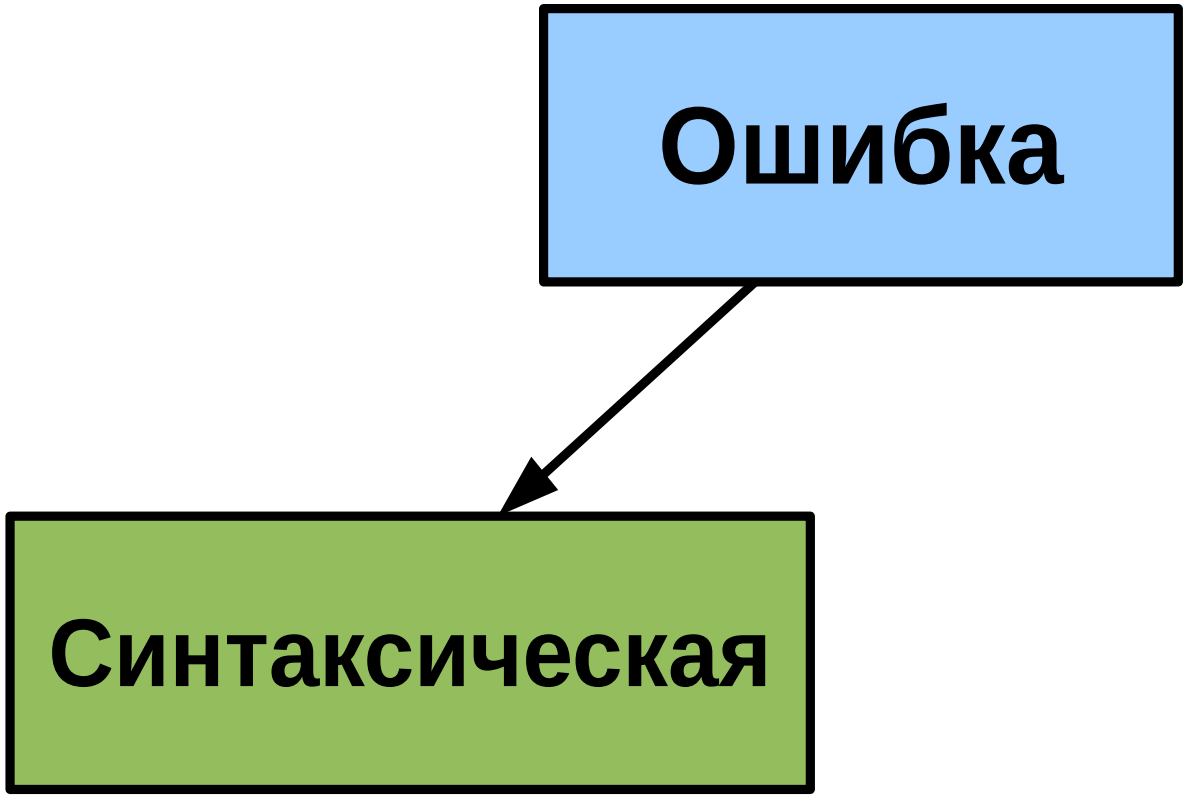
Защитное программирование

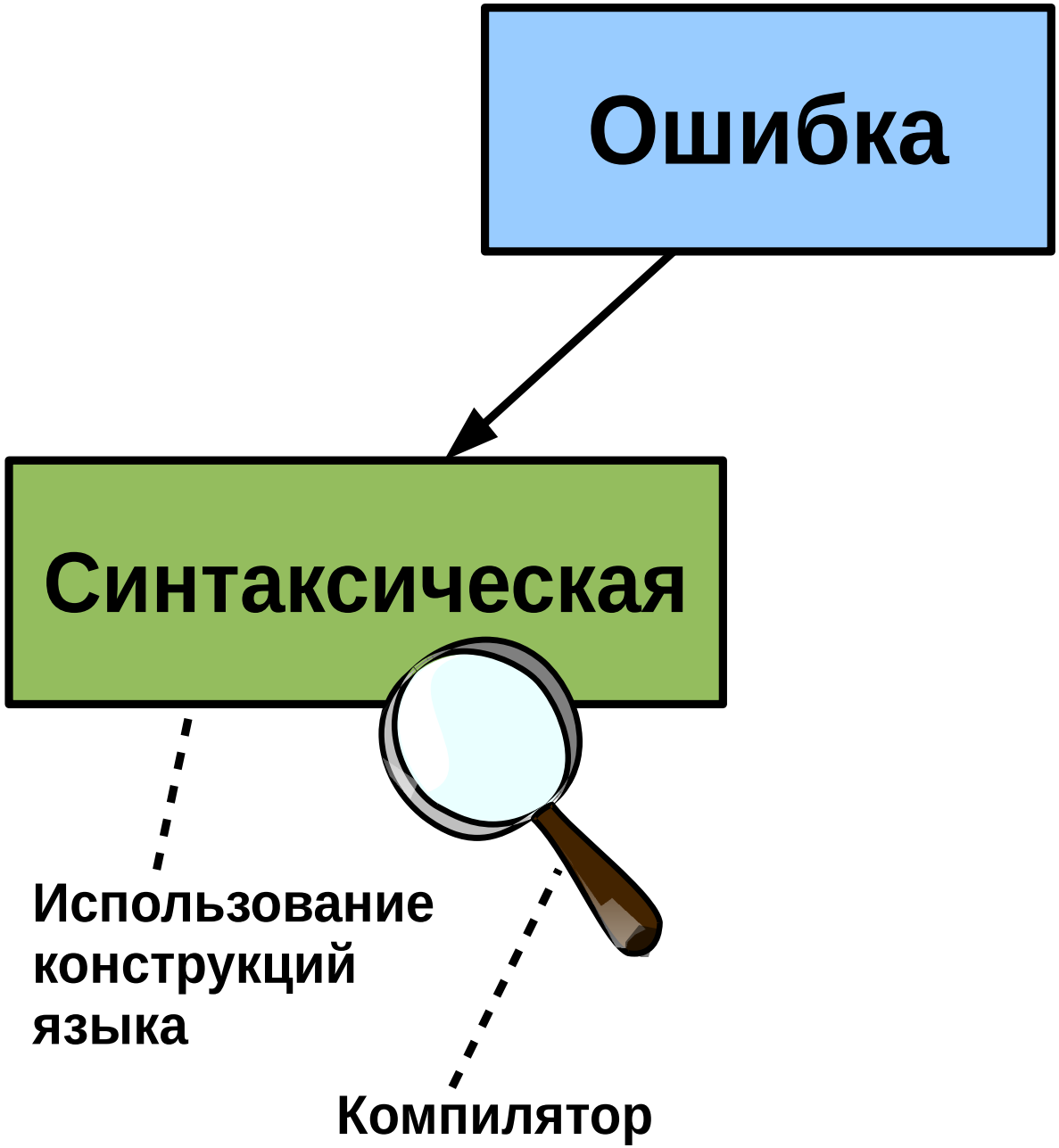
Категории ошибок

Ошибка

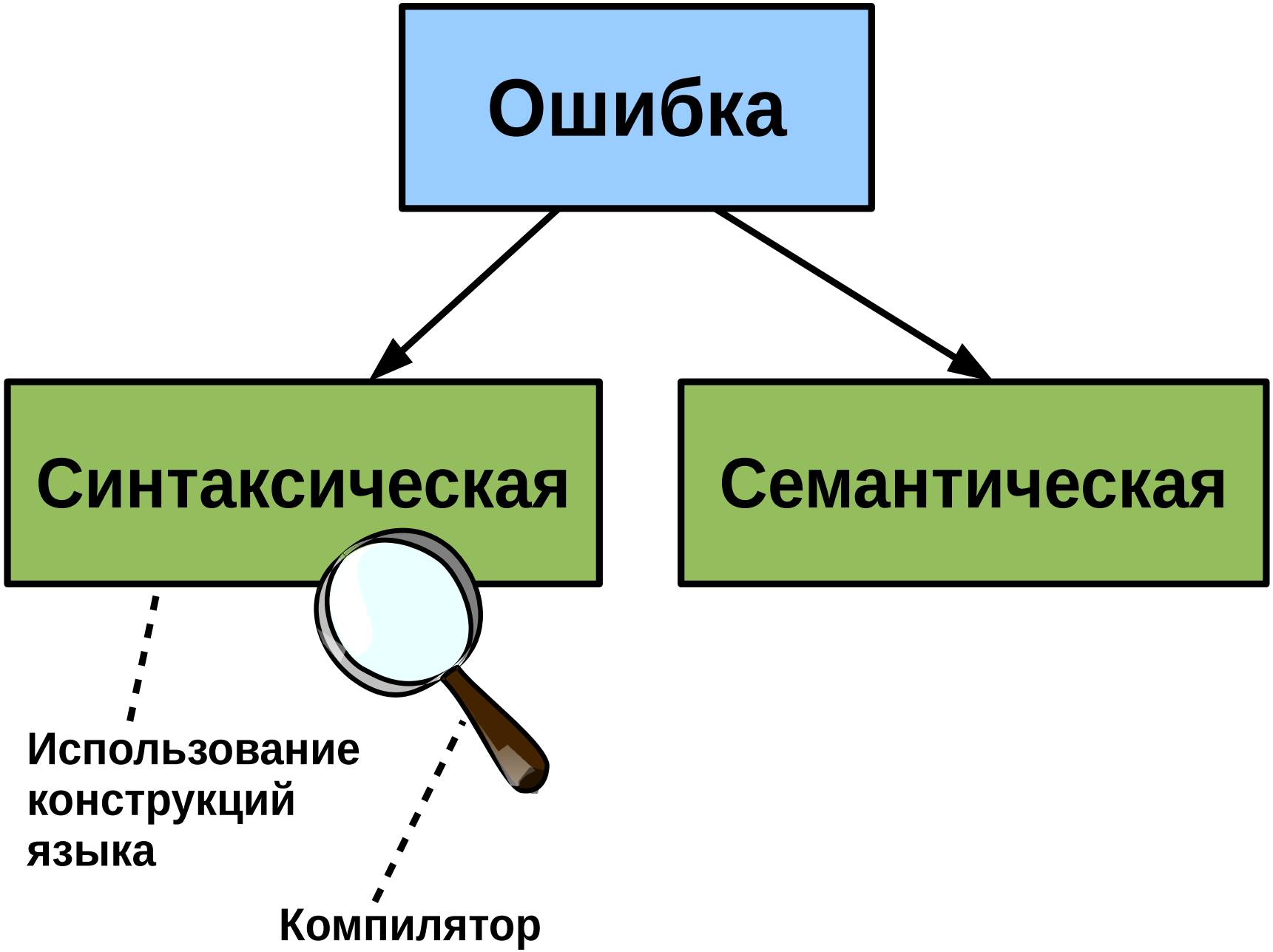
Категории ошибок



