



МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«МИРЭА – Российский технологический университет»

РТУ МИРЭА

Институт комплексной безопасности и специального приборостроения
Кафедра КБ-2 «Прикладные информационные технологии»

А.А. МЕРСОВ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ
ПРАКТИЧЕСКОЙ РАБОТЫ №1

“Структура программы на языке программирования Си. Типы данных языка
программирования и функции ввода вывода”

по дисциплине: **«Языки программирования»**
(наименование дисциплины)

Москва – 2020

УДК
ББК

Печатается по решению редакционно-издательского совета «МИРЭА – Российский технологический университет»

Мерсов А.А.

Методические указания по выполнению практической работы № 1 по языкам программирования / А.А. Мерсов– М.: МИРЭА – Российский технологический университет, 2020.

Методические указания предназначены для выполнения практической работы по дисциплине «Языки программирования» и содержит перечень вариантов практической работы, а также краткое изложение теоретического материала в форме пояснений к заданию на работу. Для студентов, обучающихся по направлениям 09.03.02, 10.03.01, 10.05.02, 10.05.03, 10.05.04.

Материалы рассмотрены на заседании учебно-методической комиссии

КБ-2 Протокол №1 от «28» августа 2020 г.

и одобрены на заседании кафедры КБ-2.

зав. кафедрой КБ-2

к.т.н.

/ О.В.Трубиенко /

УДК ББК

© Мерсов А.А., 2020

© Российский технологический университет – МИРЭА, 2020

| Содержание | |
|---|----|
| Общие указания к выполнению практической работы | 4 |
| Цель практической работы | 4 |
| Основные сведения из языков программирования | 5 |
| Структура программы | 6 |
| Типы данных языка программирования Си | 7 |
| Функции ввода вывода | 8 |
| Варианты заданий | 10 |
| Пример выполнения | 11 |

Общие указания к выполнению практической работы

Практические работы выполняются с использованием персональных компьютеров. Указания по технике безопасности совпадают с требованиями, предъявляемыми к пользователю ЭВМ. Другие опасные факторы отсутствуют.

Цель практической работы

Цель практической работы по дисциплине «Языки программирования» состоит в закреплении и углублении знаний и навыков, полученных при изучении дисциплины. Практическая работа предполагает выполнение задания разработке и тестированию программного обеспечения.

Основные сведения из языков программирования

Структура программы

Все программы на языке СИ содержат директивы препроцессора, описания, определения, выражения, операторы и функции.

-Директива препроцессора.

Директива препроцессора-это команда препроцессора языка СИ (который автоматически вызывается на первом шаге компиляции программы). Две наиболее общих директивы препроцессора-это директива `#define`, которая подставляет текст вместо заданного идентификатора, и директива `#include`, которая включает в программу текст внешнего файла.

-Описания.

Описание устанавливает имена и атрибуты переменных, функций и типов, используемых в программе. Глобальные переменные описываются вне функций и действуют от конца описания до конца файла. Локальная переменная описывается внутри функции и действует от конца описания до конца функции.

-Определение.

Определение устанавливает содержимое переменной или функции. Определение также отводит память, требуемую для переменных и функций.

-Выражение.

Выражение-это комбинация операций и операндов, которое принимает единственное значение.

-Оператор.

Операторы управляют порядком выполнения программы на языке СИ.

-Функция.

Функция-это набор описаний, определений, выражений и операторов, которое выполняет определенную задачу. Тело функции заключено в фигурные скобки. В языке СИ функции могут быть вложены друг в друга.

-Функция `main` (главная).

Все программы на языке СИ, там, где начинается выполнение программы, имеют функцию с именем `main`. Фигурные скобки, в которые заключено тело функции `main`, определяют начало и конец программы.

При разработке программ на языке программирования Си придерживаются следующих правил:

В первой строке записывает комментарий, поясняющий условие решаемой задачи.

Во второй строке помещается команда (директива) препроцессора, обеспечивающая включение в программу средств связи со стандартными потоками ввода и вывода данных. Указанные средства находятся в файле с именем `iostream.h` – заголовочный файл.

