



**Московский политехнический университет**

**Факультет Информационных технологий**

**Кафедра Информатики и информационных технологий**

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

## **Лабораторная работа № 3**

**Дисциплина: Backend-разработка**

**Тема: Функции и лямбда-выражения**

**Выполнил: студент группы 231-333**

**Фабрикант Тимур Романович**

(Фамилия И.О.)

**Дата, подпись** \_\_\_\_\_  
(Дата) (Подпись)

**Проверил: Полубояринова А.С.** \_\_\_\_\_  
(Фамилия И.О., степень, звание) (Оценка)

**Замечание:** \_\_\_\_\_

**Москва**

**2025**

**Цель:** Получение навыков работы с функциями, лямбда-выражениями и замыканиями

**Задачи:**

Написать следующие конструкции:

- функция без параметров
- функция с параметрами
- функция с несколькими параметрами со значениями по умолчанию
- функция с несколькими параметрами, у которых задан тип
- функция с неопределённым количеством параметров (args)
- функция с неопределённым количеством параметров (kwargs)
- функция, вызывающая внутри себя другую функцию
- функция, принимающая функцию как параметр (минимум 3 примера)
- функция с объявленной внутри локальной функцией (минимум 2 примера)
- лямбда-выражение без параметров
- лямбда-выражение с параметрами
- функция, принимающая лямбда-выражение как параметр, и вызывающая лямбда-выражение внутри себя
- функция с замыканиями (минимум 3 примера)

Ссылка на код:

<https://github.com/DurkaVerder/Backend-python-Polytech/tree/main/Lab3>

## Ход выполнения работы:

### Написание функций в файле funcs.py

Листинг 1. funcs.py

```
from typing import Callable, List, Dict

def welcome_message() -> str:
    return "Welcome to the system!"

def greet_user(name: str) -> str:
    return f"Hello, {name}!"

def create_user(username="Guest", role="viewer") -> Dict[str, str]:
    return {"username": username, "role": role}

def send_message(
    sender: str, recipient: str, message: str, urgent: bool = False
) -> str:
    status = "URGENT" if urgent else "normal"
    return f"[{status}] From {sender} to {recipient}: {message}"

def list_users(*users: str) -> List[str]:
    return [user for user in users]

def user_info(**info) -> str:
    return ", ".join(f"{k}: {v}" for k, v in info.items())

def announce_new_user(username: str) -> str:
    def message():
        return f"New user registered: {username}"

    return message()

def execute_function(func: Callable[[], str]) -> str:
    return func()

def repeat_function(func: Callable[[], str], times: int) -> List[str]:
    return [func() for _ in range(times)]

def call_with_username(func: Callable[[str], str], username: str) -> str:
    return func(username)
```

```
def log_user_action(username: str, action: str) -> str:
    def log_entry():
        return f"{username} performed action: {action}"

    return log_entry()

def notify_admin(user: str, message: str) -> str:
    def notification():
        return f"Notify admin: {user} -> {message}"

    return notification()

say_hello_lambda = lambda: "Hello from lambda!"

greet_lambda = lambda name: f"Hello, {name} from lambda!"

def call_lambda(func: Callable[[], str]) -> str:
    return func()

def make_greeter(name: str) -> Callable[[], str]:
    def greeter():
        return f"Hi, {name}!"

    return greeter

def make_message_logger(prefix: str) -> Callable[[str], str]:
    def logger(message: str) -> str:
        return f"[{prefix}] {message}"

    return logger

def make_user_counter() -> Callable[[], int]:
    count = 0

    def counter() -> int:
        nonlocal count
        count += 1
        return count

    return counter
```

## Написание main.py для вызова всех функций

Листинг 2. main.py

```
from funcs import *

if __name__ == "__main__":
    print(welcome_message())
    print(greet_user("Timur"))

    user = create_user()
    print(user)

    message = send_message("Alex", "Nikita", "Hi there!", urgent=True)
    print(message)

    users_list = list_users("Alex", "Nikita", "Pasha")
    print(users_list)

    info = user_info(username="Alex", age=20, role="admin")
    print(info)

    print(announce_new_user("Pasha"))

    print(execute_function(lambda: "Something"))
    print(repeat_function(lambda: "Repeat me", 3))
    print(call_with_username(greet_user, "DurkaVerder"))

    print(log_user_action("Alex", "login"))
    print(notify_admin("Pasha", "Server down"))

    print(say_hello_lambda())
    print(greet_lambda("Pasha"))
    print(call_lambda(lambda: "Calling from call_lambda"))

    greeter = make_greeter("Nikita")
    print(greeter())

    logger = make_message_logger("INFO")
    print(logger("System started"))

    counter = make_user_counter()
    print(counter())
    print(counter())
    print(counter())
```

## Вывод программы:

```
vrrrr@DurkaVerder MINGW64 ~/OneDrive/Рабочий стол/Учёба/5 семестр/Бэк/Backend-python-Polytech/Lab3 (main)
$ python main.py
Welcome to the system!
Hello, Timur!
{'username': 'Guest', 'role': 'viewer'}
[URGENT] From Alex to Nikita: Hi there!
['Alex', 'Nikita', 'Pasha']
username: Alex, age: 20, role: admin
New user registered: Pasha
Something
['Repeat me', 'Repeat me', 'Repeat me']
Hello, DurkaVerder!
Alex performed action: login
Notify admin: Pasha -> Server down
Hello from lambda!
Hello, Pasha from lambda!
Calling from call_lambda
Hi, Nikita!
[INFO] System started
1
2
3
```