

*процедурный подход*

# **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

## **Архитектура вычислительных систем**

БПИ207

Вяльцев Павел

Москва  
2021 г.

# ОПИСАНИЕ ЗАДАНИЯ

Разработка программного продукта с использованием статической типизации и процедурного подхода.

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).

**Таблица типов, используемых в программе.**

<b>Название</b>	<b>Размер</b>
<i>int</i>	<b>4</b>
<i>double</i>	<b>8</b>
<i>char</i>	<b>1</b>
<i>FILE</i>	<b>216</b>
<i>struct transport</i>	<b>24</b>
<i>enum transport type</i>	<b>4[0]</b>
<i>int speed</i>	<b>4</b>
<i>double the road</i>	<b>8</b>
<i>Union {</i>	
<i>Structure_p p</i>	<b>8[16]</b>
<i>Structure_s s</i>	<b>8[16]</b>
<i>Structure_t t</i>	<b>4[16]</b>
<i>}</i>	
<i>struct structure_p</i>	<b>8</b>
<i>Int flight range</i>	<b>4[0]</b>
<i>Int load capacity</i>	<b>4[4]</b>
<i>struct structure_s</i>	<b>8</b>
<i>enum type_s</i>	<b>4[0]</b>
<i>int water_displacement</i>	<b>4[4]</b>
<i>Struct structure_t</i>	<b>4</b>
<i>number_of_wagons</i>	<b>4[0]</b>
<i>Struct struct_cont</i>	<b>80004</b>
<i>Struct_transport data</i>	<b>8000[0]</b>
<i>Int count</i>	<b>4[4]</b>

## Схема – main

<i>Stack при ошибке</i>	<i>Stack при вводе из файла</i>	<i>Stack при генерации</i>
<b>clock</b>	<b>clock</b>	<b>clock</b>
<b>(1)WrongArguments</b>	<b>Initialize</b>	<b>Initialize</b>
<b>(2)WrongMode</b>	<b>strcmp</b>	<b>strcmp</b>
<b>TimePrint</b>	<b>fopen</b>	<b>fopen</b>
<b>exit</b>	<b>FileCheck</b>	<b>FileCheck</b>
	<b>Fill_cont</b>	<b>Fill_cont_Rand</b>
	<b>fopen</b>	<b>fopen</b>
	<b>FileCheck</b>	<b>FileCheck</b>
	<b>Out_cont</b>	<b>Out_cont</b>
	<b>Straight_Merge</b>	<b>Straight_Merge</b>
	<b>Printf / fprintf</b>	<b>Printf / fprintf</b>
	<b>Out_cont</b>	<b>Out_cont</b>
	<b>Cleat_cont</b>	<b>Cleat_cont</b>
	<b>TimePrint</b>	<b>TimePrint</b>

<i>Heap</i>	<b>Name</b>	<b>Memory</b>
"command "	argc:int	4[0]
"-f"/"-n"	argv:char**	8[4]
"input.txt"	start:clockStart	4[12]
"output.txt"	Container:struct_container	80004[16]
	Input_stream:FILE	8[80020]
	Output_stream:FILE	8[80028]
	Help_container:struct_container	80004 [80036]

<b>Глобальная память</b>	<b>Память команд</b>
0	int main(int argc, char* argv[]) {...}

## ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПО

Код программы содержится в 19 файлах.

Всего 9 заголовочных файлов и 10  
файлов реализации.

Всего строк в программе: 658

Вес исполняемого кода: 39 Кб

## Тестовые прогоны

Название	Время
Input1.txt	0.0130000003 seconds
Input2.txt	0.0109999999 seconds
Input3.txt	0.00999999978 seconds
Input4.txt	0.00600000005 seconds
Input5.txt	0.00899999961 seconds
errorInput1.txt	0.00200000009 seconds
errorInput2.txt	0.00100000005 seconds

## Прогоны генерацией

Кол-во элементов	Время
20	0.0209999997 seconds
10000	2.56500006 seconds



