

Практическое задание №6 (структуры)

Создать проект с функциями **Task6** с описанием структуры и реализацией функций согласно варианту задания.

Исходные данные хранятся в обычном текстовом файле **формата JSON**. Файл подготовить самостоятельно, используя **кодировку символов UNICODE** (см. файл «Материалы к Практическому заданию №6.pdf»). Например, для определённой в файле Teacher.hpp структуры

```
struct Teacher
{
    wchar_t Name[20];    // Фамилия
    short Number;        // Личный номер
    int Groups[5];       // Перечень номеров групп
};
```

в текстовом файле **формата JSON** (например, data.json) должны содержаться данные соответствующего вида:

```
[
  {
    "Name": "Петухова",
    "Number": 12345,
    "Groups": [7311, 7312, 7112, 7321, 7322]
  },
  {
    "Name": "Боловцов",
    "Number": 67890,
    "Groups": [7311, 7312, 7315, 7321, 7322]
  },
  {
    "Name": "Сорокин",
    "Number": 98765,
    "Groups": [7315, 7321, 7322, 7325, 7335]
  }
]
```

Чтение данных из текстового файла производить в строку с использованием функции **fread()** (см. «Материалы к Практическому заданию №6.pdf»), а затем из строки данные в поля структур считывать с использованием функции форматного ввода **swscanf()**.

Внимание: считать, что файл всегда содержит **корректно заданный** массив структур. Использовать сторонние библиотеки для разбора формата JSON **запрещается**.

Вывод в строку элементов массива структур осуществлять в виде таблицы с использованием функции форматного вывода **swprintf()**. Требуется рассчитать необходимую максимальную ширину полей в соответствии с данными, содержащимися в массиве, данные в таблице (кроме строк) форматировать по центру колонки, например:

№	Фамилия	Номер	Группа1	Группа2	дальше работает ваша фантазия
1	Петухова	12345	7311	7312	
2	Боловцов	67890	7311	7312	
3	Сорокин	98765	7315	7321	

Для сдачи проекта использовать структуру в файловой системе:

- gxxxxx/5/Task5/*** – файлы с реализацией пунктов задания (имена файлов давать в соответствии с вариантами);
- gxxxxx/5/Task5.Tests/*** – файлы с тестами пунктов задания (имена файлов давать в соответствии с вариантами).

Необходимые условия сдачи:

- стиль программирования должен соответствовать установленным правилам;
- имена переменных и функций должны быть осмысленными; никакой транслитерации и нелогичности;
- программа должна быть протестирована с помощью gtest;
- программа должна проходить тесты преподавателя (которых вы не видите).

Варианты заданий**Вариант 1**

Описать структуру с именем **Computer**, содержащую следующие поля:

- wchar_t Type[30] – тип компьютера (Desktop, Notebook и т.д.);
- double SpeedOfProcessor – быстродействие процессора (в ГГц);
- double RAM – объём оперативной памяти (в Гбайтах);
- wchar_t OperatingSystem[20] – тип операционной системы.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа Computer) и формирование динамического массива ComputerClass, состоящего из структур типа Computer; возвращать сформированный массив ComputerClass;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива ComputerClass по быстродействию процессора;
4. поиск в массиве ComputerClass компьютеров, тип которых и объём оперативной памяти совпадают с типом и объёмом, переданными в функцию в качестве параметров; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких компьютеров нет, возвращать nullptr.

Вариант 2

Описать структуру с именем **Figure**, содержащую следующие поля:

- wchar_t Name[30] – название фигуры на плоскости;
- int Coordinates[2] – координаты центра тяжести;
- wchar_t Color[20] – цвет фигуры.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа Figure) и формирование динамического массива Picture, состоящего из структур типа Figure; возвращать сформированный массив Picture;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива Picture по расстоянию центра тяжести фигур от начала координат;
4. поиск в массиве Picture фигур, цвет которых совпадает с цветом, переданным в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких фигур нет, возвращать nullptr.

Вариант 3

Описать структуру с именем **Book**, содержащую следующие поля:

- `wchar_t Author[30]` – фамилия автора;
- `wchar_t Title[80]` – название произведения;
- `wchar_t Publishing[20]` – издательство;
- `int YearOfRelease` – год выпуска.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с помощью функции форматного ввода **fwscanf()** данных (полей структур типа **Book**) и формирование динамического массива **Library**, состоящего из структур типа **Book**; возвращать сформированный массив **Library**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Library** в алфавитном порядке (по фамилиям авторов);
4. поиск в массиве **Library** книг, опубликованных издательством (переданным в функцию в качестве параметра) не ранее года, также переданного в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких книг нет, возвращать **nullptr**.

Вариант 4

Описать структуру с именем **Schedule**, содержащую следующие поля:

- `wchar_t Name[30]` – фамилия преподавателя;
- `wchar_t Subject[16]` – название предмета;
- `bool Days[6]` – дни недели, по которым есть занятия (отмеч. в массиве значением **true**).

