

Вариант №8, функция №8.

Необходимо было разработать программный продукт с использованием объектно-ориентированного подхода и статической типизации. Разработка осуществлялась на языке программирования C++. Использовались библиотеки языка C. Обобщённым артефактом являлись языки программирования, в качестве дополнительной функции реализован Quick Sort. Была использована реализация со статическим выделением памяти для контейнера.

Таблица используемых типов в разработанной программе:

Type table	
Name	Size (bytes)
int	4
double	8
void*	8
typification (enum)	4[0]
inheritance (enum)	4[0]
bool	1
ProgrammingLanguage (class)	20
virtualTablePtr	8[0]
double popularity	8[8]
int year_of_creation	4[16]
Procedural (class)	21
virtualTablePtr	8[0]
double popularity	8[8]
int year_of_creation	4[16]
bool are_abstract_types_present	1[20]
Functional (class)	24
virtualTablePtr	8[0]
double popularity	8[8]
int year_of_creation	4[16]
typification t	4[20]
bool is_lazy_calculations_support	1[24]
ObjectOriented (class)	24
virtualTablePtr	8[0]
double popularity	8[8]
int year_of_creation	4[16]
inheritance in	4[20]
Container(class)	240004
int len	4[0]
ProgrammingLanguage cont	240000[4]

Память программы с некоторыми функциями:

Program memory	
function	size
main	240024
return : int	4
argc : int	4
argv : char*	8
c : Container	240004
size : int	4
ProgrammLanguage::StaticIn	24
language : ProgrammLanguage*	8
k : int	4
double popularity	8
int year_of_creation	4
QuickSort	12
low : int	4
hihg : int	4
pivot : int	4
Partition	37
low : int	4
hihg : int	4
pivot : ProgrammLanguage	21
i : int	4
j : int	4
Swap	24
a : ProgrammingLanguage**	8
b : ProgrammingLanguage**	8
t : ProgrammingLanguage*	8

Global memory
virtualTable*

Число интерфейсных модулей: 6, число файлов с определением программных объектов: 7.

Общий размер исходных текстов: 11 326 байт, размер исполняемого файла: 24 840 байт.

Размер по сравнению с прошлым заданием почти не изменился (10 010 байт и 24 224 байт соответственно).

Проведённые тесты с замераами времени:

Tests (task 2)	
Test	Time (seconds)
1 (1 element from file)	0.000251
2 (5 elements from file)	0.000271
3 (10 elements from file)	0.000327
4 (15 elements from file)	0.000325
5 (20 elements from file)	0.000380
6 (1000 elements from random)	0.003388
7 (5000 elements from random)	0.017196
7 (10000 elements from random)	0.035957

По сравнению с прошлым заданием, (тесты ниже) на больших случайных значениях заметно увеличена скорость работы программы (где-то в 1.4 раза быстрее). Может, это связано с выводом в файл и заменой использования библиотеки `fstream` на `stdio.h`, или это особенность текущей ОО реализации.

Tests (task 1)	
Test	Time (seconds)
1 (10 elements from file)	0.000351
2 (1 element from file)	0.000250
3 (20 elements from file)	0.000362
4 (1000 elements from random)	0.006086
5 (5000 elements from random)	0.030997
6 (10000 elements from random)	0.065918