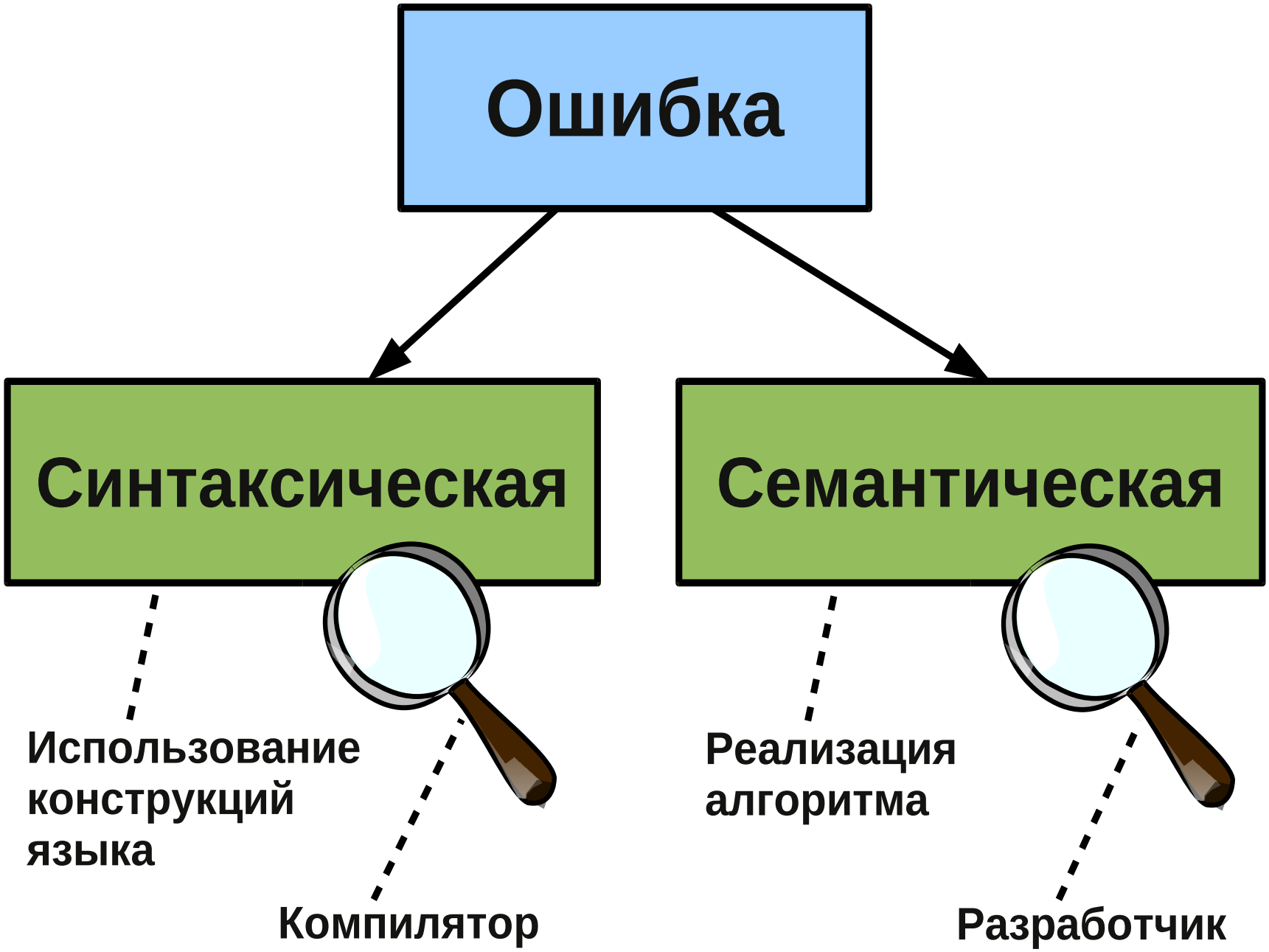
Категории ошибок



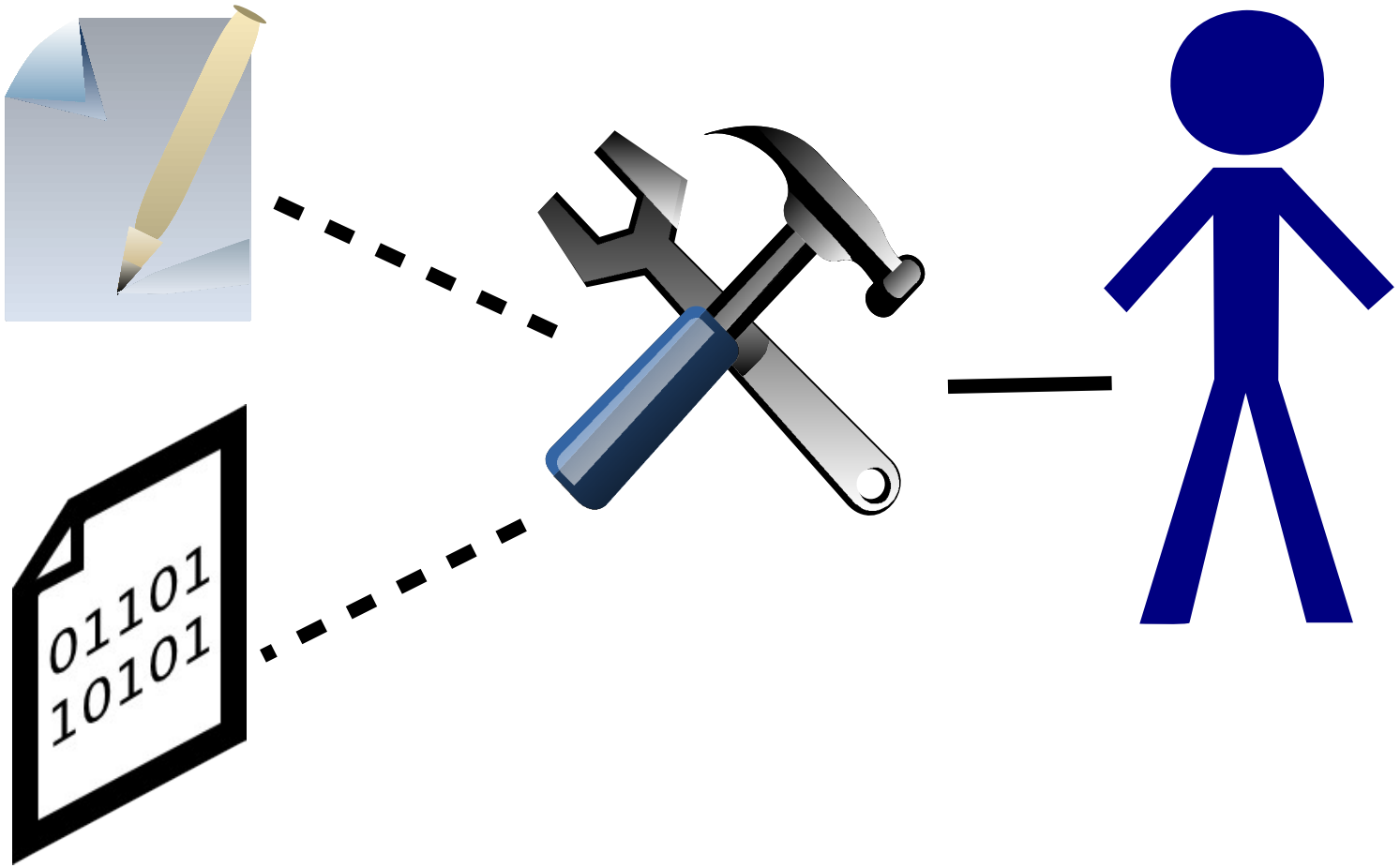
Категории ошибок



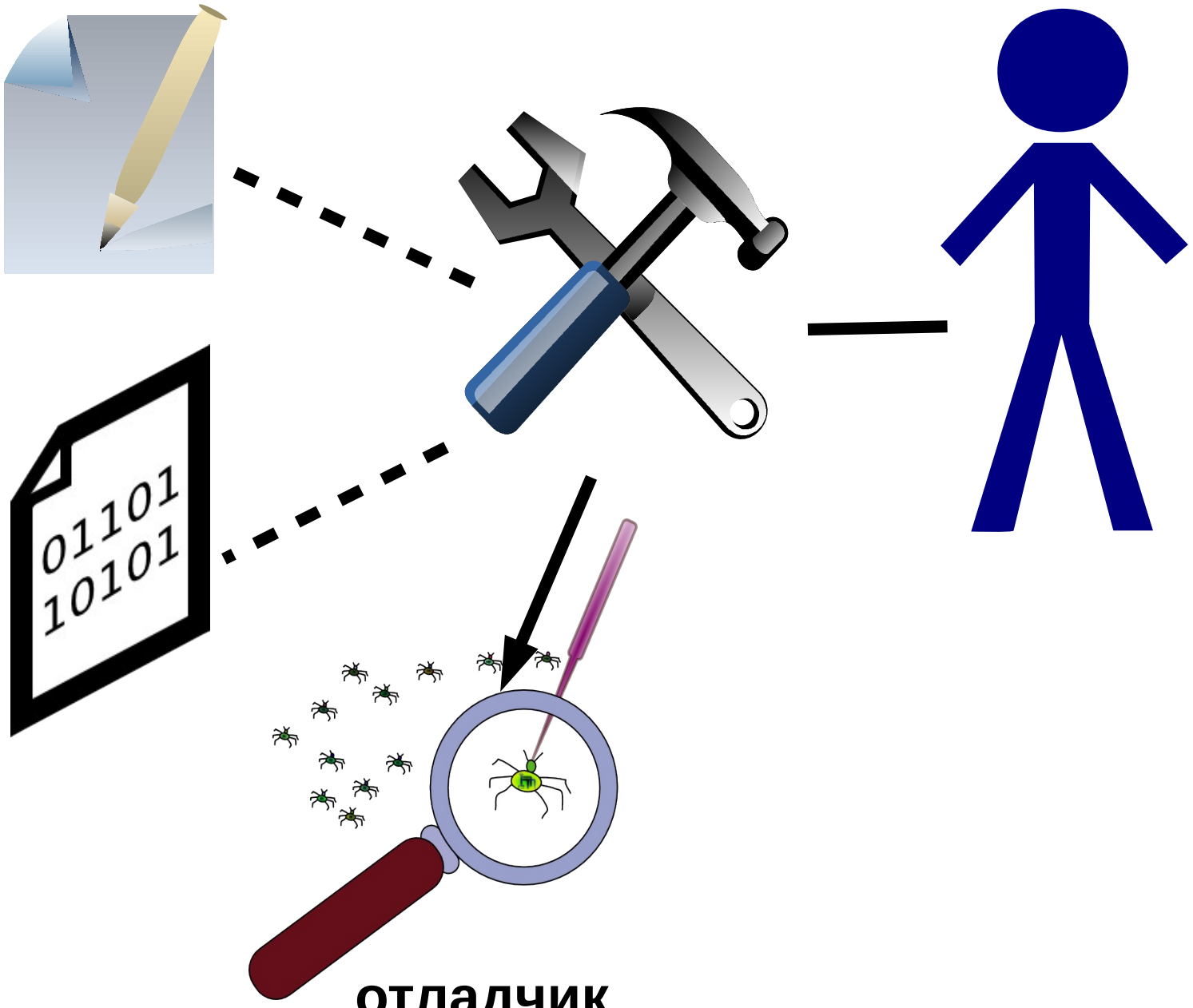
Категории ошибок



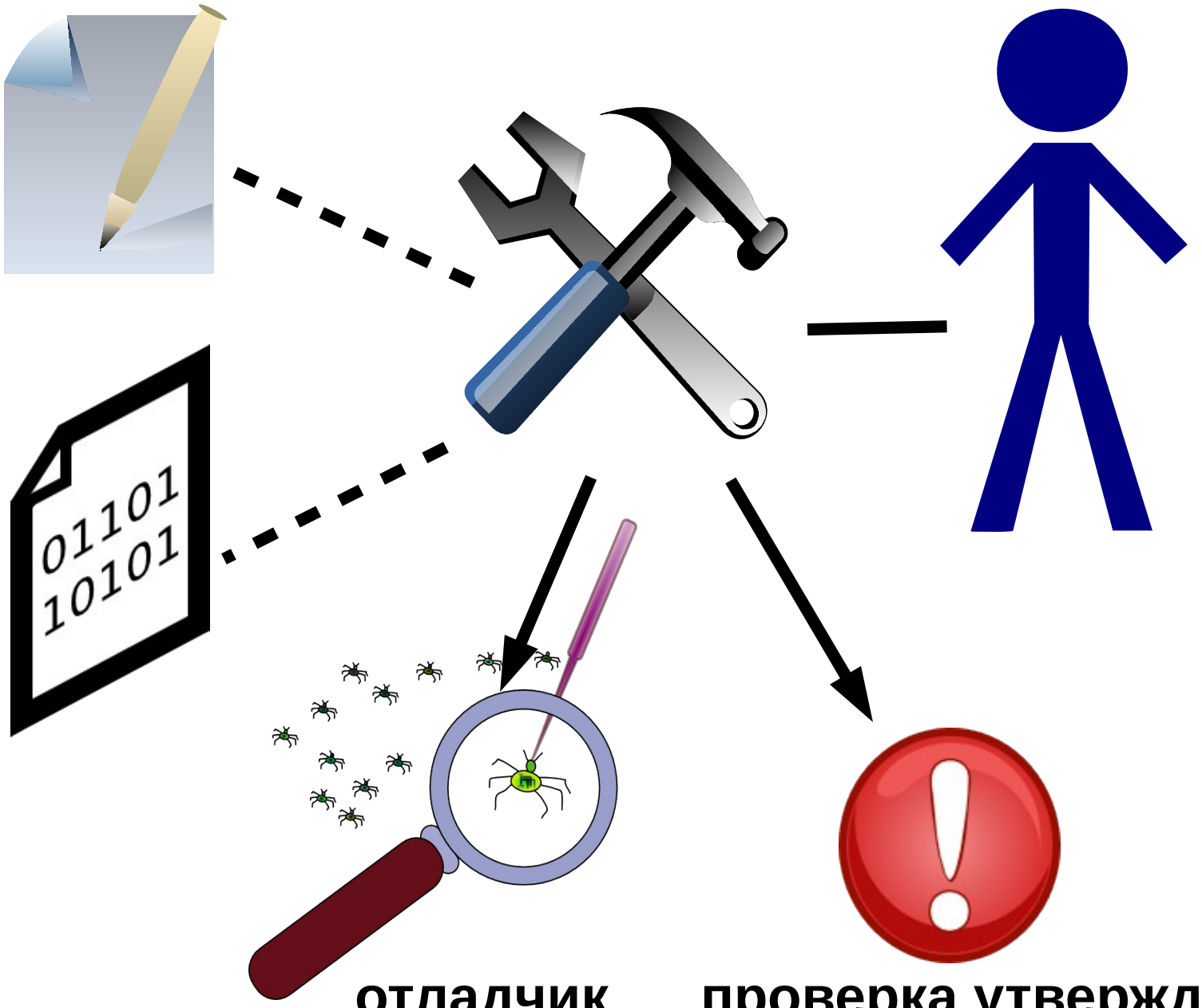
