



Рисунок 1. Схема организации работы с памятью в C++

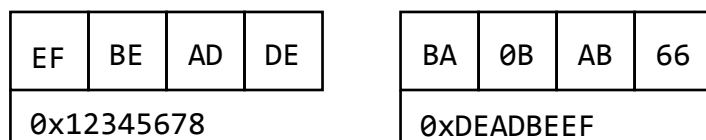


Рисунок 2. Пример расположения данных в памяти.

```

1      #include <cmath>
2      #include <stdio>
3
4      bool solveQuadricEquation(
5          double a, double b, double c, double& x1, double& x2)
6      {
7          double D = b*b - 4*a*c;
8          if (D < 0.0f) {
9              return false;
10         } else {
11             x1 = (-b + sqrt(D)) / 2*a;
12             x2 = (-b - sqrt(D)) / 2*a;
13             return true;
14         }
15     }
16
17     int main() {
18         double root1 = 0.0f, root2;
19         if (solveQuadricEquation(1.0f, 3.0f, 2.0f, root1, root2)) {
20             printf("Корни x^2 + 3*x + 2 = 0: %.2f и %.2f\n",
21                 root1, root2);
22         } else {
23             puts("Действительные корни уравнения не найдены!");
24         }
25         return 0;
26     }

```

**Листинг 1 – Пример использования ссылок**

```

1      float value = 10.0f;
2      float *address = &value;
3      float anotherValue = *address;
4      anotherValue += 20.0f;
5      *address = *address + 30.0f;
6      int wrongTypeValue = *address; // Не компилируется.

```

**Листинг 2 – Пример использования операций взятия адреса и разыменования указателя**

```

1      int array[10];
2      int *end = array + 10;
3      int arraySizeBySizeof = sizeof(array) / sizeof(array[0]);
4      int arraySizeByOffset = end - array;

```

**Листинг 3 – Пример объявления массива**

```

1 unsigned int source = 0xFEDCBA98;
2 int *destination = reinterpret_cast<int*>(&source);

```

**Листинг 4 – Пример использования reinterpret\_cast**

```

1 int length = 10;
2 int *dynamicArray = new int[length];
3 int *dynamicObject = new int;
4 delete [] dynamicArray;
5 delete dynamicObject;

```

**Листинг 5 – Пример использования операторов new, new[], delete и delete[]**

```

1 void* operator new (size_t size);
2 void operator delete (void* ptr) noexcept;
3 void* operator new[] (size_t size);
4 void operator delete[](void* ptr) noexcept;
5 void* operator new (size_t size, void* ptr) noexcept;
6 void* operator new[](size_t size, void* ptr) noexcept;

```

**Листинг 6 – Определение функций new и delete**

```

1 int pool[10];
2 int *item = new (pool) int;

```

**Листинг 7 – Пример использования placement new**

```

1 void* malloc(size_t size);
2 void* calloc(size_t size);
3 void* realloc(void* pointer, size_t size);
4 void free(void* pointer);

```

**Листинг 8 – Определение функций работы с памятью из библиотеки cstdlib**

```

1 size_t length = 10;
2 int *array = (int*)malloc(sizeof(int) * length);
3 free(array);

```

**Листинг 9 – Пример использования функций malloc и free**

```

1      #include <cstring>
2      ...
3      if (strlen(first) != strlen(second)) {
4          puts("Длины строк не равны.");
5      } else {
6          printf("Длины строк равны, ");
7          if (strcmp(fisrt, second) == 0)
8              puts("и сами строки равны.");
9          else
10             puts("а сами строки не равны.");
11     }

```

**Листинг 10 – Использование функций из библиотеки `cstring`**