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Schedule**) и формирование динамического массива **List**, состоящего из структур типа **Schedule**; возвращать сформированный массив **List**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **List** в алфавитном порядке (по названиям предметов);
4. поиск в массиве **List** преподавателей, занимающихся по дням недели, переданным (в виде массива) в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких преподавателей нет, возвращать **nullptr**.

Вариант 5

Описать структуру с именем **Employee**, содержащую следующие поля:

- `wchar_t Name[32]` – фамилия служащего;
- `wchar_t Profession[20]` – профессия;
- `wchar_t Education[15]` – уровень образования;
- `int LengthOfService` – стаж работы (в месяцах).

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Employee**) и формирование динамического массива **Office**, состоящего из структур типа **Employee**; возвращать сформированный массив **Office**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Office** в алфавитном порядке (по полю **Name**);
4. поиск в массиве **Office** работников с высшим образованием, чей стаж работы в организации превышает значение, переданное в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких работников нет, возвращать **nullptr**.

Вариант 6

Описать структуру с именем **Muzician**, содержащую следующие поля:

- `wchar_t Name[32]` – фамилия музыканта;
- `wchar_t Phone[10]` – название музыкального инструмента;
- `short PointsOfCompetition[3]` – баллы трёх этапов конкурса.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Muzician**) и формирование динамического массива **ListOfMuzicians**, состоящего из структур типа **Muzician**; возвращать сформированный массив **ListOfMuzicians**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **ListOfMuzicians** в алфавитном порядке (по полю **Name**);
4. поиск в массиве **ListOfAthletes** музыкантов, играющих на инструменте, название которого передано в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких музыкантов нет, возвращать **nullptr**.

Вариант 7

Описать структуру с именем **Medicine**, содержащую следующие поля:

- `wchar_t Name[80]` – название лекарства;
- `int Storage[3]` – предельный срок хранения (дата);
- `short Cost` – стоимость лекарства в руб.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Medicine**) и формирование динамического массива **Register**, состоящего из структур типа **Medicine**; возвращать сформированный массив **Register**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Register** в алфавитном порядке по названиям лекарств (по полю **Name**);
4. поиск в массиве **Register** лекарств, срок хранения которых истекает через *n* дней; значение *n* передаётся в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких лекарств нет, возвращать **nullptr**.

Вариант 8

Описать структуру с именем **Athlete**, содержащую следующие поля:

- `wchar_t Name[32]` – фамилия спортсмена;
- `wchar_t Sport[10]` – название вида спорта;
- `int Birthday[3]` – дата рождения (массив из трёх чисел).

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Athlete**) и формирование динамического массива **ListOfAthletes**, состоящего из структур типа **Athlete**; возвращать сформированный массив **ListOfAthletes**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **ListOfAthletes** в алфавитном порядке (по полю **Name**);
4. поиск в массиве **ListOfAthletes** спортсменов, возраст которых не превышает значение, переданное в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких спортсменов нет, возвращать **nullptr**.

Вариант 9

Описать структуру с именем **Train**, содержащую следующие поля:

- `wchar_t Destination[30]` – название пункта назначения;
- `int Number` – номер поезда;
- `int Time[3]` – время отправления (массив из трёх чисел).

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с помощью функции форматного ввода **fwscanf()** данных (полей структур типа **Train**) и формирование динамического массива **Schedule**, состоящего из структур типа **Train**; возвращать сформированный массив **Schedule**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Schedule** в алфавитном порядке (по названиям пунктов назначения);
4. поиск в массиве **Schedule** поездов, отправляющихся позже времени, переданного в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких поездов нет, возвращать **nullptr**.

Вариант 10

Описать структуру с именем **Price**, содержащую следующие поля:

- `wchar_t Goods[80]` – название товара;
- `wchar_t Shop[16]` – название фирмы-производителя;
- `int Cost` – стоимость товара в руб.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Price**) и формирование динамического массива **ListOfGoods**, состоящего из структур типа **Price**; возвращать сформированный массив **ListOfGoods**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **ListOfGoods** в алфавитном порядке по названиям товаров (по полю **Goods**);
4. поиск в массиве **ListOfGoods** товаров, изготовленных фирмой, название которой передано в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких товаров нет, возвращать **nullptr**.

Вариант 11

Описать структуру с именем **Order**, содержащую следующие поля:

- `wchar_t Payer[16]` – расчётный счёт плательщика;
- `wchar_t Recipient[16]` – расчётный счёт получателя;
- `int Summa` – перечисляемая сумма в руб.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Order**) и формирование динамического массива **List**, состоящего из структур типа **Order**; возвращать сформированный массив **List**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **List** по расчётным счетам плательщиков;
4. поиск в массиве **List** счетов, перечисленные суммы с которых превышают значение, переданное в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких счетов нет, возвращать **nullptr**.

