

Лабораторная работа № 2

Функции и пользовательские типы данных

Цель работы

Освоить применение функций и пользовательских типов данных — структур и перечислений — для декомпозиции задачи и повышения выразительности программы.

Подготовка к лабораторной работе

Необходимо иметь две готовые программы, написанные в ходе ЛР № 1.

- 1) Принимающую на вход последовательность температур с указанием шкалы для каждой, а по завершении ввода печатающую таблицу тех же температур по шкалам Кельвина, Цельсия и Фаренгейта (пункт 2.5, далее «SR1»).
- 2) Рассчитывающую значение оценки корреляции между двумя наборами действительных чисел, вводимых пользователем (пункт 4, далее «SR2»).

Задание на лабораторную работу

1. Перепишите программу для расчета корреляции, используя функции для декомпозиции задачи, как это предложено в лекции № 2.

Следующие пункты выполняются на основе программы «SR1».

2. Определите типы данных для предметной области — температурных расчетов.
 - 2.1. Определите перечисление `Scale`, представляющее одну из температурных шкал Кельвина, Цельсия или Фаренгейта.
 - 2.2. Определите структуру `Temperature`, представляющую температуру со шкалой.
3. Обеспечьте ввод и вывод типов данных для предметной области.
 - 3.1. Реализуйте для перечисления `Scale` операторы ввода из потока и вывода в поток. Ошибки ввода обрабатывайте изменением состояния потока.
 - 3.2. Реализуйте для структуры `Temperature` операторы ввода из потока и вывода в поток без проверок корректности значения температуры.
4. Структурируйте программу «SR1».

- 4.1. Реализуйте функцию

```
double get_value_of_scale(  
    const Temperature& temperature,  
    const Scale& scale),
```

которая возвращает число градусов в `temperature` по шкале `scale`.

4.2. Перепишите программу «SR1», используя типы и функции из п. п. 1—4.1.

Указание. При неудачном вводе температуры можно сбросить флаг ошибки вызовом `cin.clear()`, а затем считать ненужную строку функцией `getline()`.

4.3. Добавьте в программу п. 4.2 обработку ошибок, возбуждая в функции `get_value_of_scale()` исключения и обрабатывая их в программе.

5. Реализуйте для структуры `Temperature` операторы отношений:

а) нечетные варианты: «меньше» (`<`), «равно» (`==`);

б) четные варианты: «больше или равно» (`>=`), «не равно» (`!=`).

Все операторы должны корректно работать, если у операндов разные шкалы.

6. Добавьте в реализацию оператора ввода структуры `Temperature` проверку условия, что температура не ниже 0 °К. Следует использовать операторы отношений, реализованные в п. 4.3, и обрабатывать ошибки ввода, как в п. 3.1.

Скопируйте результаты пунктов 2, 3, 4.1, 5 и 6 в программу «SR2» и продолжайте в ней.

7. Реализуйте для структуры `Temperature` математические операторы.

7.1. Деление (`/`) температуры на действительное число. Шкала результата должна совпадать со шкалой операнда-температуры.

7.2. Вычитание (`-`) температур. Шкала результата должна совпадать со шкалой первого операнда, причем шкалы операндов могут различаться.

7.3. Присваивание, совмещенное со сложением (`+=`). Шкала увеличиваемой температуры не должна изменяться; шкалы операндов могут не совпадать.

Все операторы должны проверять, что параметры- и результаты-температуры не ниже абсолютного нуля, и в случае нарушения этого условия возбуждать исключение `domain_error` с информативным сообщением.

Указание. Функция `to_string()` позволяет преобразовывать в строку любые значение, в том числе и структуры `Temperature`.

8. Измените программу «SR2» для работы с температурами вместо чисел.

Указание. При расчете оценки СКО требуется перемножать температуры, но этой операции не реализовано. Можно поступить так:

- 1) сохранить шкалу X начального элемента выборки;
- 2) рассчитать оценку СКО по шкале X , но как действительное число;
- 3) результатом оценки СКО сделать температуру, величина которой рассчитана в шаге 2), а шкала известна из шага 1).

В шаге 2) необходимо вызывать одну из функций п. 4 в зависимости от X , — для этого удобно использовать функцию `get_value_of_scale()`, написанную в п. 4.1.

Контрольные вопросы и задания

- 1) Что такое функция в C++? Из каких элементов состоит определение функции?
- 2) Как выполняется возврат значения из функции? Как в C++ определить функцию, не возвращающую значение?
- 3) Опишите передачу параметров функции по ссылке и её использование.
- 4) Опишите передачу параметров функции по указателю и её использование.
- 5) Что такое рекурсия и каковы её виды? Чем отличается объявление функции от её определения, зачем они нужны?
- 6) Что такое указатель? Как объявляются и используются переменные-указатели?
- 7) Что такое указатель? Как соотносятся указатели и неизменяемость?
- 8) Что такое ссылка? Как объявляются и используются переменные-ссылки?
- 9) Что такое структуры в C++, как они объявляются и используются?
- 10) Что такое перечисления в C++, как они объявляются и используются?
- 11) Как организуется вывод в поток пользовательских типов данных?
- 12) Как организуется ввод из потока пользовательских типов данных?
- 13) Какими соображениями следует руководствоваться при выделении неоднократно используемых фрагментов программы в функции?
- 14) Каким образом функции могут использоваться для декомпозиции программы?