

Помните! Код чаще читается, чем пишется, поэтому не экономьте на понятности и чистоте кода ради скорости набора.

1. Имена представляющие типы должны начинаться с большой буквы в смешанном стиле

Line, SavingsAccount

2. Имена переменных должны начинаться с маленькой буквы в смешанном стиле

line, savingsAccount

3. Имена констант должны быть в верхнем регистре с использованием подчеркивания для разделения слов

MAX_ITERATIONS, COLOR_RED, PI

4. Имена функций должны состоять из глагола и начинаться с маленькой буквы в смешанном стиле

getName(), computeTotalWidth()

5. Все имена должны быть на английском (международная разработка)

fileName; // NOT: *filNavn*

6. Значимые переменные должны иметь длинные имена, малозначащие короткие.

7. Индексные переменные должны называться *i, j, k*

```
for (int i = 0; i < nTables); i++) {  
:  
}
```

9. Префикс *is* используется для булевых переменных

isSet, isVisible, isFinished, isFound, isOpen

10. Избегайте аббревиатур

computeAverage(); // NOT: *compAvg()*

11. Разбиение строк должно быть очевидным

```
totalSum = a + b + c +  
          d + e;  
function (param1, param2,  
          param3);  
for (int tableNo = 0; tableNo < nTables;  
     tableNo += tableStep) {  
  ...  
}
```

12. Переменные должны инициализироваться сразу после их объявления

```
int x, y, z;  
getCenter(&x, &y, &z);
```

13. Указатели пишутся рядом с типом, а не именем

```
float* x;       // NOT: float *x;  
int& y;        // NOT: int &y;
```

14. Только одна переменная должна контролировать цикл

```
sum = 0;          // NOT: for (i = 0, sum = 0; i < 100; i++)
for (i = 0; i < 100; i++)      sum += value[i];
sum += value[i];
```

15. Для бесконечного цикла применяется следующая конструкция.

```
while (true) { : }
for (;;) { // NO! : }
while (1) { // NO! : }
```

16. Нельзя использовать goto

17. Базовый отступ должен быть 2 символа

```
for (i = 0; i < nElements; i++)
    a[i] = 0;
```

18. Блоки циклов

```
while (!done) {
    doSomething();
    done = moreToDo();
}
while (!done)
{
    doSomething();
    done = moreToDo();
}
while (!done)
{
    doSomething();
    done = moreToDo();
}
```

19. Блоки условий

```
if (condition) {
    statements;
}
if (condition) {
    statements;
}
else {
    statements;
}
```

19. Использование switch

```
switch (condition) {  
    case ABC :  
        statements;  
    case DEF :  
        statements;  
        break;  
    case XYZ :  
        statements;  
        break;  
    default :  
        statements;  
        break;  
}
```

20. Пробелы

```
a = (b + c) * d;           // NOT: a=(b+c)*d  
while (true)               // NOT: while(true)  
doSomething(a, b, c, d);   // NOT: doSomething(a,b,c,d);  
case 100 :                 // NOT: case 100: f  
or (i = 0; i < 10; i++)    // NOT: for(i=0;i<10;i++){
```

21. Выравнивание переменных при объявлении

```
AsciiFile*  file;  
int          nPoints;  
float        x,  y;
```