Стандартный поток `cin` обеспечивает считывание символов с клавиатуры и преобразование их в соответствующие числовые значения переменных.

Стандартный поток вывод `cout` обеспечивает вывод данных на экран дисплея.

Препроцессор – это программа, действующая как фильтр на этапе компиляции. Перед тем как попасть на вход компилятора, исходная программа проходит через препроцессор.

`# include <заголовочный файл>` (файлы заголовков) ; `#` - директива препроцессора.
`#include<iostream.h>`

Файлы заголовков в директиве `include` заканчиваются на `.h`. Угловые скобки `<>` указывают препроцессору, что этот файл ищется в стандартном каталоге подключаемых файлов.

Пример заголовочных файлов:

`math.h` – подключение математических функций;

`string.h` – подключение строковых функций.

`stdio.h` – функции форматированного ввода-вывода.

`conio.h` – работа с консолью (экраном и клавиатурой).

`ctype.h` – функции преобразования символов.

`stdio.h` – подключение функций ввода-вывода

Примечание: в конце директивы `;` не ставится.

Объявление пользовательских констант и глобальных переменных. Например, `#define PI 3.14`

Объявление функции и/или их прототипов.

Каждая программа на языке C++ должна иметь в своем составе функцию с именем `main()`, с которой начинается выполнение программы. Поэтому эта функция называется главной и имеет соответствующее фиксированное имя. По умолчанию функции `main` имеет тип `int`. Это означает, что данная функция возвращает в место вызова целочисленное значение. Местом вызова главной функции является среда выполнения или среда операционной системы. Принято соглашение, что любая программа при аварийном завершении должна возвращать в операционную систему ненулевое значение. При правильном выполнении программы передаётся нулевой результат. В конце функции `main()` всегда стоит оператор `return 0`.

```
int main ()
{тело функции (основная программа);
return 0;
}
```

Правила объявления функции `main()`.

После заголовка функции никаких знаков препинания не ставятся.

Тело функции включает:

- объявление локальных переменных и констант и их инициализацию (присвоение начальных значений);
- ввод исходных данных (диалог с пользователями);
- обработку – обращение к функциям вычисления выражений;
- выполнение операторов;
- вывод результата;
- возвращение кода завершения главной функции (`return 0`);

Общие принципы, позволяющие написать синтаксически правильную программу, таковы:

- прежде чем использовать функции, переменные, типы данных, следует объявить их или подключить файлы с их объявлениями (сделать известными компилятору);
- чтобы вызов функции мог быть выполнен, функция должна быть определена, т.е. описаны действия, которые она осуществляет;
- в любой последовательности действий нужно стремиться к триаде: инициализация (ввод), обработка, возвращение значения (вывод).

Правила оформления текста программы, направленные на облегчение понимания смысла и повышение наглядности, таковы:

- разделять логические части программы пустыми строками;
- разделять операнды и операции пробелами;
- для каждой фигурной скобки отводить отдельную строку;

- в каждой строке должно быть, как правило, не более одного оператора;
- ограничивать длину строки 60-70 символами;
- отступами слева отражать вложенность операторов и блоков;
- длинные операторы располагать в нескольких строках;

проводить алгоритмизацию так, чтобы определение одной функции занимало, как правило, не более одного экрана текста;

стремиться использовать типовые заготовки фрагментов программ, включая и типовую структуру блока и программы в целом.

Пример: Вычисление площади круга

```
// вычисление площади круга
#include<iostream.h>
#define PI 3.14
int main()
{float r, pl;
printf("Ввести радиус\n"); // вывод на экран приглашения о вводе радиуса
scanf("%f", &r); // ввод радиуса
pl=r*r*PI; // вычисление площади по формуле
printf ("Площадь круга =%f\n", pl); // вывод на экран результата
return 0;
}
```

Типы данных языка программирования Си

Переменная – ячейка памяти. Назначение типа определяет размер этой ячейки, какие операции можно выполнять с данной переменной, допустимые значения, которые может принимать переменная.