Поиск семантических ошибок



Поиск семантических ошибок

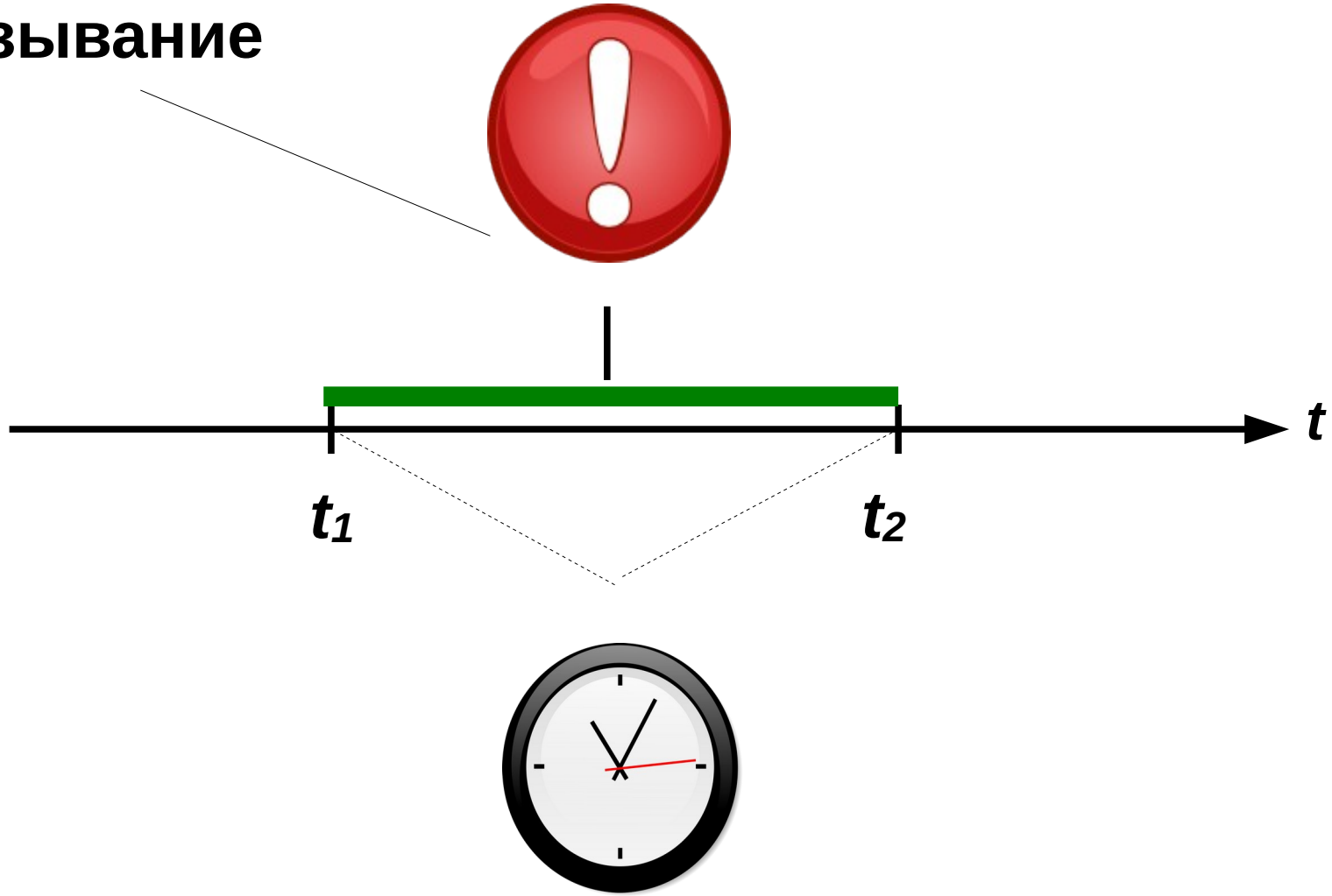


Поиск семантических ошибок



Утверждение (1)

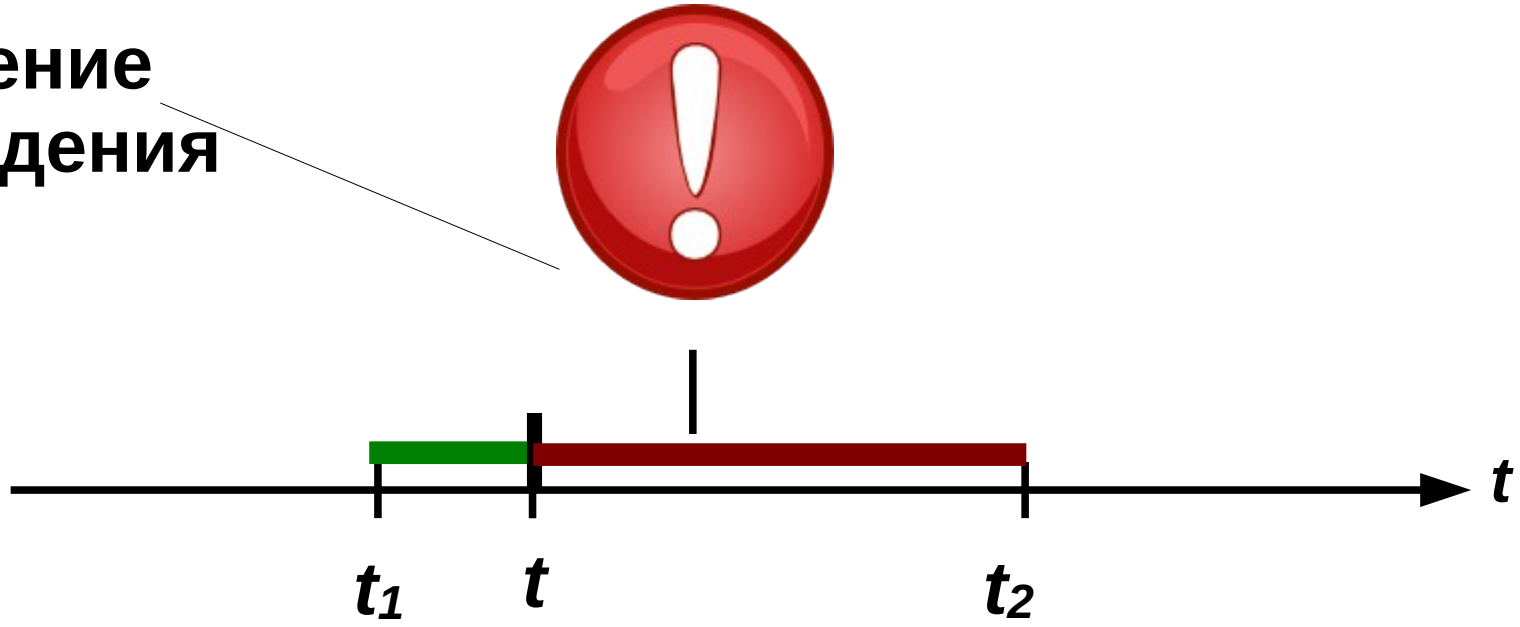
Высказывание



Утверждение (2)

