Абстрактные типы данных

Юрий Литвинов
yurii.litvinov@gmail.com

26.10.2018

АТД

- АТД — некоторая математическая модель и набор операций, определённый в рамках этой модели
  - Обобщение понятия “тип”
- Состоит из типа данных и операций, выполняющих над ним преобразования
  - Внутреннее устройство типа данных невидимо для остальной программы (принцип сокрытия деталей реализации)
  - Работа с АТД — только с помощью связанных с ним функций
  - Тип данных и операции для работы с ним лежат в одном модуле, так, чтобы все изменения в АТД были локализованы и не затрагивали остальную программу (принцип инкапсуляции)
- Дальнейшее обобщение АТД – классы

Пример — стек

- stack.h / stack.cpp, при этом структура данных описана только в .cpp-файле, в .h-файле только её предварительное объявление
  - Так компилятор может гарантировать сокрытие деталей реализации
    - Всё, что не проверяется автоматически, можно считать не работающим!
  - Все функции принимают только указатель на структуру, для значения нужно знать размер
- Функции:
  - createStack()
  - deleteStack()
  - push()
  - pop()
  - isEmpty()
- Внешнему миру вообще всё равно, как стек устроен внутри
  - Может быть на массиве

Ещё пример — список

- Требуется целых два типа — сам список и позиция внутри списка
  - Что-то вроде индекса элемента массива, но может быть устроена хитрее
  - Позиция должна обеспечивать быструю работу с элементом, на который она указывает
  - Внешнему миру всё равно, как устроен список и что такое позиция
    - Может быть, список на массивах, а позиция — число, или список на указателях, а позиция — указатель на элемент списка (или даже на предыдущий элемент)
- Список может хранить разные типы элементов
  - typedef — “шаблоны для бедных”
    - ```
      typedef int Value;
      struct ListElement {
        Value value;
        ListElement *next;
      }
      ```
    - typedef же может использоваться для описания типа позиции
      ```

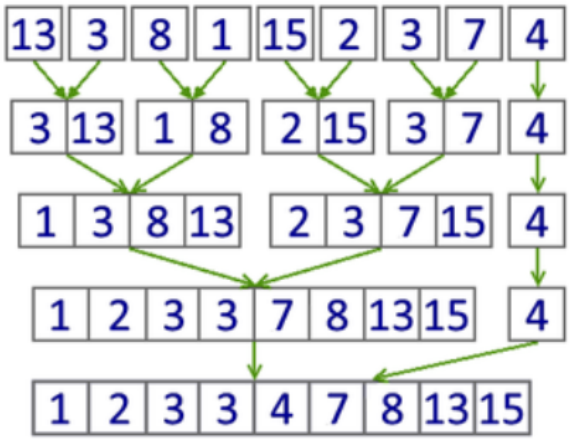
Инвариант

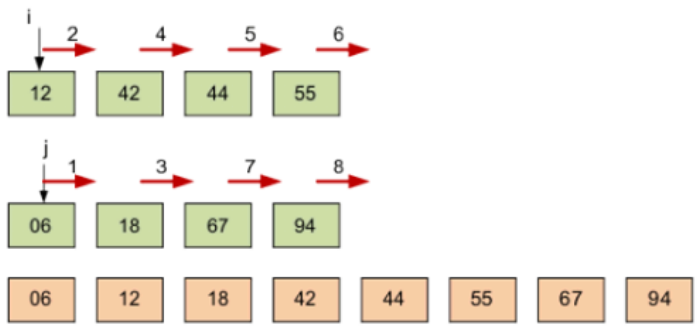
- Некоторое логическое условие, верное всё время жизни АТД
  - Не совсем, внутри функции АТД инвариант может нарушаться
    - Не всегда, потому что бывают многопоточные программы
- АТД отвечает за поддержание своего инварианта
  - Поскольку работа с АТД только через его функции, у внешнего мира нет способа его испортить
- Пример — размер списка
  - Можно считать за $O(n)$ каждый раз
  - Можно хранить как элемент структуры, тогда должен соблюдаться инвариант
- Ещё пример — head и tail у очереди

Пример применения АТД — сортировка слиянием

Если в списке больше одного элемента, делим его на два, вызываем mergesort, получаем два отсортированных списка, которые сливаем в один отсортированный

- $O(n * log(n))$ в среднем и худшем случае
- Устойчива
- Внешняя (подходит для больших данных, не помещающихся в память)
- http://www.ee.ryerson.ca/~courses/coe428/sorting/mergesort.html
- Ей не надо знать внутреннего устройства списка





Немного C++

- vector<**int**> array(10);
- string str = "C++ roxx";
- stack, queue, priority_queue, ...
  - Обратите внимание, STL (Standard Template Library, это оттуда все эти классы) не следует правилам стайлгайда
- cin/cout
- Итераторы и умные указатели (только для тех, кто знает, что это)

Пример, чтение из файла

```
#include <fstream>
#include <iostream>
#include <vector>
#include <string>

using namespace std;

int main()
{
    ifstream file("test.txt", ios::in);
    if (!file.is_open())
    {
    cout << "File not found!" << endl;
        return 1;
    }

    ...

    return 0;
}
```

Собственно, чтение

```
int main()
{
    ...
    vector<string> data;

    while (!file.eof()) {
        string buffer;
        file >> buffer;
        data.push_back(buffer);
    }

    file.close();

    for (string const &line : data)
    {
        cout << line << endl;
    }

    return 0;
}
```

using namespace, what?

Карго-культ, также религия самолётопоклонников или культ Даров небесных — термин, которым называют группу религиозных движений в Меланезии. В культах карго верят, что западные товары созданы духами предков и предназначены для меланезийского народа. Считается, что белые люди нечестным путём получили контроль над этими предметами. В культах карго проводятся ритуалы, похожие на действия белых людей, чтобы этих предметов стало больше.

© Wikipedia

Пространства имён

```
namespace namespace1
{
    struct A
    {
    }
}

namespace namespace2
{
    struct A
    {
    }
}
```

using namespace

```
using namespace namespace2;

int main()
{
    A a;
    namespace1::A a1;
}
```