

Пояснительная записка

Описание полученного задания:

Задача номер 6 (пассажирский транспорт), функция номер 11 (сортировка выбором по убыванию).

- ✓ Обобщенный артефакт, используемый в задании – Пассажирский транспорт;
- ✓ Базовые альтернативы и их уникальные параметры:
 1. *Самолет*: дальность полета – целое, грузоподъемность – целое;
 2. *Поезд*: количество вагонов – целое;
 3. *Корабль*: водоизмещение – целое, вид судна – перечислимый тип (лайнер, буксир, танкер).
- ✓ Общие для всех альтернатив переменные: скорость – целое, расстояние между пунктами отправления и назначения – действительное.
- ✓ Общая для всех альтернатив функция: идеальное время прохождения пути (действительное).
- ✓ Функция: упорядочить элементы контейнера по убыванию используя сортировку Сортировка с помощью прямого выбора (Straight Selection). В качестве ключей для сортировки и других действий используются результаты функции, общей для всех альтернатив.

Структура проекта:

project/

test/ - входные и выходные файлы для тестирования программы;

main.cpp – точка входа.

/.../ - файлы реализации проекта.

Файлы реализации проекта: transport.py, container.py

Основные характеристики программы:

- Количество модулей: 3
- Размер исходных файлов: ≈ 19.6 Кб
- Время выполнения программы на различных тестовых файлах (взято усредненное значение по результатам нескольких запусков):

<i>Количество элементов</i>	<i>Тип входных данных</i>	<i>Время работы (мс)</i>
0	Ввод с файла	15.1
2	Ввод с файла	15.56
4	Ввод с файла	15.624
8	Ввод с файла	18.7
10	Ввод с файла	22.4
10	Генерация данных	18.19
100	Генерация данных	26.38
1000	Генерация данных	683.13
5000	Генерация данных	11242.9
10000	Генерация данных	45888.2789

СРАВНЕНИЕ РЕЗУЛЬТАТОВ РАБОТЫ ПРОГРАММ:

Для обоих типов ввода данных время выполнения программы заметно увеличилось в данной реализации на Python в сравнении с предыдущими двумя реализациями на C/C++. Это связано:

- 1) С тем, что Python – интерпретируемый язык программирования, а интерпретатор выполняет код без предварительного перевода, что сильно сказывается на времени выполнения программы. C/C++ – компилируемые языки, поэтому скомпилированный код исполняется в изначальном контексте.
- 2) В Python – динамическая типизация, а это значит, что при в ходе работы программы тратится больше времени на проверку типов переменных (чего получается избежать в языках программирования со статической типизацией, как C/C++).

Примечание к тестовым файлам: для каждого типа входных данных отведено по 5 тестов, идущих по возрастанию количества обрабатываемых элементов. В выходном файле присутствуют и несортированный, и отсортированный варианты. Пара входной-выходной файл имеет одинаковый индекс (1-5 для первого типа входных данных, 5-6 для второго типа входных данных).

Архитектура приложения:

Схема классов имеет следующий вид:

transport

- speed: int;
- distance: float;
- def __init__(self, speed, distance);
- def __str__(self);
- def input_file(in_file); @staticmethod
- def input_random(); @staticmethod
- def get_time(self) -> float;
- def output_random(self, input_file); @abstractmethod

airplane

- speed: int;
- distance: float;
- range_of_flight: int;
- lifting_capacity: int;
- def __init__(self, speed, distance, range_fl, capacity);
- def __str__(self);
- def input_file(in_file); @staticmethod
- def input_random() -> 'Airplane'; @staticmethod
- def output_random(self, input_file);

train

- speed: int;
- distance: float;
- count_of_cars;
- def __init__(self, speed, distance, count);
- def __str__(self);
- def input_file(in_file); @staticmethod
- def input_random() -> 'Train'; @staticmethod
- def output_random(self, input_file);

ship

- speed: int;
- distance: float;
- int displacement;
- ship_type: ShipType;
- def __init__(self, speed, distance, displacement, type_sh);
- def __str__(self);
- def input_file(in_file); @staticmethod
- def input_random() -> 'Ship'; @staticmethod
- def output_random(self, input_file);

Enum:

- TransportType
- ShipType

container

- MAX_LENGTH = 10000;
- transports: list;
- def __init__(self);
- def __str__(self);
- def add_transport(self, transport) -> bool
- def input_file(in_file); @staticmethod
- def input_random(count); @staticmethod
- def output_file(self, out_file);
- def output_random(self, random_file);
- def straight_selection(self);

Инструментальные средства:

- Сведения о системе:
 - ОС Windows 10
 - 2 ядра
 - 4 логических процессора
 - Оперативная память: 4.00 ГБ
- Языки программирования: Python 3.9;
- IDE: PyCharm Community (2021.2.3);
- Модули: sys, time, abs, random, enum