ФАКТ НАЛИЧИЯ СЕМАНТИЧЕСКОЙ ОШИБКИ

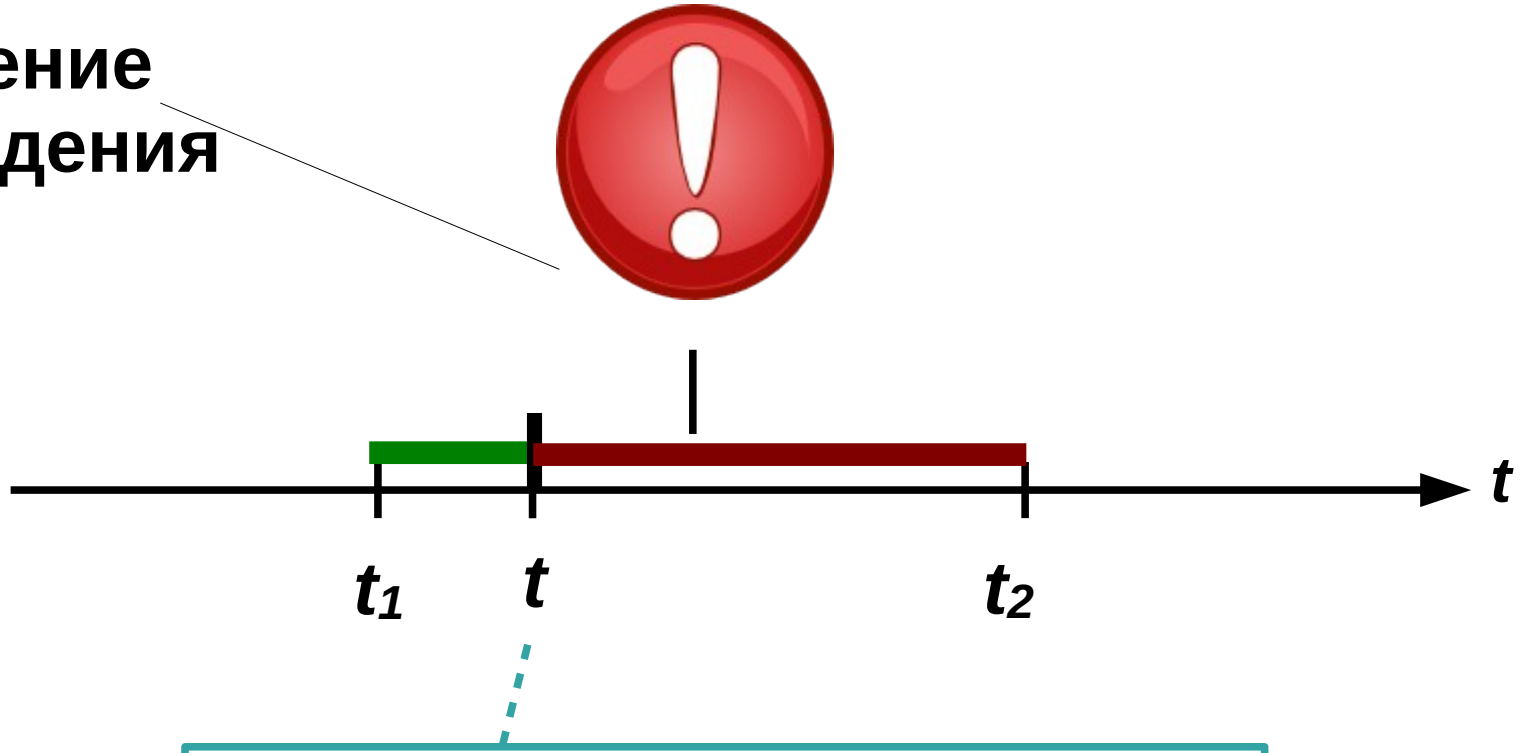
Нарушение утверждения



Утверждение (2)

ФАКТ НАЛИЧИЯ СЕМАНТИЧЕСКОЙ ОШИБКИ

Нарушение утверждения

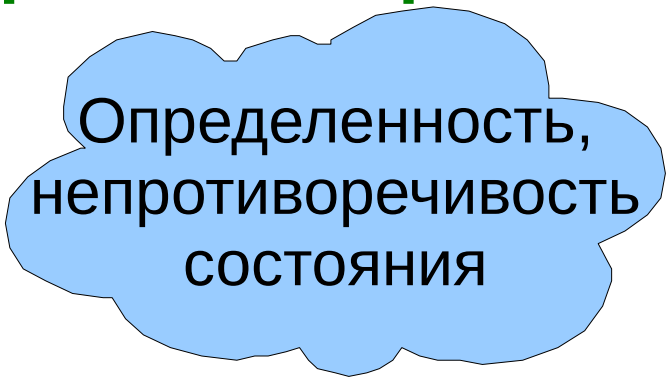


Время ОБНАРУЖЕНИЯ ошибки
Не момент совершения ошибки

Что проверять (1)

ТО, ЧТО «НУ НИКАК НЕ МОЖЕТ ПРОИЗОЙТИ»

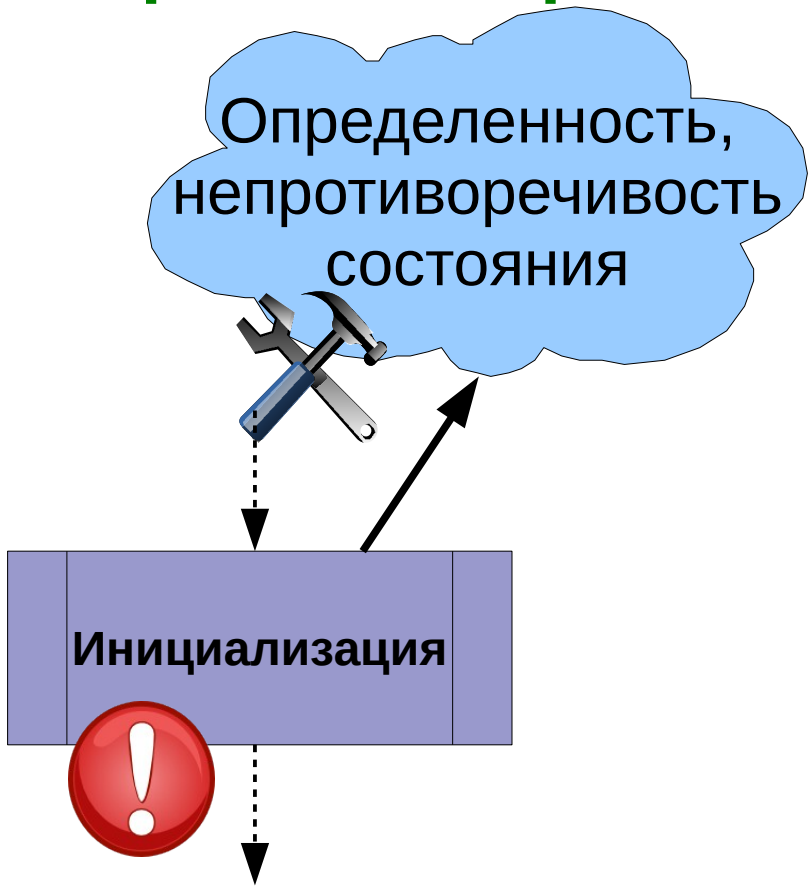
Инвариант абстрактного типа данных



Что проверять (1)

ТО, ЧТО «НУ НИКАК НЕ МОЖЕТ ПРОИЗОЙТИ»

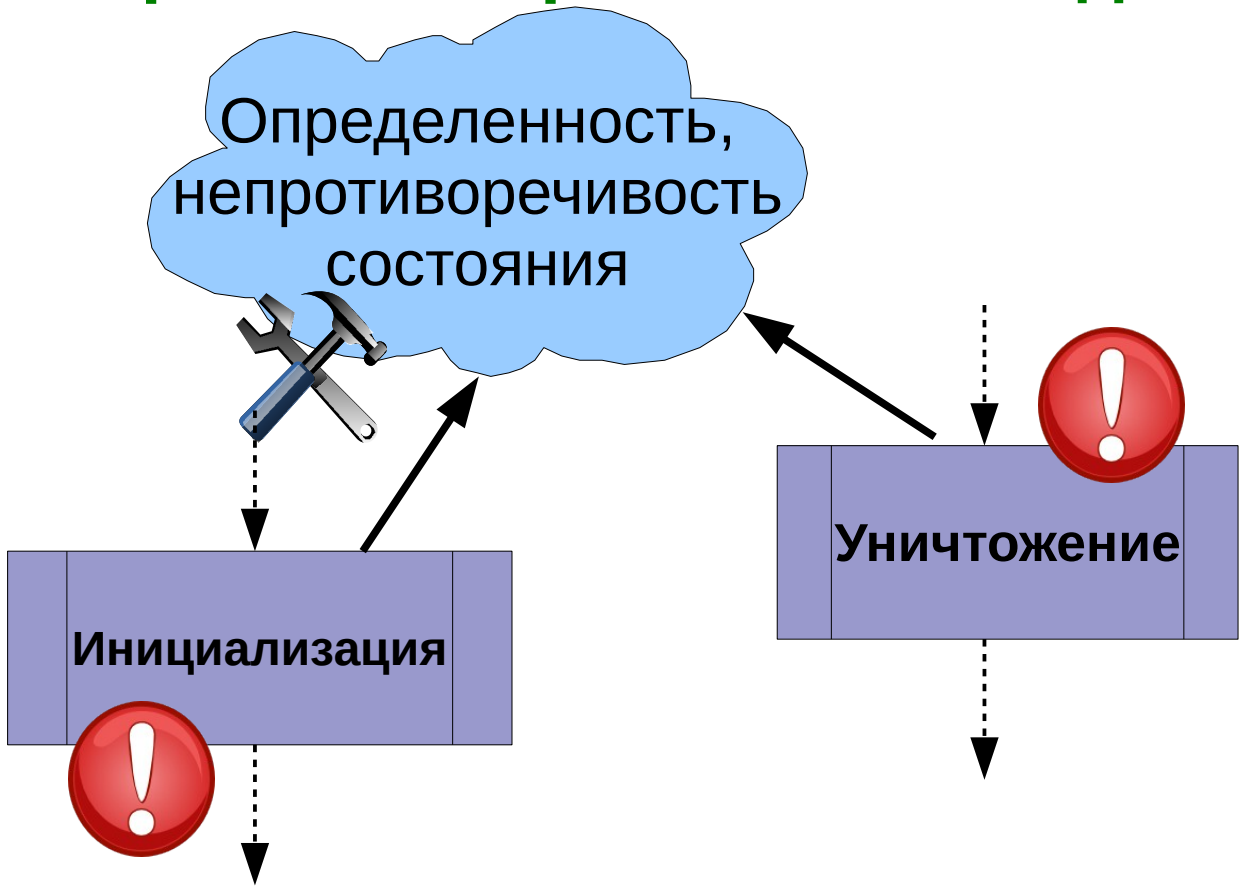
Инвариант абстрактного типа данных



Что проверять (1)

