

Лабораторная работа № 3

Функции и пользовательские типы данных

Цель работы

1. Овладеть навыком применения функций и пользовательских типов для декомпозиции задачи и избежания дублирования кода программы.
2. Изучить средства форматного и файлового ввода-вывода в C++, а также возможности перегрузки операторов для повышения выразительности кода.
3. Опробовать различные подходы к обработке ошибок в программах.
4. Освоить физическое разделение кода программы в целях упрощения работы при большом объеме и сложной структуре программы.

Подготовка к лабораторной работе

Необходимо иметь готовую программу, написанную в ходе ЛР № 1, которая принимает на вход последовательность температур с указанием шкалы для каждой, а по завершении ввода печатающую таблицу тех же температур по шкалам Кельвина, Цельсия и Фаренгейта.

Задание на лабораторную работу

Указание. В отчете необходимо отражать каждый пункт, но достаточно вносить только код, который был добавлен или изменен (и при необходимости — контекст).

1. В программе из ЛР № 1 выделить код для преобразования температуры между шкалами в функцию

```
double convert(double temperature, char from, char to);
```

При некорректных значениях аргументов печатать из функции предупреждения.

2. Завести перечисление `Error` для кодов ошибок, которые могут возникать в функции `convert()`, чтобы использовать его в пунктах 3 и 3.
3. Реализовать обработку ошибок вовне `convert()` по кодам возврата.
4. Реализовать обработку ошибок вовне `convert()` через код последней ошибки.
5. Реализовать обработку ошибок вовне функции через механизм исключительных ситуаций. Типы исключений выбрать самостоятельно из числа стандартных, при этом:
 - 1) использовать не менее двух типов исключений;
 - 2) предусмотреть в обрабатываемом коде, что в будущем функция сможет возбуждать иные типы исключений.

Указание. Обработка в `convert()` означает, что на вызывающей стороне, то есть в функции `main()`, необходимо предусмотреть проверку, что функция `convert()` завершилась неудачно, и напечатать сообщение об этом.

6. Создать (изменить) проект так, чтобы в нем были:

- 1) основной файл исходного кода с функцией `main()`;
- 2) заголовочный файл с объявлением функции `convert()`;
- 3) файл реализации с определением функции `convert()`.

Указание. В отчет внести структуру проекта и журнал сборки программы (Build Log).

7. Добавить в заголовочный файл определение перечисления `Scale`, заменить тип параметров `to` и `from` функции `convert()` на `Scale`.

8. Добавить возбуждение исключения в функции `convert()` в том случае, если алгоритма преобразования между заданными шкалами в ней не реализовано.

Указание. В отчете объяснить, какие потенциальные проблемы этим решаются.

9. Добавить в заголовочный файл определение структуры `Temperature`, представляющей температуру со шкалой, с конструктором

`Temperature(double value, char scale).`

Реализацию конструктора поместить в отдельный файл исходного кода в проекте, куда впредь помещать реализации всех функций, относящиеся к `Temperature` (объявления же помещать в заголовочный файл).

10. Модернизировать основную программу: использовать `Temperature` вместо отдельных переменных для значения и шкалы, а также для хранения в векторе.

11. Перегрузить операторы ввода и вывода в поток для перечисления `Scale` и для структуры `Temperature` (итого четыре новые функции). Задействовать их в основной программе.

12. Перегрузить оператор сравнения для двух температур (`<`), сложения двух температур (`+`), деления температуры на число (`/`) и соответствующие им операторы сокращенного присваивания (`+=`, `/=`). Задействовать их для того, чтобы по окончании ввода данных печатать наибольшую, наименьшую и среднюю температуру.

13. Выделить вычисление наибольшей, наименьшей и средней температур в функцию. Типы и способы передачи параметров выбрать самостоятельно.

14. Добавить сохранение всех введенных температур в текстовый файл перед окончанием работы программы.

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?
15. В чем заключается подходы к обработке ошибок в функциях через коды возврата и через коды последней ошибки, каковы недостатки этих методов?
16. В чем заключается использование механизма исключительных ситуаций для обработки ошибок, каковы его преимущества?
17. Каков корректный порядок расположения блоков перехвата исключений при использовании иерархии исключений?
18. Из каких двух видов файлов с исходным кодом состоит программа на C++, для чего служит каждый из видов?