

Защищено:
Гапанюк Ю.Е.

"__" 2025 г.

Демонстрация:
Гапанюк Ю.Е.

"__" 2025 г.

**Отчет по ДЗ по курсу
Парадигмы и конструкции языков программирования
ГУИМЦ**

*Тема работы: "изучение ранее неизвестного языка
программирования."*

3
(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-54Б

Щетинин Д.С.

(подпись)

"__" 2025 г.

1. Тема и задание для выполнения лабораторной работы.

Цель домашнего задания: изучение ранее неизвестного языка программирования.

Задание:

1. Выберите язык программирования (который Вы ранее не изучали) и (1) напишите по нему реферат с примерами кода или (2) реализуйте на нем небольшой проект (с детальным текстовым описанием).
2. Реферат (проект) может быть посвящен отдельному аспекту (аспектам) языка или содержать решение какой-либо задачи на этом языке.
3. Необходимо установить на свой компьютер компилятор (интерпретатор, транспилятор) этого языка и произвольную среду разработки.
4. В случае написания реферата необходимо разработать и откомпилировать примеры кода (или модифицировать стандартные примеры).
5. В случае создания проекта необходимо детально комментировать код.
6. При написании реферата (создании проекта) необходимо изучить и корректно использовать особенности парадигмы языка и основных конструкций данного языка.
7. Приветствуется написание черновика статьи по результатам выполнения ДЗ. Черновик статьи может быть подготовлен группой студентов, которые исследовали один и тот же аспект в нескольких языках или решили одинаковую задачу на нескольких языках.

2. Листинг программы

Файл: di.cpp.py

```
# Интерфейс (абстрактный класс)
from abc import ABC, abstractmethod
#рефакторинг - улучшение переводится. С++ и Си Шарп для себя , питон не очень для развития
class Database(ABC):
    @abstractmethod
    def save(self, data: str) -> None:
        pass

# Реализация зависимости
class PostgreSQL(Database):
    def save(self, data: str) -> None:
        print(f"Сохранение '{data}' в PostgreSQL")

# Класс, принимающий зависимость через конструктор
class Service:
    def __init__(self, database: Database):
        self.database = database

    def process_data(self, data: str) -> None:
        self.database.save(data)

# Использование
db = PostgreSQL()
service = Service(db)
service.process_data("Привет, DI на Python!")
```

3. Результаты работы программы

"C:\Program Files\Python313\python.exe" "C:\Users\Денис\Desktop\ДЗ
Парадигмы\Питон\di.cpp.py"
Сохранение 'Привет, DI на Python!' в PostgreSQL

Process finished with exit code 0

```
"C:\Program Files\Python313\python.exe" "C:\Users\Денис\Desktop\ДЗ Парадигмы\Питон\di.cpp.py"
Сохранение 'Привет, DI на Python!' в PostgreSQL
```

```
Process finished with exit code 0
```