ТО, ЧТО «НУ НИКАК НЕ МОЖЕТ ПРОИЗОЙТИ»

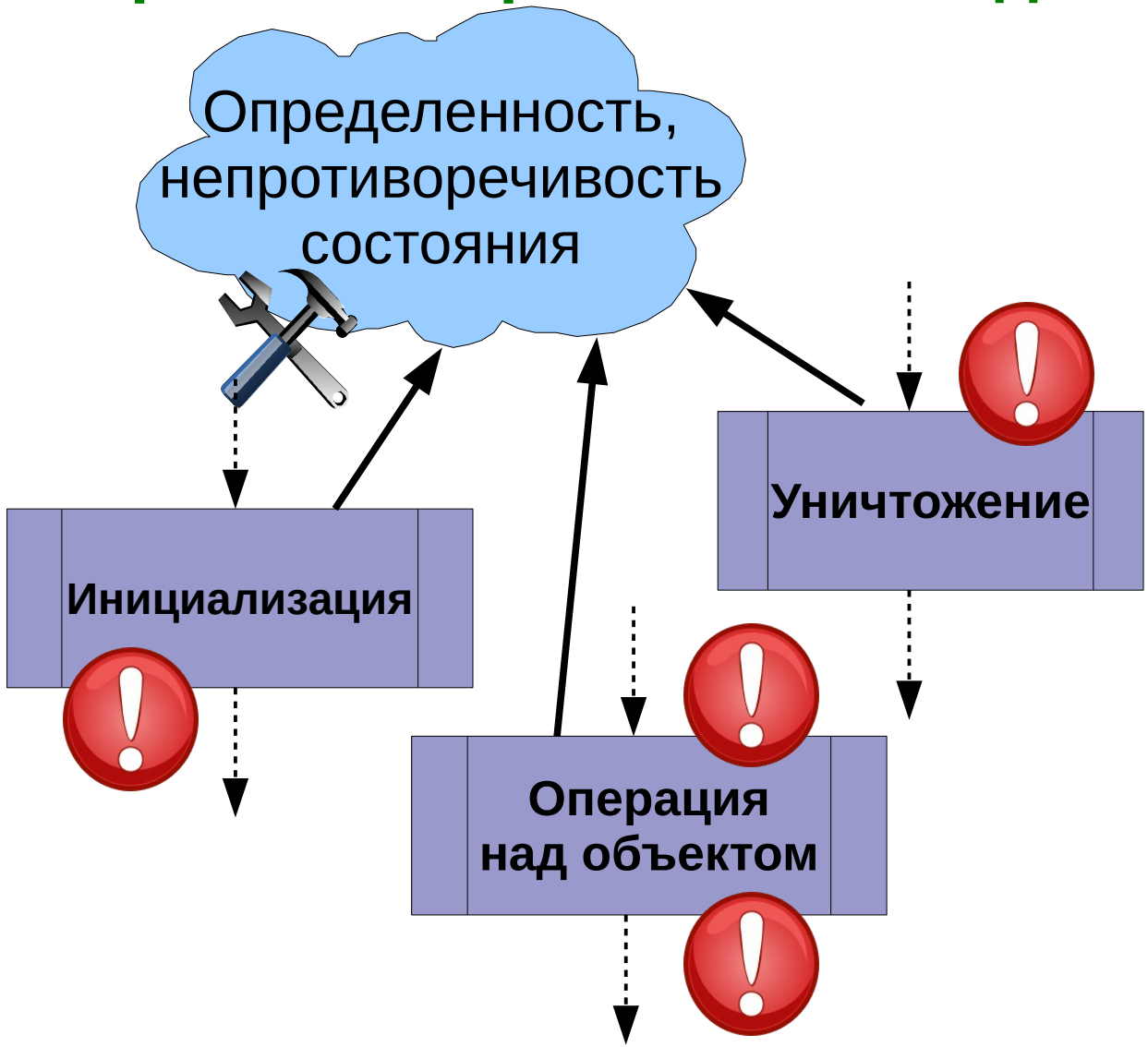
Инвариант абстрактного типа данных



Что проверять (1)

ТО, ЧТО «НУ НИКАК НЕ МОЖЕТ ПРОИЗОЙТИ»

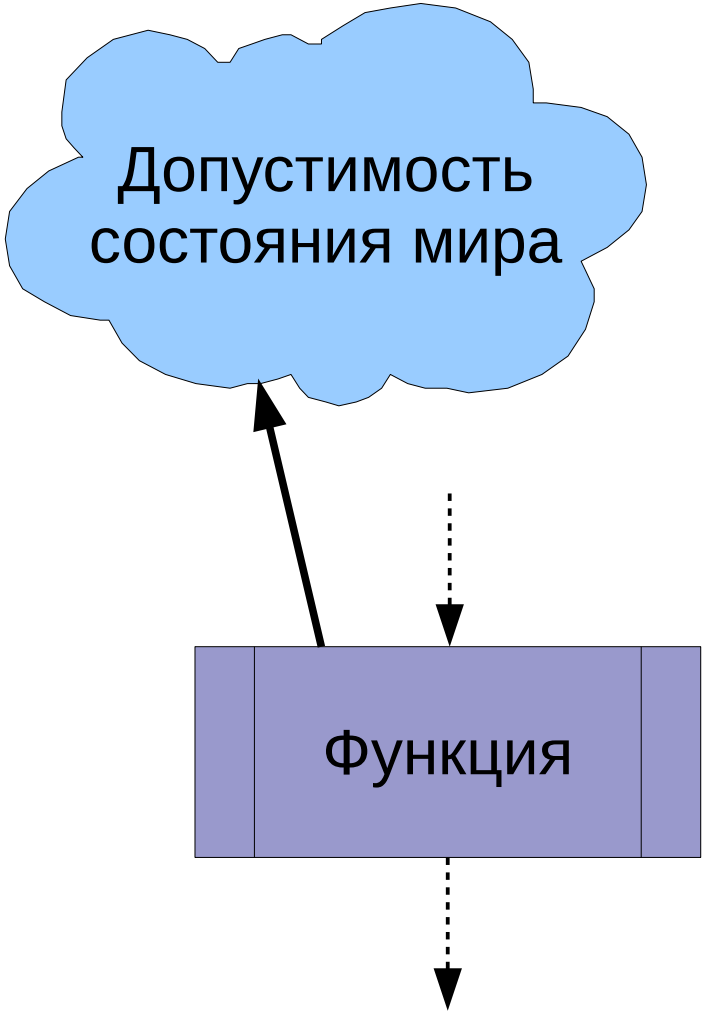
Инвариант абстрактного типа данных



Что проверять (2)

ТО, ЧТО «НУ НИКАК НЕ МОЖЕТ ПРОИЗОЙТИ»

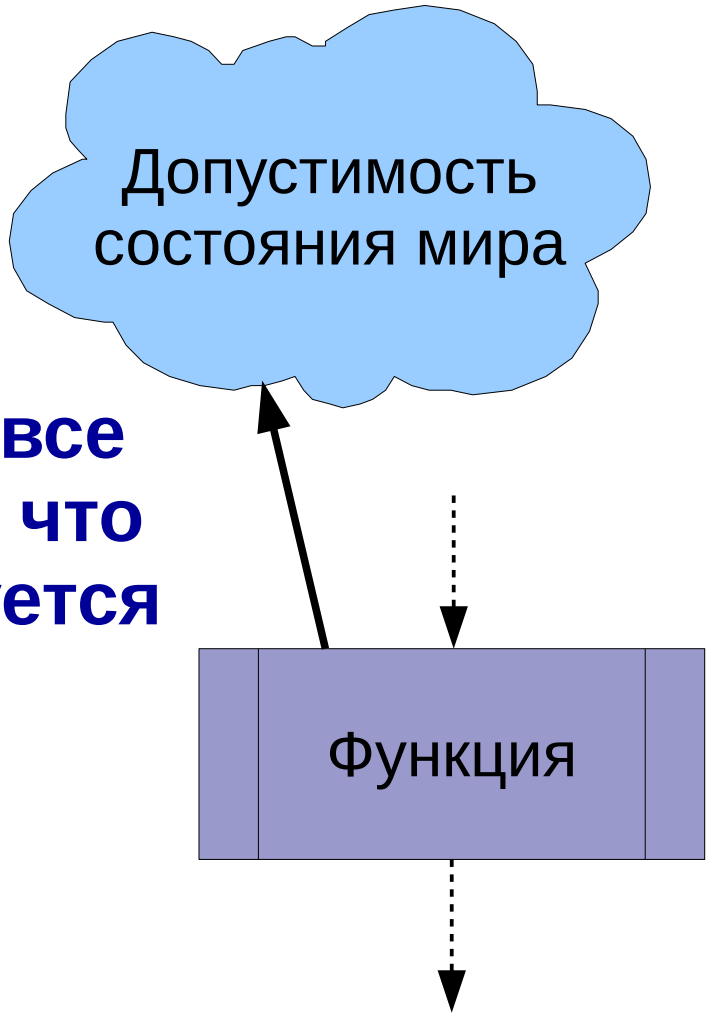
Предусловия и постусловия функций



Что проверять (2)

ТО, ЧТО «НУ НИКАК НЕ МОЖЕТ ПРОИЗОЙТИ»

