

Кафедра компьютерной инженерии и моделирования

Порозов Кирилл Сергеевич

отчет по практической работе №5
по дисциплине «ПАРАДИГМЫ ПРОГРАММИРОВАНИЯ»

Направление подготовки:
09.03.04 "Программная инженерия"



Оценка -

Симферополь, 2022

Практическая работа №5. Тема: "Прототипное программирование"

Цель работы: Изучить парадигму прототипного программирования, преимущества и недостатки парадигмы, научиться создавать простейшие приложения на языке высокого уровня, реализующие прототипный подход к созданию программ.

Описание ключевых понятий:

(при необходимости)

Перед выполнением практической работы изучена следующая литература:

1. Презентация лектора курса: «Прототипное программирование»
2. Прослушана видеолекция и посещены практические занятия Милюкова Виктора Васильевича.
3. Прочитаны материалы статей: [Шаблон проектирования Прототип \(Prototype\) в Python - подробно с примерами \(pythonpip.ru\)](http://pythonpip.ru) ; [The Prototype Design Pattern in Python \(stackabuse.com\)](http://stackabuse.com)

Написана программа на языке Python, реализующая алгоритм:

Пример использования прототипной парадигмы для объявления новых экземпляров класса:

[Prog_Paradigms/Отчет структурное программирование.docx at main · JustBlood/Prog_Paradigms \(github.com\)](https://github.com/JustBlood/Prog_Paradigms/blob/main/Отчет%20структурное%20программирование.docx)

```
"""
Прототипное программирование:
Модель-прототип обеспечивает независимость от конкретного класса. Мы можем реализовать новый объект независимо от существующего класса.
Это также полезно для решения повторяющихся и сложных проблем разработки программного обеспечения.
"""

if __name__ == '__main__':
    # подгружаются все кеши для определенных классов для возможности их копирования
    Courses_Cache.load()

    # Новый объект создается не объявлением нового экземпляра класса Python() (также для Java(), R() и др.),
    # а с помощью копирования уже существующего класса из кеша-функции с помощью кеша.
    Python1 = Courses_Cache.get_course("1")
    print(Python1.get_type())

    Java2 = Courses_Cache.get_course("2")
    print(Java2.course())

    java3 = Courses_Cache.get_course("2")
    java4 = Courses_Cache.get_course("2")
    java5 = Courses_Cache.get_course("2")
    java6 = Courses_Cache.get_course("2")
    java7 = Courses_Cache.get_course("2")
    print(id(java3),
          id(java4),
          id(java5),
          id(java6),
          id(java7),
          "Как можем заметить - все ID РАЗНЫЕ! => объекты копируются корректно",
          sep=', ')

    R3 = Courses_Cache.get_course("3")
    print(R3.get_type())
```

main ×

"C:\Users\User\Pictures\уник\2 курс\Парадигмы программирования\venv\Scripts\python.exe" "C:\Users\User\Pictures\уник\2 курс\Парадигмы программирования\Python Basic and Algorithm

Inside Java :: course() method.

None

2318231841184, 2318231841328, 2318231841520, 2318231841616, 2318231841712, Как можем заметить - все ID РАЗНЫЕ! => объекты копируются корректно

R programming language

Более подробно – на странице задания на гитхабе.

Проект представлен преподавателю в электронной форме, продемонстрирована работоспособность программы, разъяснены детали программного кода.

Вопросы, заданные преподавателем: