

Архитектура уровня динамически типизированного языка

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Архитектура вычислительных систем

БПИ207

Вяльцев Павел

Москва
2021 г.

ОПИСАНИЕ ЗАДАНИЯ

Разработка программного продукта с использованием динамической типизации на Python.

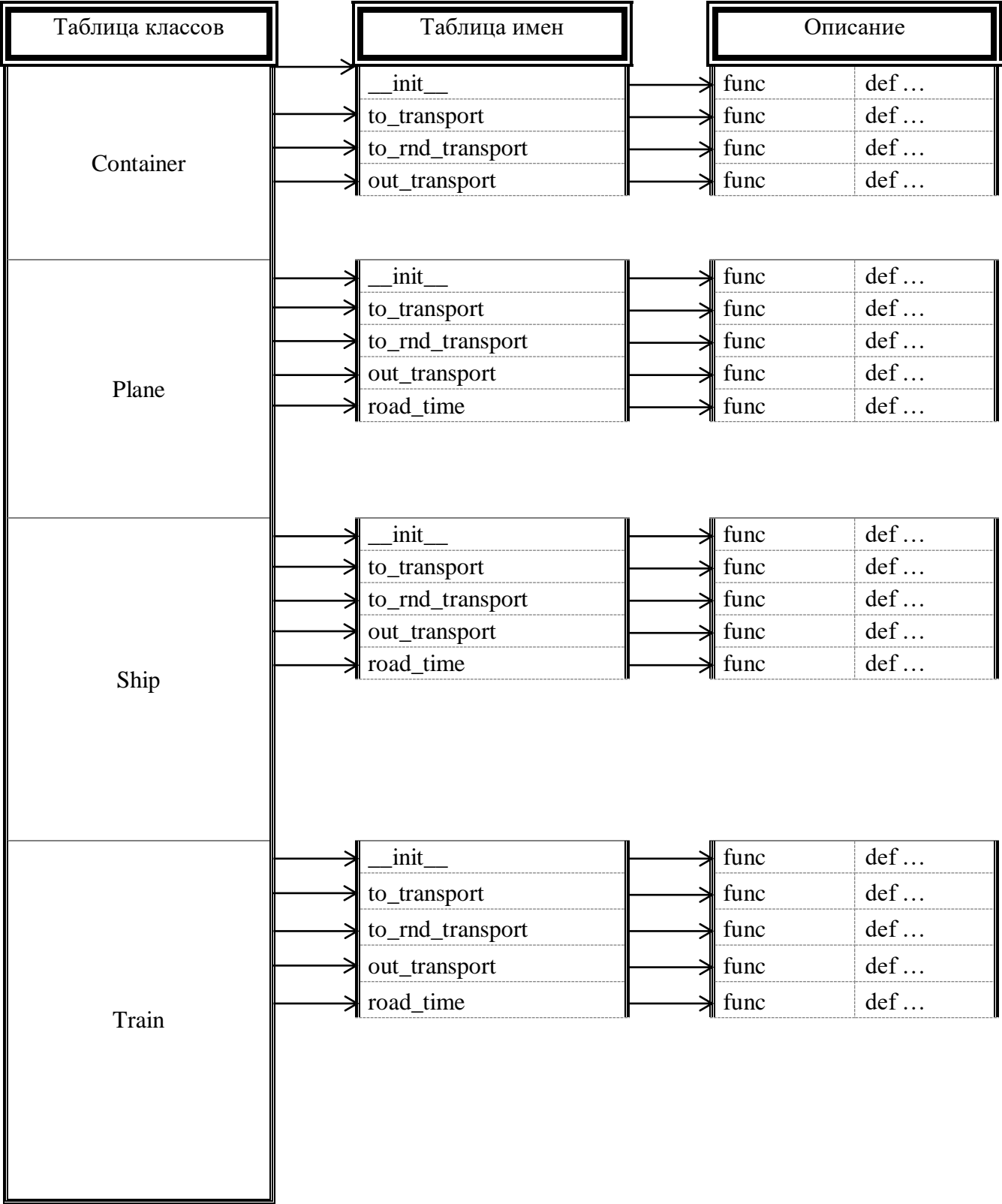
Номер варианта: 244

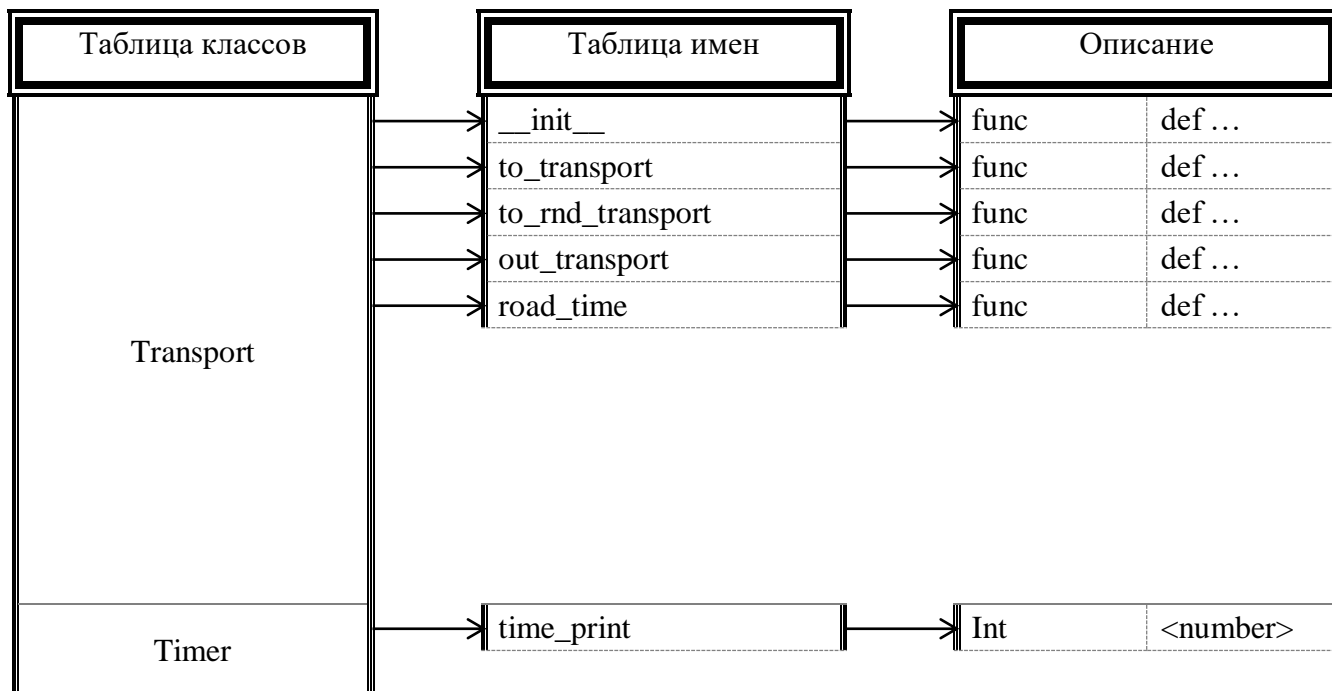
Программа содержит следующие структуры и функции:

Обобщенный артефакт, используемый в задании	Базовые альтернативы (уникальные параметры, задающие отличительные признаки альтернатив)	Общие для всех альтернатив переменные	Общие для всех альтернатив функции
Пассажирский транспорт	1. Самолеты (дальность полета – целое, грузоподъемность – целое) 2. Поезда (количество вагонов – целое) 1. Корабли (водоизмещение – целое; вид судна – перечислимый тип = (лайнер, буксир, танкер)	Скорость – целое; Расстояние между пунктами отправления и назначения – действительное	Идеальное время прохождения пути (действительное число)

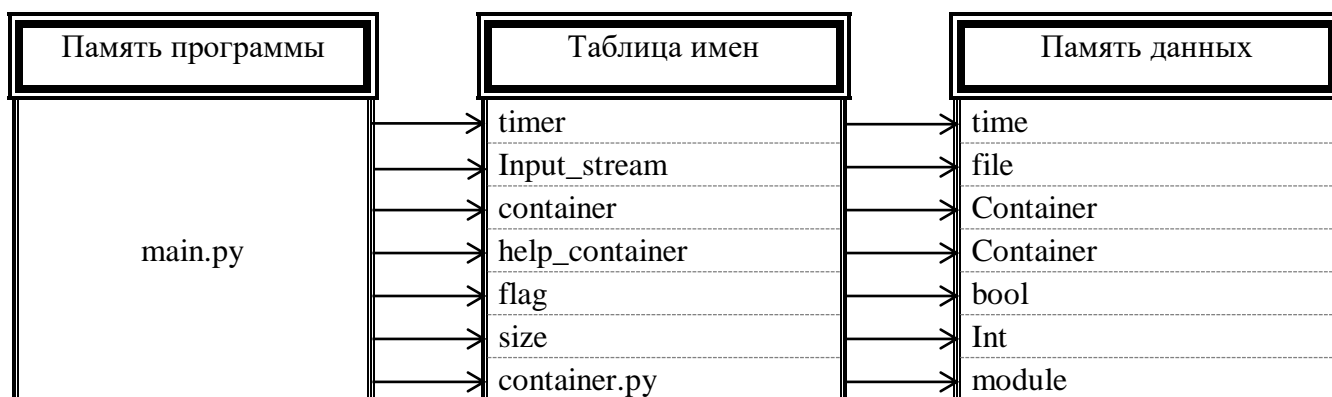
Также программа должна сортировать элементы контейнера по убыванию используя сортировку с помощью прямого слияния (Straight Merge). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив (т. е идеальное время прохождения пути).

СТРУКТУРНАЯ СХЕМА КЛАССОВ:





ОТРАЖЕНИЕ НА ПАМЯТЬ МЕТОДОВ КЛАССОВ:



ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ:

- Число модулей реализации – 7
- Общий размер исходных текстов – 345.
- Размер файлов – 15Кб
- Время выполнения программы для различных тестовых прогонов:

тест	Время
Input 1	0.000966787
input 2	0.000999928
input 3	0.000997782
input 4	0.000960827
input 5	0.001002073
Генерация 20 элементов	0.006009102
Генерация 10000 элементов	0.638967037

СРАВНЕНИЕ С ПРЕДЫДУЩИМИ ВЕРСИЯМИ ПРОГРАММЫ:

Вспомним результаты подобной программы, реализованной на С(процедурный подход) и С++(ооп). Оказывается, что программа, разработанная на Python работает в несколько раз медленнее, причем строки кода уменьшились почти что в 2 раза

Из-за динамической типизации, тип объектов, определяется во время исполнения и не указывается явно. Это и становится причиной уменьшения строк кода, но увеличения времени работы.

За счет “утиной” типизации, методы больше не ожидают конкретных типов данных для работы, а подстраиваются под выделенные.

Python не просто так до сих пор множество программистов используют как основной язык. Для программ, где не важна скорость, он незаменим, так как интуитивно понятен и удобен.