Предусловия и постусловия функций



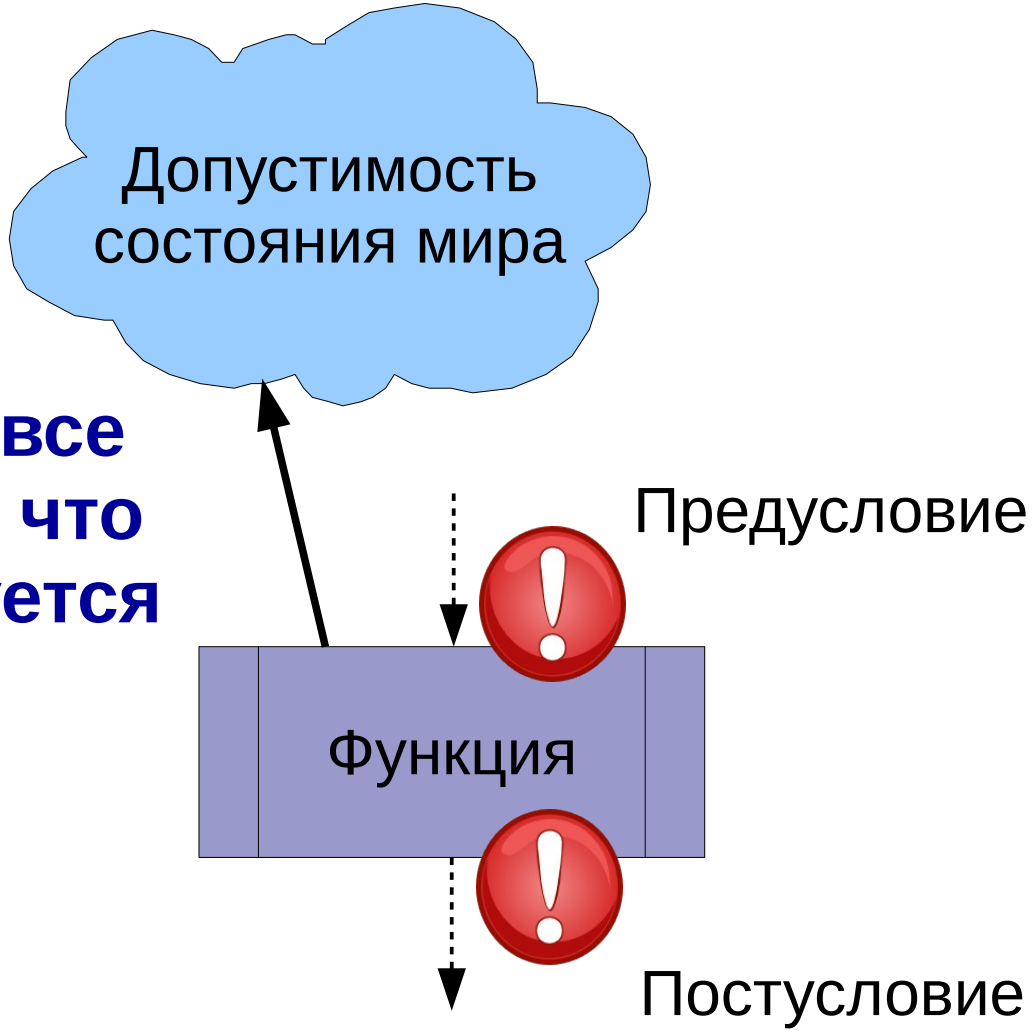
Проверь все внешнее, что используется внутри

Что проверять (2)

ТО, ЧТО «НУ НИКАК НЕ МОЖЕТ ПРОИЗОЙТИ»

Предусловия и постусловия функций

Проверь все внешнее, что используется внутри



Как проверять

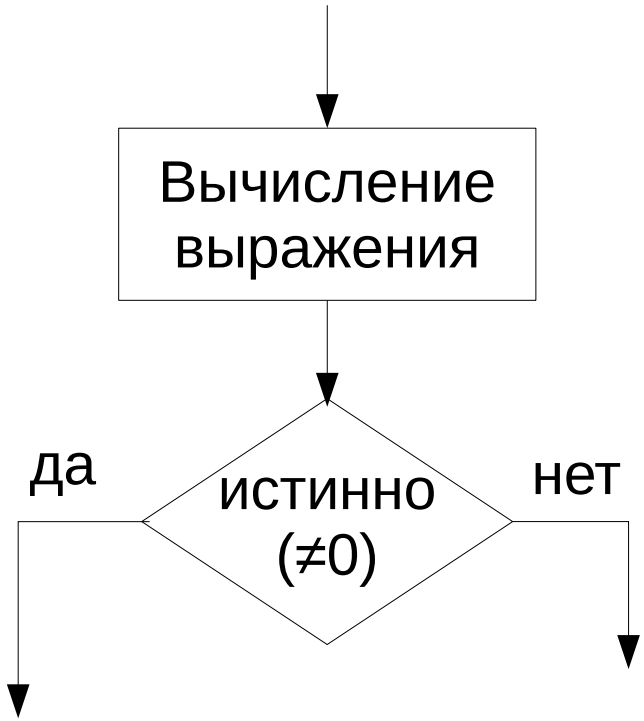
```
#include <cassert>

void assert(выражение);
```

Как проверять

```
#include <cassert>

void assert(выражение);
```

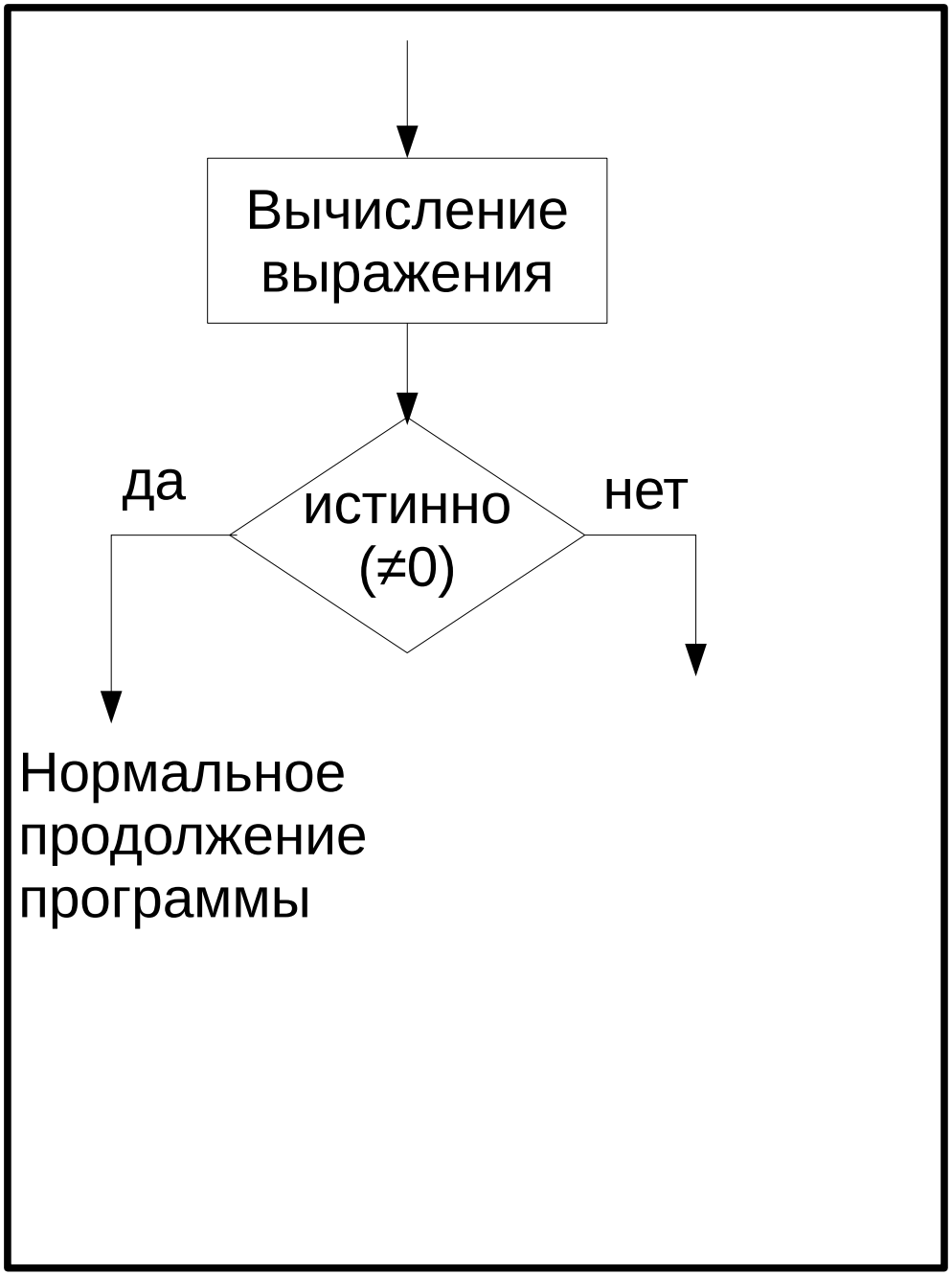
Как проверять

```
#include <cassert>

void assert(выражение);
```



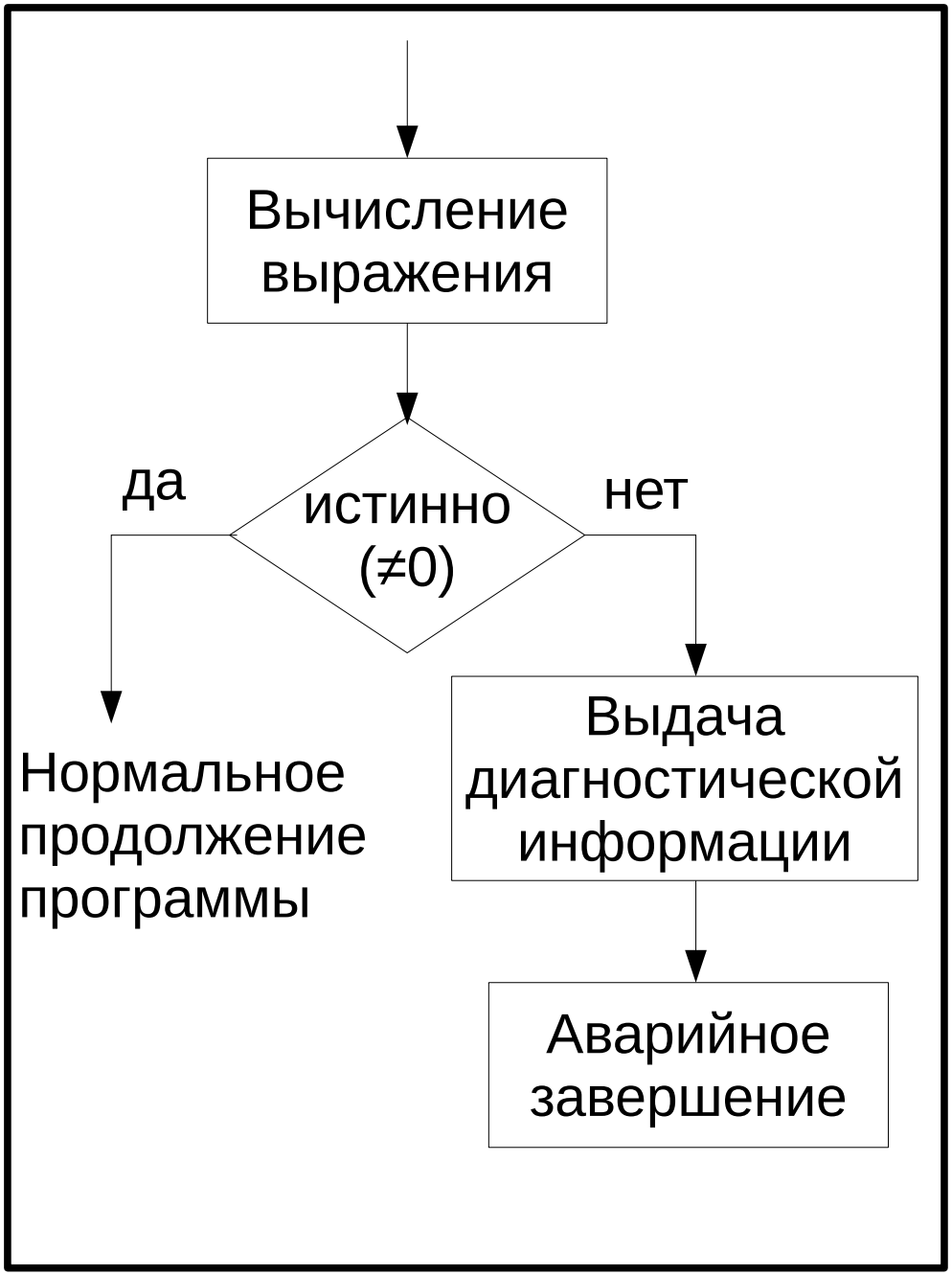
Как проверять

```
#include <cassert>

void assert(выражение);
```