Классификация типов данных:

- Простые (скалярные) – целые, вещественные, логические (булевские).
- Структурированные – массивы, структуры, объединения, классы.
- Адресные – указатели, ссылки.

Простые типы данных

Целый тип

char – целый символьный тип;

short int – короткий целый тип;

int – целый тип;

long int – длинный целый тип.

Каждый из целочисленных типов может быть определён как знаковый (signed) или беззнаковый (unsigned) тип. При отсутствии ключевого слова, по умолчанию принимается знаковый тип.

Вещественный тип

float – вещественный тип одинарной точности;

double – вещественный тип двойной точности;

long double – вещественный тип расширенной точности;

Логический тип

bool – логический тип данных, принимает значение «истина» (true или 1), или «ложь» (false или 0).

| Тип данных | Размер, байт | Диапазон значений |
|------------|--------------|-------------------|
| char | 1 | -128... +128 |

| | | |
|-------------------|-----|-------------------------------|
| int | 2/4 | зависит от системы |
| unsigned char | 1 | 0... 255 |
| unsigned int | 2/4 | зависит от системы |
| short int | 2 | -32768... 32767 |
| unsigned short | 2 | 0... 65535 |
| long int | 4 | -2147483648... 2147483648 |
| unsigned long int | 4 | 0... 4294967295 |
| float | 4 | $\pm(3.4E-38... 3.4E+38)$ |
| double | 8 | $\pm(1.7E-308... 1.7E+308)$ |
| long double | 10 | $\pm(3.4E-4932... 1.1E+4932)$ |

Анализируя данные таблицы, можно сделать следующие выводы:
если не указан базовый тип, то по умолчанию подразумевается int;
если не указан модификатор знаков, то по умолчанию подразумевается signed;
с базовым типом float модификаторы не употребляются;
модификатор short применим только к базовому типу int.

Приведение типов (явное и неявное)

Приведением типов называется преобразование значения одного типа к значению другого типа, при этом возможно, как сохранение величины этого значения, так и изменение этой величины. Приведение типов также называют преобразованием типов.

При делении двух переменных целого типа результат получится целый. Для получения вещественного результата, необходимо явное преобразование типа.

```
int a=5; int b=2; float t;
```

```
t=a/b (при этом t будет равняться 2)
```

$t = (\text{float})a/b$ (при этом t будет равняться 2,5). На момент вычисления изменили тип переменной, а – это и есть явное приведение типов. Приведённый тип указывается в круглых скобках перед переменной.

Неявное преобразование типов используется, когда в бинарной операции операнды имеют различные типы. В этом случае компилятор выполняет преобразование типов автоматически, т.е. все они приводятся к типу с наибольшим диапазоном значений.

```
int a=5; float b=2; float t;
```

```
t=a/b (результат = 2,5)
```

При неявном преобразовании типов выполняется правило переводов низших типов в высшие для точности представления данных и их непротиворечивости (низший тип – char, высший - long double; см. таблицу).

Функции ввода вывода

Функции стандартного ввода - вывода описаны в файле stdio.h.

printf() - форматный вывод на экран:

```
int printf(char *format, <список вывода>);
```

Первый параметр является символьной строкой, которая задает спецификации формата. Остальные параметры - перечисление переменных и выражений, значения которых выводятся. Каждая спецификация формата имеет вид (параметры в квадратных скобках необязательны):

```
%[flags][width][.prec][F|N|h|l]type
```

| | |
|------|------------------|
| type | тип спецификации |
| - | |

| | |
|-----------------------------------|--|
| d или i | целое десятичное число со знаком |
| u | десятичное число без знака |
| x | целое 16-ричное число без знака |
| f | число с плавающей точкой |
| e | число в E-форме |
| g | число с плавающей точкой или в E-форме |
| c | один символ |
| s | строка |
| % | символ % |
| flags - + или пусто - | признак выравнивания: выравнивание по правому краю выравнивание по левому краю |
| width - | целое число - общая ширина поля. Если это число начинается с цифры 0, вывод дополняется слева нулями до заданной ширины. В заданную ширину входят все символы вывода, включая знак, дробную часть и т.п. |
| prec - | целое число, количество знаков после точки при выводе чисел с плавающей точкой |
| F - | соответственный элемент списка вывода является дальним указателем |
| N - | соответственный элемент списка вывода является близким указателем |
| l - | соответствующий элемент списка вывода является long int или double |

scanf() - форматный ввод с клавиатуры:

```
int scanf(char *format, <список ввода>);
```

