

Занятие 12. Указатели и списки

Задание 1

Используя рассмотренный на лекции пример, напишите функцию `std::string findAddress(shared_ptr<Node> head, std::string name)`, которая по имени человека возвращает его адрес. На вход функции подаётся указатель `head` на список структур `Node` и имя искомого человека.

Структура `Node` содержит два умных указателя `std::shared_ptr`, один из которых указывает на следующий элемент списка, а второй – на класс с данными `TGUPerson`. Класс `TGUPerson` содержит поля: `_name` – имя, и `_address` – адрес (тип обоих полей `std::string`). Функция возвращает адрес человека или пустую строку, если такого человека в списке нет.

Задание 2

Реализуйте структуру `IntNode` и минимальный набор функций для работы со списками целых чисел.

Структура `IntNode` должна включать:

- данные – значение целого типа;
- умный указатель `shared_ptr` на следующий элемент списка.

Для работы со списками целых чисел реализуйте:

- `shared_ptr<IntNode> createList()` – создаёт пустой список (т.е. первый элемент для головы);
- `void addElement(shared_ptr<IntNode> head, int value)` – добавляет в начало списка, заданного указателем на голову `head`, новый элемент со значением `value`.
- `std::ostream& operator << (std::ostream& stream, shared_ptr<IntNode> head)` – оператор вывода, печатающий содержимое списка, заданного указателем на голову `head`, в поток `stream`.

При реализации этих функций ориентируйтесь на рассмотренный на лекции пример, обратив внимание, что в классе рассматривался вариант, в котором в качестве поля данных в узле списка лежал указатель на класс, а в этом задании

поле данных целого типа размещается прямо в узле списка (для него не нужен указатель).

Задание 3

Напишите функцию `bool listCompare(shared_ptr<IntNode> head1, shared_ptr<IntNode> head2)`, которая сравнивает два поданных на вход списка целых чисел. Функция возвращает `true`, если списки равны, иначе `false`. Списки равны, если они содержат одни и те же элементы, которые идут в одинаковом порядке.

Задание 4

Напишите функцию `void swapElems(shared_ptr<IntNode> prev)`, которая меняет местами два соседних элемента списка путём переписывания ссылок между элементами. На вход функции подаётся указатель на элемент списка, предшествующий первому из переставляемых элементов. Если отсутствует хотя бы один из переставляемых элементов, функция не должна ничего делать.

Задание 5

Напишите функцию `void bubbleSort(shared_ptr<IntNode> head)`, которая сортирует методом пузырька поданный на вход список целых чисел, используя функцию из задания 4 для перестановки соседних элементов.

Оценка задания

Задание 1	- 15%.
Задание 2	- 20%.
Задание 3	- 15%.
Задание 4	- 25%.
Задание 5	- 25%.

Баллы за каждую задачу включают:

- правильность решения задания 60 %;
- оформление кода согласно <http://prog.tversu.ru/pr3/codeStyle.pdf> 10 %;
- программа компилируется без предупреждений компилятора на 4-ом уровне 10%;
- модульные тесты 20%.