

Упражнения

1. Ниже приведены несколько фрагментов кода. Каждый из них должен быть вставлен в определенное место программы (файл `errors.cpp` на яндекс-диске). В некоторых из них есть ошибки, а в некоторых — нет. Ваша задача — найти и устранить все ошибки. Устранив эти ошибки, скомпилируйте программу, выполните её и выведите на экран слово «**Success!**». Даже если вы считаете, что нашли все ошибки, вставьте в программу исходный (неисправленный) вариант и протестируйте его; может быть, ваша догадка об ошибке была неверной или во фрагменте их несколько. Кроме того, одной из целей этого задания является анализ реакции компилятора на разные виды ошибок. Не набирайте эти фрагменты вручную — для этого существует приём «copy–paste». Не устраняйте проблемы, просто удаляя инструкции; исправляйте их, изменяя, добавляя или удаляя символы.

- (a) `Cout<<"Success!\n";`
- (b) `cout<<"Success!\n";`
- (c) `cout<<"Success"<<!\n"`
- (d) `cout<<success<<' \n';`
- (e) `string res=7; vector<int> v(10); v[5]=res; cout<<"Success!\n";`
- (f) `vector<int> v(10); v(5)=7; if (v(5)!=7) cout<<"Success!\n";`
- (g) `if (cond) cout<<"Success!\n"; else cout<<"Fail!\n";`
- (h) `bool c=false; if (c) cout<<"Success!\n"; else cout<<"Fail!\n";`
- (i) `string s="ape"; boo c="fool"<s; if (c) cout<<"Success!\n";`
- (j) `string s="ape"; if (s=="fool") cout<<"Success!\n";`

2. Откройте файл `student_errors.cpp` (размещённый на яндекс-диске). В таком стиле пишут программы многие студенты. Объясните, что делает программа. Возможно, сразу это не удастся. Скомпилируйте код. Устраните ошибки, выявленные компилятором. Для этого полезно отформатировать программу. Устраните логические ошибки и заставьте программу работать правильно. Вставьте пояснительное приглашение, чтобы было понятно, что требуется на ввод программе.
- ... Страуструп, *Программирование: принципы и практика использования C++*, глава 5 • Ошибки Выполните задания, ответьте на контрольные вопросы и постарайтесь проделать все упражнения из этой главы.