abc`. Примеры и мотивация использования.
9. Протоколы. Протоколы, как пример структурного подтипирования. Протокол контекстного менеджера.
10. Эмуляция числовых типов данных за счет определения некоторых специальных арифметических и логических методов.
11. Бинарные арифметические операции. Объект `NotImplemented`. Определение прямых и отраженных арифметических операций. Порядок выполнения бинарных арифметических операций интерпретатором.
12. Определение методов для преобразования объекта в числовые типы данных.
13. Протоколы контейнера и объекта с размером (`Container` и `Sized`). Ограничения метода `__len__`. Определение метода `__bool__` через `__len__`.
14. Протокол итерируемого объекта и итератора. Встроенные функции `next` и `iter`. Исключение `StopIteration`. Примеры реализации итерируемого объекта и итератора. Определение оператора `in` с помощью итерируемости.
15. Протокол контейнера. Определение операций чтения, перезаписи и удаления элементов контейнера. Определение итерируемости через метод `__getitem__`.

16. Генераторные функции и генераторы. Мотивация и примеры использования. Ключевое слово `yield`. Разница между итерируемым объектом, итератором и генератором. Генераторные выражения и включения.
17. Встроенные объект, производящие генераторы: `range`, `enumerate`, `zip`, `map`. Примеры использования.