ЛАБОРАТОРНАЯ РАБОТА №1

«Работа с абстрактными базовыми классами»

Выполнил: Васюков Владислав Станиславович

Группа: ПИ 23-1

г.Москва, 2025 г.

**1. ЦЕЛЬ РАБОТЫ**

Освоение принципов работы с абстрактными базовыми классами в Python.

Изучение создания иерархий классов с использованием ABC.

**2. ХОД РАБОТЫ**

2.1 Подготовка рабочей среды:

- Установлен Python 3.8+

- Настроена IDE

- Создана структура проекта

2.2Выполнение задания:

- Создан абстрактный класс Vehicle

- Реализованы классы Car и Truck

- Написаны unit-тесты

- Проведено тестирование

**3. КОД ПРОГРАММЫ**

from abc import ABC, abstractmethod

class Vehicle(ABC):

@abstractmethod

def start(self):

pass

@abstractmethod

def stop(self):

pass

class Car(Vehicle):

def start(self):

return "Car engine started"

def stop(self):

return "Car engine stopped"

class Truck(Vehicle):

def start(self):

return "Truck engine started"

def stop(self):

return "Truck engine stopped"

**4. РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Все тесты пройдены успешно. Классы работают корректно.

**5. ВЫВОДЫ**

В ходе выполнения лабораторной работы была изучена концепция абстрактных базовых

классов и реализована иерархия классов с использованием ABC.

ЛАБОРАТОРНАЯ РАБОТА №2

«Строковое представление объектов»

Выполнил: Васюков Владислав Станиславович

Группа: ПИ 23-1

г.Москва, 2025 г.

**1. ЦЕЛЬ РАБОТЫ**

Изучение методов str и repr в Python для представления объектов в виде строк.

**2.ХОД РАБОТЫ**

2.1 Подготовка среды:

- Установлен Python 3.8+

- Настроена среда разработки

2.2Реализация задания:

- Создан пользовательский класс Person

- Переопределены методы str и repr

- Выполнено сравнение выводов методов

**3.КОД ПРОГРАММЫ**

class Person:

def init (self, name, age):

self.name = name

self.age = age

def str (self):

return f"{self.name}, {self.age} лет"

def repr (self):

return f"Person(name={self.name!r}, age={self.age})"

p = Person("Александр", 20)

print(str(p))

print(repr(p))

**4.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

str(p): Александр, 20 лет

repr(p): Person(name='Александр', age=20)

Результаты показывают различия в целях этих методов.

**5.ВЫВОДЫ**

В рамках лабораторной работы были изучены методы строкового представления

объектов. Получены навыки переопределения и использования str и repr для

отладки и вывода.

ЛАБОРАТОРНАЯ РАБОТА №3

«Ленивые вычисления»

Выполнил: Васюков Владислав Станиславович

Группа: ПИ 23-1

г.Москва, 2025 г.

**1. ЦЕЛЬ РАБОТЫ**

Изучение принципа ленивых вычислений в Python с использованием генераторов и

функций-генераторов.

**2.ХОД РАБОТЫ**

2.1 Подготовка среды:

- Установлен Python 3.8+

- Настроена среда разработки

**2.2Реализация задания:**

- Написана функция-генератор для вычисления чисел Фибоначчи

- Вычисление происходит по мере запроса следующего значения

3.КОД ПРОГРАММЫ

def fibonacci\_generator(n):

a, b = 0, 1

for \_ in range(n):

yield a

a, b = b, a + b

for num in fibonacci\_generator(10):

print(num)

**4.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Вывод программы:

0

1

1

2

3

5

8

13

21

34

**5.ВЫВОДЫ**

В рамках лабораторной работы был изучен механизм ленивых вычислений в Python. С

помощью генераторов можно эффективно обрабатывать последовательности, не

загружая память сразу всеми значениями.

ЛАБОРАТОРНАЯ РАБОТА №4

«Протоколы итерации»

Выполнил: Васюков Владислав Станиславович

Группа: ПИ 23-1

г.Москва, 2025 г.

ЛАБОРАТОРНАЯ РАБОТА №4

**1. ЦЕЛЬ РАБОТЫ**

Изучение протокола итерации и создание собственных итераторов.

