

Sem 1.01.2 Линейная программа

Презентация к семинару в рамках курса
«Программирование на Си»
Подготовил Кострицкий А. С.

Кафедра ИУ-7
МГТУ имени Н. Э. Баумана
Москва, Россия

Москва — 2023 — TS2302162105

Определение

Будем называть **препроцессором** программу, осуществляющую **препроцессинг** — комплекс действий над текстом на языке высокого уровня перед основной трансляцией.

Определение

Будем называть **транслятором** программу-переводчик с языка высокого уровня на язык машины.

Определение

Будем называть **компоновщиком** программу, преобразующую набор оттранслированных модулей в единую программу.

К сожалению, теоретическая база, необходимая для написания даже простейшей программы с абсолютным пониманием написанного, достаточно велика.

Сразу решим одну из простейших задач, по ходу дела рассматривая некоторые компоненты программы, не углубляясь в теоретический материал.

Пусть требуется, приняв от пользователя величины a и b равнобедренной трапеции и угол φ при втором основании, найти и вывести на экран площадь трапеции.

Как же решать эту задачу?

- 1 Выписать входные данные. *Какие данные требуется ввести с клавиатуры?*

Как же решать эту задачу?

- 1 Выписать входные данные. *Какие данные требуется ввести с клавиатуры?*

В нашем случае это a , b , φ .

Как же решать эту задачу?

- 1 Выписать входные данные. *Какие данные требуется ввести с клавиатуры?*
В нашем случае это a , b , φ .
- 2 Выписать выходные данные. *Какие данные требуется вернуть явно или неявно?*

Как же решать эту задачу?

- 1 Выписать входные данные. *Какие данные требуется ввести с клавиатуры?*

В нашем случае это a , b , φ .

- 2 Выписать выходные данные. *Какие данные требуется вернуть явно или неявно?*

В нашем случае это S — площадь трапеции и, вообще говоря, ещё выходным можно считать код возврата из приложения, несущий информацию об успешности выполнения действий. Но этот пункт мы всегда будем считать очевидным.

Как же решать эту задачу?

- 3 Установить связь между входными и выходными данными, иначе говоря, решить задачу *аналитически*. Это очень важный шаг, про который зачастую забывают — кодируется не сама задача, а её решение, и при должных аккуратности и внимательности кодирование занимает много меньше половины потраченного на реализацию проекта времени. Каково аналитическое решение нашей задачи?

Как же решать эту задачу?

- 3 Установить связь между входными и выходными данными, иначе говоря, решить задачу *аналитически*. Это очень важный шаг, про который зачастую забывают — кодируется не сама задача, а её решение, и при должных аккуратности и внимательности кодирование занимает много меньше половины потраченного на реализацию проекта времени. Каково аналитическое решение нашей задачи?

$$S = \frac{1}{4}(b - a)(b + a) \tan(\varphi).$$

- 4 Установить валидность связей и ограничения на диапазоны данных. Каковы они для нашей задачи?

- 4 Установить валидность связей и ограничения на диапазоны данных. Каковы они для нашей задачи?

$$a \in \mathbb{R}, a > 0(\text{geometry}), b \in \mathbb{R}, b > 0(\text{geometry}),$$

Плюс ограничение на случай прямого угла (исключение квадрата).

- 4 Установить валидность связей и ограничения на диапазоны данных. Каковы они для нашей задачи?

$$a \in \mathbb{R}, a > 0(\text{geometry}), b \in \mathbb{R}, b > 0(\text{geometry}),$$

Плюс ограничение на случай прямого угла (исключение квадрата).

Ограничения на период не учитываются, чтобы расширить для пользователя валидный диапазон угла. Нам не так сложно — тангенс всё равно съест периоды, это гораздо проще, чем указывать в документации ограничение, хотя формально стоило бы.

- 4 Установить валидность связей и ограничения на диапазоны данных. Каковы они для нашей задачи?

$$a \in \mathbb{R}, a > 0(\text{geometry}), b \in \mathbb{R}, b > 0(\text{geometry}),$$

Плюс ограничение на случай прямого угла (исключение квадрата).

Ограничения на период не учитываются, чтобы расширить для пользователя валидный диапазон угла. Нам не так сложно — тангенс всё равно съест периоды, это гораздо проще, чем указывать в документации ограничение, хотя формально стоило бы.

Почему нет ограничения на соотношение величин оснований?
Можно ли у реальной трапеции, нарисованной на бумаге, получить отрицательную площадь?

Как же решать эту задачу?

- 5. Выбрать используемые типы данных.

Как же решать эту задачу?

- 5. Выбрать используемые типы данных.

Об алгоритме выбора типа данных для переменной мы поговорим позднее, тут укажем готовое решение: так как задаются физические величины, причём без указания штук/единиц/целых, то и типом переменных для хранения величин будет служить ЧПТ `double`.

Как же решать эту задачу?

- 5. Выбрать используемые типы данных.

Об алгоритме выбора типа данных для переменной мы поговорим позднее, тут укажем готовое решение: так как задаются физические величины, причём без указания штук/единиц/целых, то и типом переменных для хранения величин будет служить ЧПТ `double`.

- 6. Реализовать ввод и проверку валидности.
- 7. Реализовать расчёт.
- 8. Реализовать вывод результата.

Полное решение этой задачи находится в файле:

[./Listings/trapezoid.naive.c](#)

В нём не реализованы проверки, так как это первый семинар, и логические конструкции ещё не изучены. Для особо внимательных есть эталонная реализация с проверками:

[./Listings/trapezoid.good.c](#)

Определение

Будем называть **точкой входа** строку, с которой начинается исполнение программы, будь она интерпретируемой, а не компилируемой.

В Си точка входа у программы единственна.

Определение

Будем называть **точкой выхода** строку, после которой исполнение программы прекращается.

В Си точек выхода у приложения может быть несколько.

Создание приложения:

```
gcc -o app.exe trapezoid.good.c -Werror \
-Wall -Wextra -Wpedantic
```

или, по шагам:

```
gcc -c trapezoid.good.c -Werror \
-Wall -Wextra -Wpedantic
gcc -o app.exe trapezoid.good.o
```

Где тут трансляция, а где — компоновка? Где препроцессинг? Где
объектный модуль, а где — программа?

Вопросы для самопроверки I

- 1 Что указывает препроцессору директива `include`? Можно ли говорить, что она *подключает* внешний файл?
- 2 Что указывает препроцессору директива `define` с одним «аргументом»? С двумя?
- 3 Обрабатывается ли макрос внутри строчного литерала?
- 4 Директивы препроцессора доходят ли до транслятора? Иначе говоря, получает ли в тексте программы транслятор хоть один октоторп? А вне строчных литералов?
- 5 Что Вы назвали бы точкой входа в приложение? Может ли у приложения не быть точки входа? Быть несколько?

Вопросы для самопроверки II

- 6 Что Вы назвали бы точкой выхода из приложения? Может ли у приложения не быть точки выхода? Быть несколько?
- 7 Почему константа задаётся в решении не иначе, как `#define PI (4.0 * atan(1.0))`? Иначе говоря, можно ли использовать в качестве константы π числовое значение 3.14159265369?