Первый параметр является символьной строкой, которая задает спецификации формата (см. функцию printf()). Остальные параметры - перечисление адресов переменных, в которые вводятся данные. В этом списке перед именами всех переменных, кроме тех, которые вводятся по спецификации типа %, должен стоять символ &.

putchar () - вывод одного символа на экран:

```
int putchar (int ch);
```

Параметр функции - код символа, который выводится. При успешном выполнении функция возвращает этот же код, при неуспешном - EOF.

getchar () - ввод одного символа с клавиатуры:

```
int getchar(void);
```

Функция возвращает код введенного символа.

puts() - вывод строки символов на экран:

```
int puts(char *string);
```

Параметр функции - указатель на начало той строки, из которой выводятся данные. Функция возвращает количество выведенных символов.

gets () - ввод строки символов с клавиатуры:

```
char *gets (char *string);
```

Параметр функции - указатель на начало той строки, в которую вводятся данные. Функция возвращает тот же самый указатель.

Варианты заданий

Разработать программу, которая вводит фактические данные из таблицы, представленной в Вашем варианте индивидуального задания и выводит на экран таблицу, подобную той, которая находится в индивидуальном задании (включая заголовок и примечания).

Вариант 0

| Отдел кадров | | | |
|---|----------|----------|--------|
| Фамилия | Инициалы | Год рожд | Оклад |
| Иванов | И.И. | 1975 | 517.50 |
| Петренко | П.П. | 1956 | 219.10 |
| Панковский | М.С. | 1967 | 300.00 |
| Примечание: оклад установлен по состоянию на 1 января 2000 года | | | |

Вариант 1

| Фирмы - производители СКБД | | | |
|--|-----------------------|----------------------------|-----------------|
| Фирма | Количество. продуктов | Годовой объем продажи (\$) | Часть рынка (%) |
| Oracle | 1 | 2488000000 | 31.1 |
| IBM | 3 | 2392000000 | 29.9 |
| Microsoft | 2 | 1048000000 | 13.1 |
| Примечание: по данным Gartner Group за 1999г | | | |

Вариант 2

| Характеристики ПЭВМ | | | |
|---|---------------|----------|-----|
| Процессор | Частота (Mgz) | RAM (Mb) | Тип |
| Pentium-III | 233 | | C |
| AMD-K6 | 166 | | C |
| PowerPC-620 | 2000 | | R |
| Тип: C - CISC-процессор, R - RISC-процессор | | | |

Вариант 3

| Каталог библиотеки | | | |
|--|----------|-------------|--------|
| Автор книги | Название | Год выпуска | Группа |
| Сенкевич | Потоп | 1978 | X |
| Ландау | Механика | 1989 | Y |
| Дойль | Сумчатые | 1990 | C |
| Примечание: X - художественная литература; Y - учебная литература; C - справочная литература | | | |

Вариант 4

| Офисные пакеты | | | |
|---|---------------|-------------------------|-----------|
| Наименование | Производитель | Количество сост. частей | Цена (\$) |
| Office | Microsoft | 4 | 870 |
| SmartSuite | Lotus | 5 | 1020 |
| StarOffice | Sun | 4 | 9 |
| Примечание: возможно бесплатно получить продукт StarOffice через Internet | | | |

Вариант 5

| Сельскохозяйственные культуры | | | |
|---------------------------------------|-----|-----------------------|--------------------|
| Наименование | Тип | Посевная площадь (га) | Урожайность (ц/га) |
| Соя | Б | 13000 | 45 |
| Чумиза | З | 8000 | 17 |
| Рис | З | 25650 | 24 |
| Примечание: З - зерновые, Б - бобовые | | | |