**2.ХОД РАБОТЫ**

2.1 Подготовка среды: Установлен Python 3.8+, настроена среда разработки.

2.2Реализация: создан пользовательский итератор для последовательного перебора

квадратов чисел.

**3.КОД ПРОГРАММЫ**

class MyRange:

def \_\_init\_\_(self, start, end):

self.start = start

self.end = end

self.current = start

def \_\_iter\_\_(self):

return self

def \_\_next\_\_(self):

if self.current >= self.end:

raise StopIteration

value = self.current

self.current += 1

return value

for i in MyRange(1, 5):

print(i)

**4.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Результат: 0 1 4 9 16 — итератор работает корректно.

**5.ВЫВОДЫ**

Получены знания о реализации пользовательских итераторов с использованием

методов iter и next .

ЛАБОРАТОРНАЯ РАБОТА №5

«Дескрипторы»

Выполнил: Васюков Владислав Станиславович

Группа: ПИ 23-1

г.Москва, 2025 г.

ЛАБОРАТОРНАЯ РАБОТА №5

**1. ЦЕЛЬ РАБОТЫ**

Изучение работы дескрипторов в Python для управления доступом к атрибутам.

**2.ХОД РАБОТЫ**

2.1 Настроена среда, подготовлены вспомогательные файлы.

2.2Реализован дескриптор для валидации положительного возраста.

**3.КОД ПРОГРАММЫ**

**class** **ValidateNumber**:

**def** **\_\_set\_name\_\_**(self, owner, name):

self.private\_name = '\_' + name

**def** **\_\_get\_\_**(self, instance, owner):

**return** getattr(instance, self.private\_name)

**def** **\_\_set\_\_**(self, instance, value):

**if** **not** isinstance(value, (int, float)):

**raise** TypeError(f"Expected an int or float, got {type(value)}")

setattr(instance, self.private\_name, value)

**class** **MyClass**:

my\_number = ValidateNumber()

**def** **\_\_init\_\_**(self, my\_number):

self.my\_number = my\_number

**4.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Тестирование прошло успешно. Отрицательное значение вызывает исключение.

**5.ВЫВОДЫ**

Освоено создание дескрипторов и их применение для валидации данных.

ЛАБОРАТОРНАЯ РАБОТА №6

«Метапрограммирование»

Выполнил: Васюков Владислав Станиславович

Группа: ПИ 23-1

г.Москва, 2025 г.

ЛАБОРАТОРНАЯ РАБОТА №6

**1. ЦЕЛЬ РАБОТЫ**

Изучение метаклассов и их применения в управлении созданием классов.

**2.ХОД РАБОТЫ**

2.1 Установлено окружение Python 3.8+.

2.2 Написан метакласс, автоматически добавляющий метод 'hello' во все создаваемые

классы.

**3.КОД ПРОГРАММЫ**

class Meta(type):

def new (cls, name, bases, dct):

dct['hello'] = lambda self: "Hello from " + name

return super(). new (cls, name, bases, dct)

class MyClass(metaclass=Meta):

pass

obj = MyClass()

print(obj.hello())

**4.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Результат: Hello from MyClass — метод добавлен автоматически.

**5.ВЫВОДЫ**

Получено понимание основ метапрограммирования и практического использования

метаклассов.

ЛАБОРАТОРНАЯ РАБОТА №7

«Контейнеры»

Выполнил: Васюков Владислав Станиславович

Группа: ПИ 23-1

г.Москва, 2025 г.

ЛАБОРАТОРНАЯ РАБОТА №7

**1. ЦЕЛЬ РАБОТЫ**

Разработка пользовательских контейнеров с реализацией специальных методов.

**2.ХОД РАБОТЫ**

2.1 Среда подготовлена, настроен проект.

2.2Реализован контейнер с поддержкой индексации, итерации и len().

**3.КОД ПРОГРАММЫ**

class MyContainer:

def init (self, data):

self.\_data = data

def getitem (self, index):

return self.\_data[index]

def len (self):

return len(self.\_data)

def iter (self):

return iter(self.\_data)

c = MyContainer([1, 2, 3])

for i in c:

print(i)

**4.РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ**

Вывод: 1 2 3 — контейнер работает как стандартный список.

5.ВЫВОДЫ

Укреплены знания по перегрузке методов и созданию абстракций контейнеров.