Как проверять

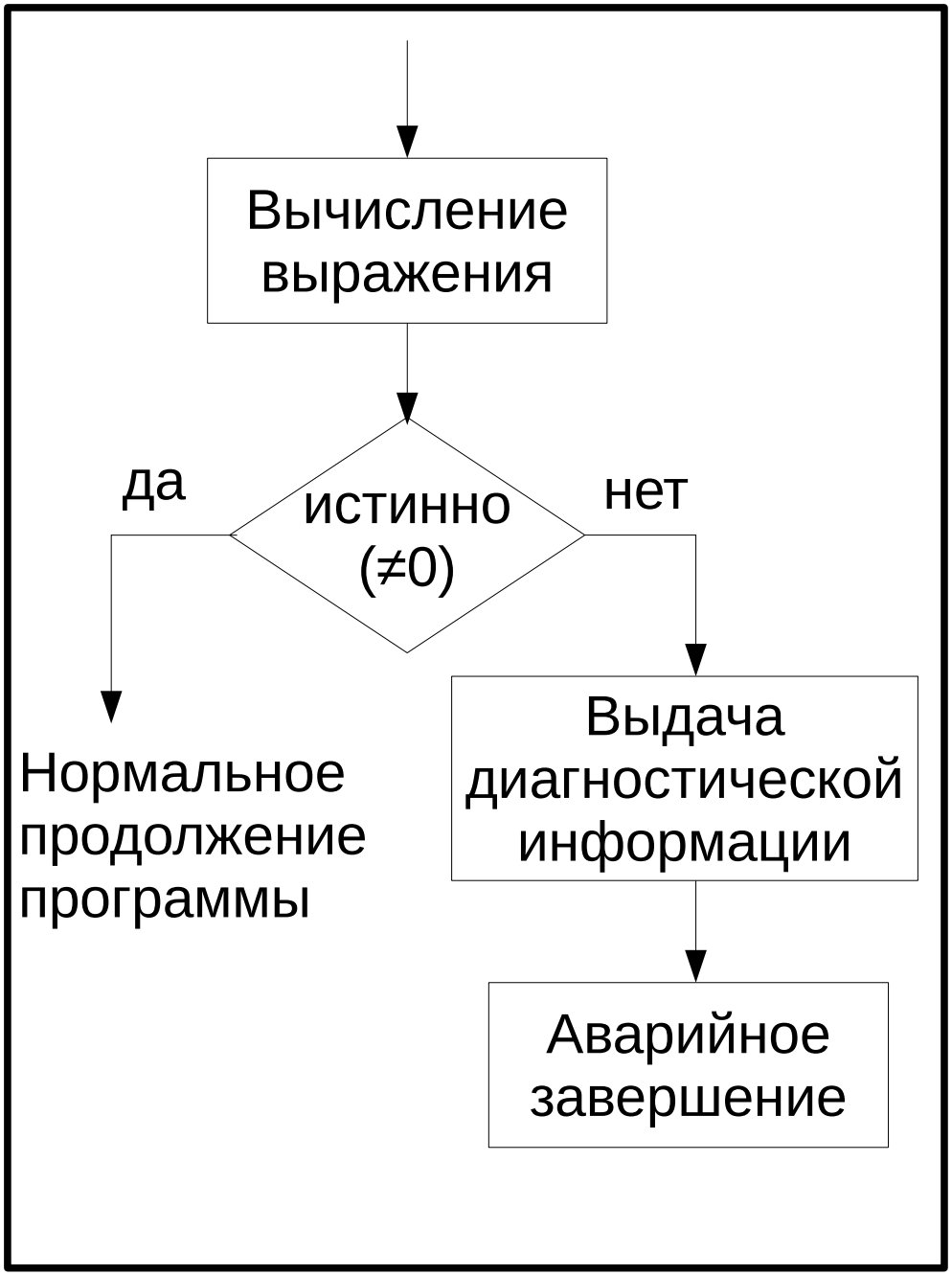
```
#include <cassert>

void assert(выражение);
```



Как проверять

```
#include <cassert>

void assert(выражение);
```

- ✔**Макроопределение (через #define)**
- ✔**Выражение — проверяемое утверждение**
- ✔**Значение выражения — логическое или целочисленное**
- ✔**Управление макросом - NDEBUG**



Пример проверки утверждения

ВЫЧИСЛЕНИЕ СКАЛЯРНОГО ПРОИЗВЕДЕНИЯ ВЕКТОРОВ

$$(X, Y) = \sum_{i=1}^{n} X_i \cdot Y_i,$$

Предусловие: размерности векторов совпадают

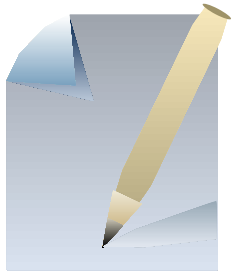
Пример проверки утверждения

ВЫЧИСЛЕНИЕ СКАЛЯРНОГО ПРОИЗВЕДЕНИЯ ВЕКТОРОВ

$$(X, Y) = \sum_{i=1}^{n} X_i \cdot Y_i,$$

Предусловие: размерности векторов совпадают

```
double vectorScalar(const Vector& first,
                    const  Vector& second) {
```



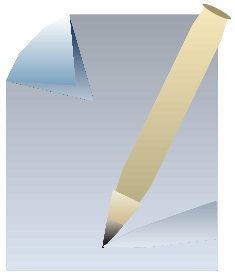
Пример проверки утверждения

ВЫЧИСЛЕНИЕ СКАЛЯРНОГО ПРОИЗВЕДЕНИЯ ВЕКТОРОВ

$$(X, Y) = \sum_{i=1}^{n} X_i \cdot Y_i,$$

Предусловие: размерности векторов совпадают

```
double vectorScalar(const Vector& first,
                    const Vector& second) {
    assert( first.coordinates.size() ==
            second.coordinates.size() );
```



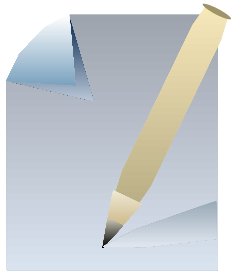
Пример проверки утверждения

ВЫЧИСЛЕНИЕ СКАЛЯРНОГО ПРОИЗВЕДЕНИЯ ВЕКТОРОВ

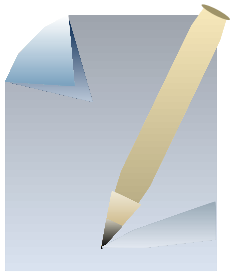
$$(X, Y) = \sum_{i=1}^{n} X_i \cdot Y_i,$$

Предусловие: размерности векторов совпадают

```
double vectorScalar(const Vector& first,
                    const  Vector& second) {
    assert( first.coordinates.size() ==
            second.coordinates.size() );
    double result = 0;
    unsigned size = first.coordinates.size();
    for(unsigned i = 0; i < size; ++i)
        result += first.coordinates.[i] *
                  second.coordinates.[i];
    return result;
}
```

