

Технология программирования на ЭВМ

Структуры

Баев А.Ж.

Казахстанский филиал МГУ

12 декабря 2018

Пример: время

Цель: описать момент времени 12:10:05.

```
1 int hour= 12;  
2 int minute = 10;  
3 int second = 5;
```

Неудобно: инициализировать, копировать, передавать как аргументы функции, отсутствие единого имени, множество таких объектов.

Пример: время

```
1 struct Time {  
2     int hour;  
3     int minute;  
4     int second;  
5 };  
6 int main() {  
7     struct Time t;  
8     t.hour = 12;  
9     t.second = 10;  
10    t.minute = 5;  
11    ...  
12 }
```

Пример: время

Инициализация

```
1 struct Time {  
2     int hour;  
3     int minute;  
4     int second;  
5 };  
6 int main() {  
7     struct Time t = {12, 10, 5};  
8     ...  
9 }
```

Пример: время

Копирование

```
1 struct Time {
2     int hour;
3     int minute;
4     int second;
5 };
6 int main() {
7     struct Time t1 = {12, 10, 5};
8     struct Time t2;
9     t1 = t2;
10    ...
11 }
```

Трёхмерный вектор

```
1 struct Vector3D {  
2     double x, y, z;  
3 };  
4 int main() {  
5     struct Vector3D v = {1.0, 2.0, 3.0};  
6     ...  
7 }
```

Комплексное число

```
1 struct Complex {
2     double re, im;
3 };
4 int main() {
5     struct Complex c;
6     c.re = 1.0;
7     c.im = 2.0;
8     ...
9 }
```

Пример: студент

```
1 struct Student {
2     char name[10];
3     char surname[10];
4     int age;
5 };
6 int main() {
7     struct Student s = {"Bob", "Sponge", 10};
8     s.age = 12;
9     s.name[0] = 'R';
10    puts(s.name);
11    strcpy(s.name, s.surname);
12    ...
13 }
```

Какой размер занимает переменная *student*?

Пример: треугольник

```
1 struct Point {
2     int x, y;
3 };
4 struct Triangle {
5     struct Point a, b, c;
6 };
7 int main() {
8     struct Triangle triangle;
9     struct Point d = {0, 0};
10    triangle.a = d;
11    triangle.b.x = 2;
12    triangle.b.y = 3;
13    ...
14 }
```

Какой размер занимает переменная *triangle*?

Алиас для названия

```
1 struct Vector3d {  
2     int x, y, z;  
3 };  
4 typedef struct Vector3d Vector;  
5 ...  
6 Vector a;
```

Аргументы передаются по значению, то есть создаётся локальная для функции переменная *v1*, в которую КОПИРУЮТСЯ значения.

```
1 double dot(struct Vector a, struct Vector b)
2     return a.x * b.x + a.y * b.y + a.z * b.z;
3 }
4 ...
5 Vector a = {1.0, 2.0, 3.0};
6 Vector b = {3.0, 1.0, 0.0};
7 double product = dot(a, b);
8 ...
```

```
1 Vector cross(Vector a, Vector b) {  
2     Vec3 result;  
3     result.x = a.y * b.z - a.z * b.y;  
4     result.y = a.z * b.x - a.x * b.z;  
5     result.z = a.x * b.y - a.y * b.x;  
6     return result;  
7 }  
8 ...  
9 Vector a = {1.0, 2.0, 3.0};  
10 Vector b = {3.0, 1.0, 0.0};  
11 Vector c;  
12 c = cross(a, b);  
13 ...
```

Передача по значению — не меняет значение исходной переменной.

```
1 void initzero(Vector a) {  
2     a.x = 0;  
3     a.y = 0;  
4     a.z = 0;  
5 }  
6 ...  
7 struct Vector a = {1.0, 2.0, 3.0};  
8 initzero(a);  
9 ...
```

Передача по указателю — меняет значение исходной переменной.

```
1 void initzero(Vector *a) {  
2     (*a).x = 0;  
3     (*a).y = 0;  
4     a->z = 0;  
5 }  
6 ...  
7 struct Vector a = {1.0, 2.0, 3.0};  
8 initzero(&a);  
9 ...
```

Дано время в формате hh:mm:ss. На следующей строке дано целое число d от 0 до $24 \cdot 60 \cdot 60$. Вывести отсечку времени, которое будет через d секунд (возможен переход на новые сутки).

```
struct Time add_time(struct Time current, int delta)
```

Ввод

```
12:10:00  
4801
```

Вывод

```
13:30:01
```

```
1 #include <stdio.h>
2 struct Time {
3     int hour, minute, second;
4 };
5
6 struct Time add_time(struct Time time,
7                     int delta) {
8     int absolute = time.second +
9                   time.minute * 60 +
10                  time.hour * 60 * 60;
11     absolute += delta;
12     time.second = absolute % 60;
13     absolute /= 60;
14     time.minute = absolute % 60;
15     absolute /= 60;
16     time.hour = absolute % 24;
17     return time;
18 }
```



```
1  int main() {
2      struct Time time, new_time;
3      int delta;
4      scanf("%d:%d:%d\n%d",
5            &time.hour,
6            &time.minute,
7            &time.second,
8            &delta);
9      new_time = add_time(time, delta);
10     printf("%02d:%02d:%02d\n",
11           new_time.hour,
12           new_time.minute,
13           new_time.second);
14     return 0;
15 }
```

Дано z (вещественная и мнимая часть числа z). Далее дано целое число n от 0 до 100. Вычислить z^n .

Ввод

```
2.00 1.00
```

```
4
```

Вывод

```
-7.00 24.00
```

```
4
```

```
1  #include <stdio.h>
2  struct Complex {
3      double re, im;
4  };
5
6  typedef struct Complex Comp;
7
8  Comp multiply(Comp a, Comp b) {
9      Comp c;
10     c.re = a.re * b.re - a.im * b.im;
11     c.im = a.re * b.im + a.im * b.re;
12     return c;
13 }
14 4
```

Комплексные числа

```
1  Comp power(Comp z, int n) {
2      int i;
3      Comp result = {1.0, 0.0};
4      for (i = 0; i < n; i++)
5          result = multiply(result, z);
6      return result;
7  }
8
9  int main() {
10     Comp z, ans;
11     int n;
12     scanf("%lf_%lf\n%d", &z.re, &z.im, &n);
13     answer = power(z, n);
14     printf("%.2lf_%.2lf\n", ans.re, ans.im);
15     return 0;
16 }
```