Вариант 6

| Ведомость спортивных состязаний |
|---------------------------------|
|---------------------------------|

| Фамилия участника | Код команды | Количество баллов | Место в итоге |
|---|-------------|-------------------|---------------|
| Баландин | С | 123.7 | 2 |
| Шишков | Ш | 79.98 | 3 |
| Кравченко | Д | 134.8 | 1 |
| Примечание: Д - "Динамо", С - "Спартак", Ш - "Шахтер" | | | |

Вариант 7

| Ведомость общественного транспорта | | | |
|--|----------|-----------------------------|----------------------|
| Вид транспорта | маршрута | Протяженность маршрута (км) | Время в дороге (мин) |
| Тр. | 12 | 27.55 | 75 |
| Т-с | 17 | 13.6 | 57 |
| А | 12а | 57.3 | 117 |
| Примечание: Тр - трамвай, Тс - троллейбус, А - автобус | | | |

Вариант 8

| Сельскохозяйственные предприятия | | | |
|--|-------------------|--------------------|-----------------|
| Название | Вид собственности | Площадь земли (га) | Кол. работников |
| Заря | Д | 300 | 120 |
| Росинка | К | 174 | 27 |
| Петренко | Ч | 56 | 6 |
| Вид собственности: Д - государственная, Ч - частная, К - кооперативная | | | |

Вариант 9

| Прайс-лист | | | |
|--|------------|--------------------|---------------------------------|
| Наименование товара | Тип товара | Цена за 1 шт (грн) | Минимальное количество в партии |
| Папка | К | 4.75 | 4 |
| Бумага | К | 13.90 | 10 |
| Калькулятор | О | 411.00 | 1 |
| Примечание: К – канцтовары, О - оргтехника | | | |

Вариант 00

| Буддийские монастыри Японии периода Нара | | | |
|--|-------|--------------------|--------------------|
| Название | Школа | Количество монахов | Площадь земли (га) |
| Тодайдзи | Т | 220 | 368.8 |
| Якусидзи | С | 50 | 54.7 |
| Дайандзи | Д | 10 | 12.2 |
| Примечание: Т - Тэндай; С - Сингон; Д – Дзедзицу | | | |

Примеры решения задач (вариант 00)

Разработка алгоритма решения

Общий метод решения

В таблице, приведенной в индивидуальном задании, первая строка - общий заголовок таблицы, вторая - заголовки столбцов таблицы. Фактические данные составляют третью, четвертую и пятую строки. Шестая строка - примечания.

Программа должна начинаться с ввода фактических данных. Для удобства оператора перед вводом на экран должно выводиться приглашение, в котором перечисляется, какие именно данные ожидает программа. За одно приглашение-ввод будет вводиться одна строка фактических данных таблицы. Т.е., приглашение-ввода должно повторяться трижды. Введенные данные будут сохраняться в переменных программы.

На втором этапе программа выполняет вывод данных на экран. Каждая строка таблицы может выводиться одним оператором вывода. Среди строк, которые выводятся, будут строки без переменных значений (заголовки и линии) и строки с переменными значениями (строки с фактическими данными).

Для формирования наглядного табличного вида можно применять символы '|' и '-|'.

Поскольку алгоритм программы чисто линейный, схему алгоритма мы не приводим

Определение переменных программы

Определение переменных является первой и самой главной задачей нашей программы. Для каждой переменной нужно подобрать такой тип, который бы обеспечивал представление в памяти ее значения с достаточным диапазоном и достаточной точностью.

Название. Во всех фактических значениях этот столбец является символьной строкой из 8 символов. Т.е. для представления его в памяти переменные, в которых хранятся значения этого столбца, должны быть объявлены как:
`char name1[9], name2[9], name3[9];`

дополнительный символ резервируется для обозначения конца символьной строки.