Вариант 12

Описать структуру с именем **Zodiac**, содержащую следующие поля:

- `wchar_t Name[32]` – фамилия человека;
- `int Birthday[3]` – дата рождения (массив из трёх чисел);
- `wchar_t ZodiacSign[16]` – название знака зодиака.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Zodiac**) и формирование динамического массива **Book**, состоящего из структур **Zodiac**; возвращать сформированный массив **Book**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Book** в алфавитном порядке (по полю **Name**);
4. поиск в массиве **Book** людей, родившихся под знаком, наименование которого передано в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких людей нет, возвращать **nullptr**.

Вариант 13

Описать структуру с именем **Worker**, содержащую следующие поля:

- `wchar_t Name[32]` – фамилия работника;
- `wchar_t Post[10]` – название занимаемой должности;
- `int Data` – дата поступления на работу (массив из трёх чисел).

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с помощью функции форматного ввода **fwscanf()** данных (полей структур типа **Worker**) и формирование динамического массива **Table**, состоящего из структур типа **Worker**; возвращать сформированный массив **Table**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Table** в алфавитном порядке (по полю **Name**);
4. поиск в массиве **Table** работников, чей стаж работы в организации превышает значение, переданное в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких работников нет, возвращать **nullptr**.

Вариант 14

Описать структуру с именем **TrainRoute**, содержащую следующие поля:

- `wchar_t StartingPoint[20]` – название начального пункта маршрута;
- `wchar_t TerminalPoint[20]` – название конечного пункта маршрута;
- `int Number` – номер маршрута;
- `int distance` – длина маршрута в км.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с помощью функции форматного ввода **fwscanf()** данных (полей структур типа **TrainRoute**) и формирование динамического массива **Traffic**, состоящего из структур **TrainRoute**; возвращать сформированный массив **Traffic**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Traffic** в алфавитном порядке (по полю **StartingPoint**);
4. поиск в массиве **Traffic** маршрутов, название начального пункта которого передано в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких маршрутов нет, возвращать **nullptr**.

Вариант 15

Описать структуру с именем **Passage**, содержащую следующие поля:

- `wchar_t Destination[20]` – название пункта назначения рейса;
- `int Number` – номер рейса;
- `wchar_t Planetype[10]` – тип самолета.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Passage**) и формирование динамического массива **Aeroport**, состоящего из структур типа **Passage**; возвращать сформированный массив **Aeroport**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Aeroport** по возрастанию номера рейса;
4. поиск в массиве **Aeroport** рейсов, вылетающих в пункт назначения, название которого передано в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких рейсов нет, возвращать **nullptr**.

Вариант 16

Описать структуру с именем **Note**, содержащую следующие поля:

- `wchar_t Name[32]` – фамилия;
- `wchar_t Phone[10]` – номер телефона;
- `short Birthday[3]` – дата рождения (массив из трёх чисел).

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с помощью функции форматного ввода **fwscanf()** данных (полей структур типа **Note**) и формирование динамического массива **Blocknote**, состоящего из структур **Note**; возвращать сформированный массив **Blocknote**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **Blocknote** в алфавитном порядке (по полю **Name**);
4. поиск в массиве **Blocknote** людей, чьи дни рождения приходятся на месяц, значение которого передано в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких людей нет, возвращать **nullptr**.

Вариант 17

Описать структуру с именем **Entrant**, содержащую следующие поля:

- `wchar_t Name[30]` – фамилия абитуриента;
- `wchar_t Phone[10]` – номер телефона;
- `int Estimates[4]` – оценки по предметам за ЕГЭ.

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла с использованием функции форматного ввода **fwscanf()** данных (полей структур типа **Entrant**) и формирование динамического массива **ListOfEntrants**, состоящего из структур типа **Entrant**; возвращать сформированный массив **ListOfEntrants**;
2. вывод в файл массива структур в виде таблицы с использованием функции форматного вывода **fwprintf()**;
3. упорядочение массива **ListOfEntrants** в алфавитном порядке (по полю **Name**);
4. поиск в массиве **ListOfEntrants** абитуриентов, средний балл оценок за ЕГЭ которых больше значения, переданного в функцию в качестве параметра; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких абитуриентов нет, возвращать **nullptr**.

Вариант 18

Описать структуру с именем **Student**, содержащую следующие поля:

- `wchar_t name[30]` – фамилия студента;
- `int groupNumber` – номер группы;
- `short marks[5]` – успеваемость (массив из пяти элементов – экзаменационных оценок по предметам в сессию).

Разработать функции, выполняющие следующие действия:

1. чтение из текстового файла в строку с использованием функции **fread()**;
2. формирование динамического массива `Students`, состоящего из структур типа `Student` и заполнение полей структур из строки с использованием функции форматного ввода данных **swscanf()**; функция должна возвращать сформированный массив `Students`;
3. вывод в строку массива структур в виде таблицы с использованием функции форматного вывода **swprintf()**;
4. упорядочение массива `Students` в алфавитном порядке фамилий студентов;
5. поиск в массиве `Students` студентов, не имеющих неудовлетворительных и удовлетворительных экзаменационных оценок по предметам в сессию; результаты поиска возвращать в виде указателя на динамически созданный массив с результатами поиска; если таких студентов нет, возвращать `nullptr`.