Школа. Эти данные являются одним символом. Их объявление:
`char sch1, sch2, sch3;`

Количество монахов. Эти данные являются целыми числами в диапазоне 10 - 220. Этот диапазон перекрывается диапазоном возможных значений типа ***unsigned short*** (диапазон этого типа: 0 - 255). Т.е. объявление соответствующих переменных:
`unsigned short cnt1, cnt2, cnt3;`

Площадь земель. Это число с дробной частью, следовательно, соответствующие переменные должны быть переменными с плавающей точкой. Точность данных - всего один знак после точки, поэтому для их представления будет достаточно типа ***float***:
`float sq1, sq2, sq3;`

Разработка текста программы

Начинаем разработку текста программы с заголовка главной функции ***main()***:
`int main(void)`

Далее открывается тело функции и в нее включается описание переменных (см. п.5.2).

Кодовая часть программы начинается с приглашения - вывода строки-константы:
`printf("1. Введите: название, школу, количество, площадь >");`

за которым считываются данные, вводимые оператором:

`scanf("%s %c %d %f", name1, &sch1, &cnt1, &sq1);`

Поскольку описания этих функций хранятся в файле ***stdio.h***, включаем этот файл в начало программы:

`#include <stdio.h>`

При вводе строка вводится по спецификации типа ***%s***, один символ - по спецификации типа ***%c***, целое число - по спецификации типа ***%d***, число с плавающей точкой - по спецификации типа ***%f***. Спецификации разделяются пробелами, т.е. и данные при вводе должны разделяться пробелами или переходом на новую строку. В списке ввода перед всеми элементами кроме того, который вводится по ***%s***, ставится знак ***&***.

Приглашение-ввод повторяется трижды, с разными переменными в списке ввода.

Для формирования вывода следует подсчитать ширину каждого столбца. Ширина первого столбца определяется размером фактических данных в строке - 9 символьных мест, с учетом пробелов в начале и в конце - 11. Ширина остальных столбцов определяется шириной текста в заголовках столбцов и составляет соответственно 7, 11 и 13 символьных мест. Учитывая вертикальные линии ширина строки составляет 47 знакомест.

Первая строка - горизонтальная линия, которая состоит из символа '-', повторенного 47 раз. Вторая строка - общий заголовок, дополненный до ширины 47 пробелами. Третья строка - еще одна горизонтальная линия. Четвертая и пятая строки - заголовки столбцов, каждый из них имеет установленную ширину. Шестая строка - еще одна горизонтальная линия.

Наконец, строки с седьмой до девятой - фактические данные. В каждой строке выводятся значения набора переменных для одной строки. Например,:
`printf("| %9s | %c | %-3d | %5.1f |\n", name1, sch1, cnt1, sq1);`

Значения имени выводятся по спецификации типа `%s` с шириной 9. Значения школы выводятся по спецификации типа `%s`. Поскольку значения в этом столбце выводятся по центру столбца, перед и после него ставятся пробелы. Значения количества выводятся по спецификации типа `%d` с шириной 3, а площадь - `%f` с общей шириной 5 и одним знаком после точки. Значения в этих столбцах дополняются до нужной ширины пробелами. В столбце 3 ставится признак выравнивания по правому краю.

Остальные четыре строки выводятся как текстовые константы.

Отладка программы

При отладке программы можно использовать пошаговый режим отладки с отслеживанием значений переменных - тех, которые вводятся. Если возникнут проблемы с вводом переменных, есть смысл вводить каждую переменную отдельным оператором. Аккуратный формат таблицы достигается несколькими запусками программы с последующим выравниванием столбцов по результатам запуска.

Результаты работы программы

При работе программы на экран было выдано следующее:

1. Введите: название, школу, количество, площадь > Тодайдзи Т 220 368.8
2. Введите: название, школу, количество, площадь > Якусидзи С 50 54.7
3. Введите: название, школу, количество, площадь > Дайаедзи Д 10 12.2

| | | | |
|---|----------|-----------------------|--------------------|
| Буддийские монастыри Японии периода Нара | | | |
| Название----- | Школа -- | Количество монахов -- | Площадь земли (га) |
| Тодайдзи | Т | 220 | 368.8 |
| Якусидзи | С | 50 | 54.7 |
| Дайаедзи | Д | 10 | 12.2 |
| Примечание: Т - Тендай; С - Сингон; Д - Дзедзицу | | | |