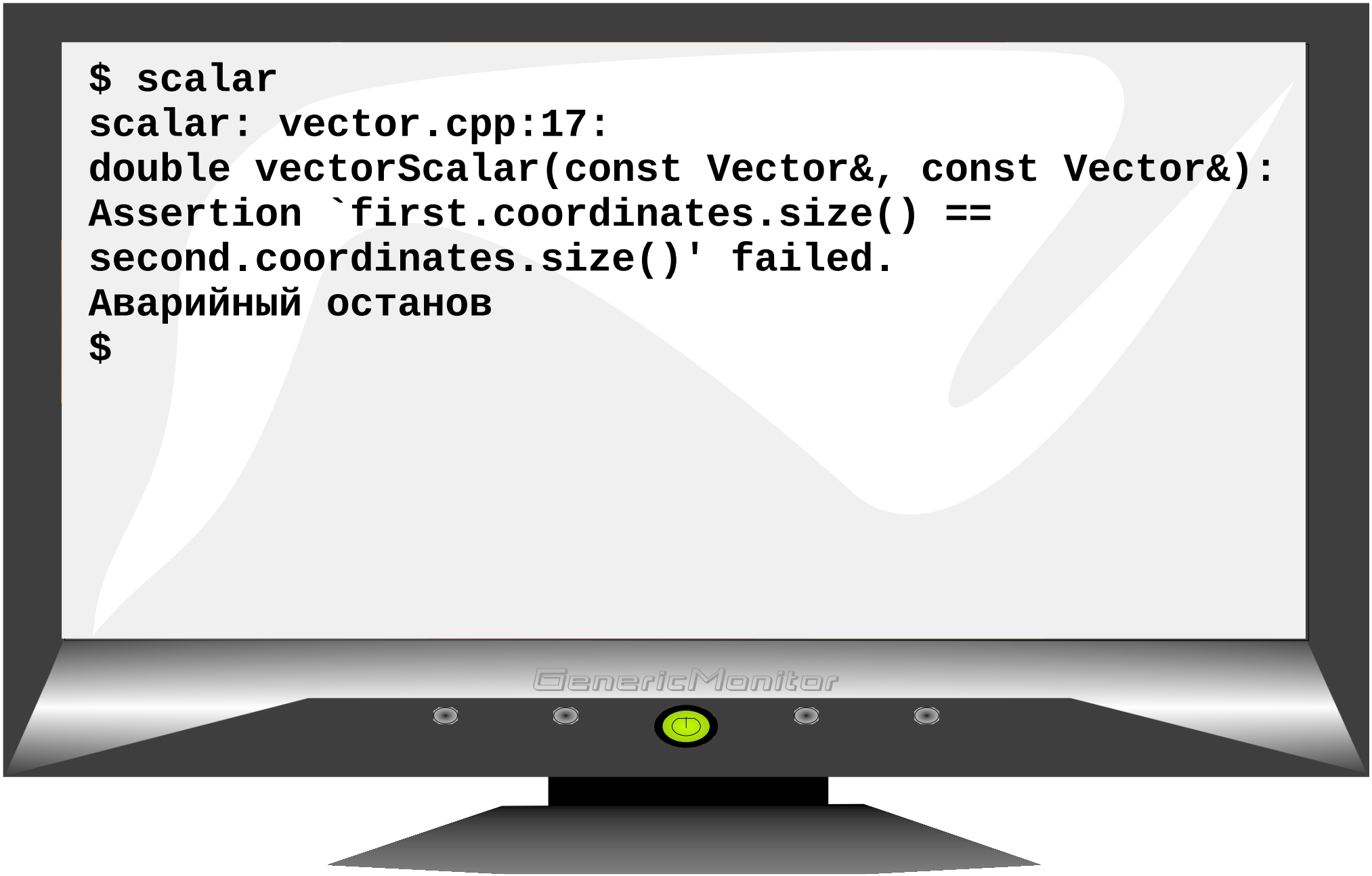


Пример нарушения утверждения



```
int main()
{
    Vector x;
    Vector y;
    vectorResize(x,5);
    vectorResize(y,6);
    double answer = vectorScalar(x,y);
    std::cout << "answer: " << answer << std::endl;
    return 0;
}
```

Пример выполнения



Особые ситуации

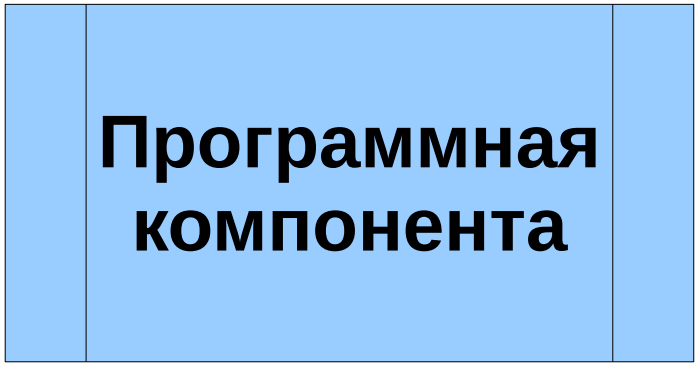
ОНИ МОГУТ НАСТУПИТЬ В ЛЮБОМ СЛУЧАЕ

ОНИ МОГУТ НАСТУПИТЬ В ЛЮБОМ СЛУЧАЕ

- ✔Деление на 0
- ✔Переполнение разрядной сетки
- ✔Отсутствие свободной памяти
- ✔Чтение недопустимого значения переменной из потока
- ✔Отсутствие файла при попытке его открытия
- ✔. . .

ОНИ МОГУТ НАСТУПИТЬ В ЛЮБОМ СЛУЧАЕ

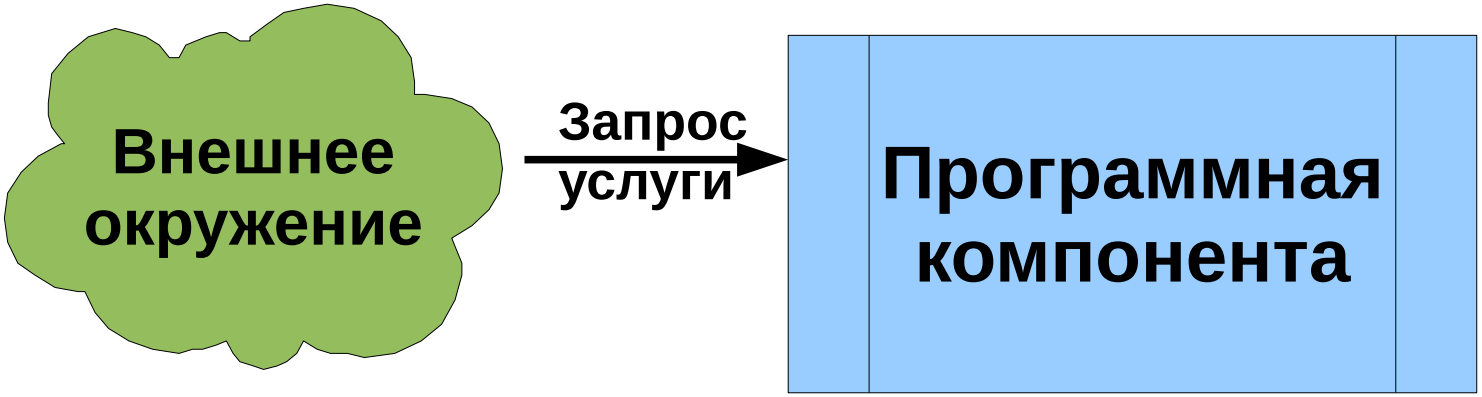
- ✔Деление на 0
- ✔Переполнение разрядной сетки
- ✔Отсутствие свободной памяти
- ✔Чтение недопустимого значения переменной из потока
- ✔Отсутствие файла при попытке его открытия
- ✔. . .



Особые ситуации

ОНИ МОГУТ НАСТУПИТЬ В ЛЮБОМ СЛУЧАЕ

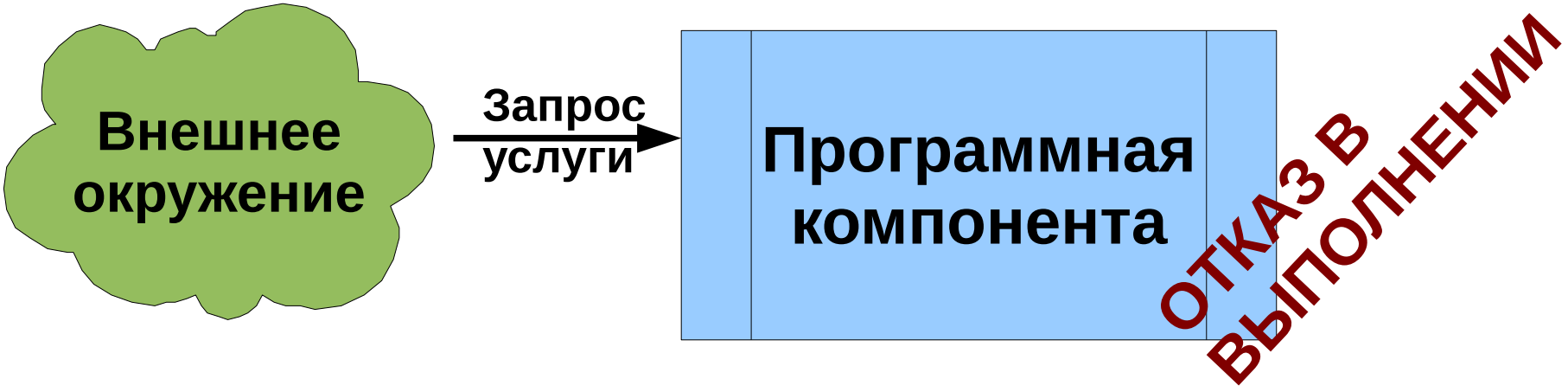
- ✔Деление на 0
- ✔Переполнение разрядной сетки
- ✔Отсутствие свободной памяти
- ✔Чтение недопустимого значения переменной из потока
- ✔Отсутствие файла при попытке его открытия
- ✔. . .



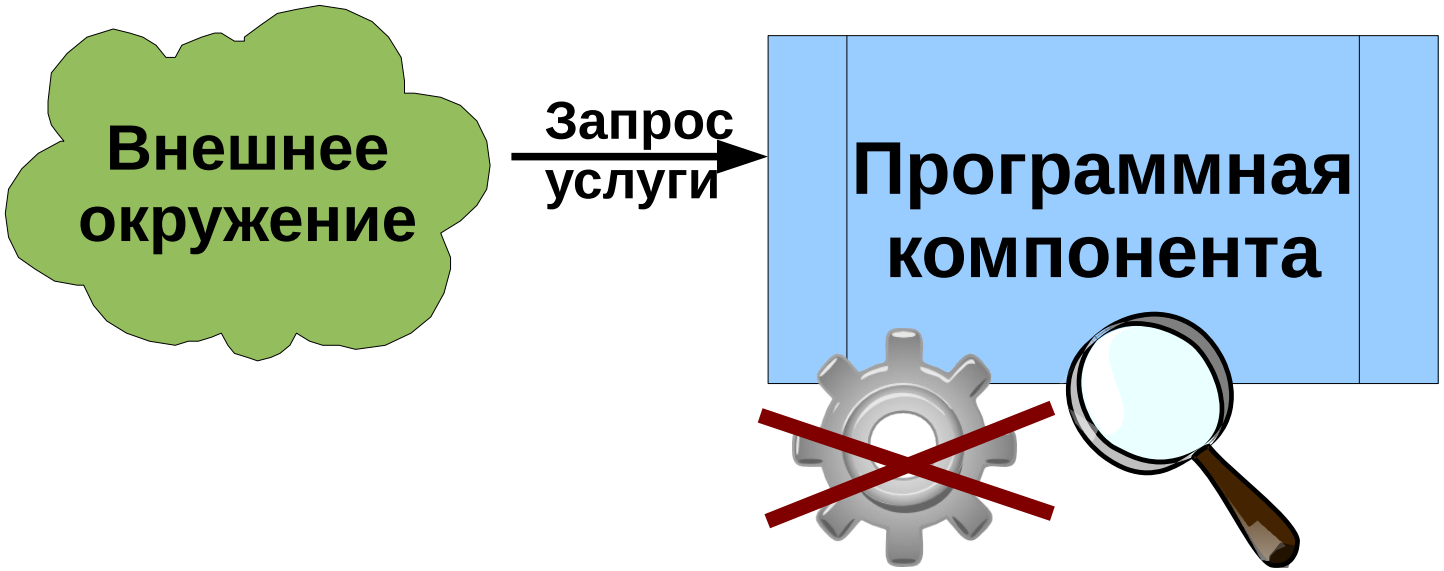
Особые ситуации

ОНИ МОГУТ НАСТУПИТЬ В ЛЮБОМ СЛУЧАЕ

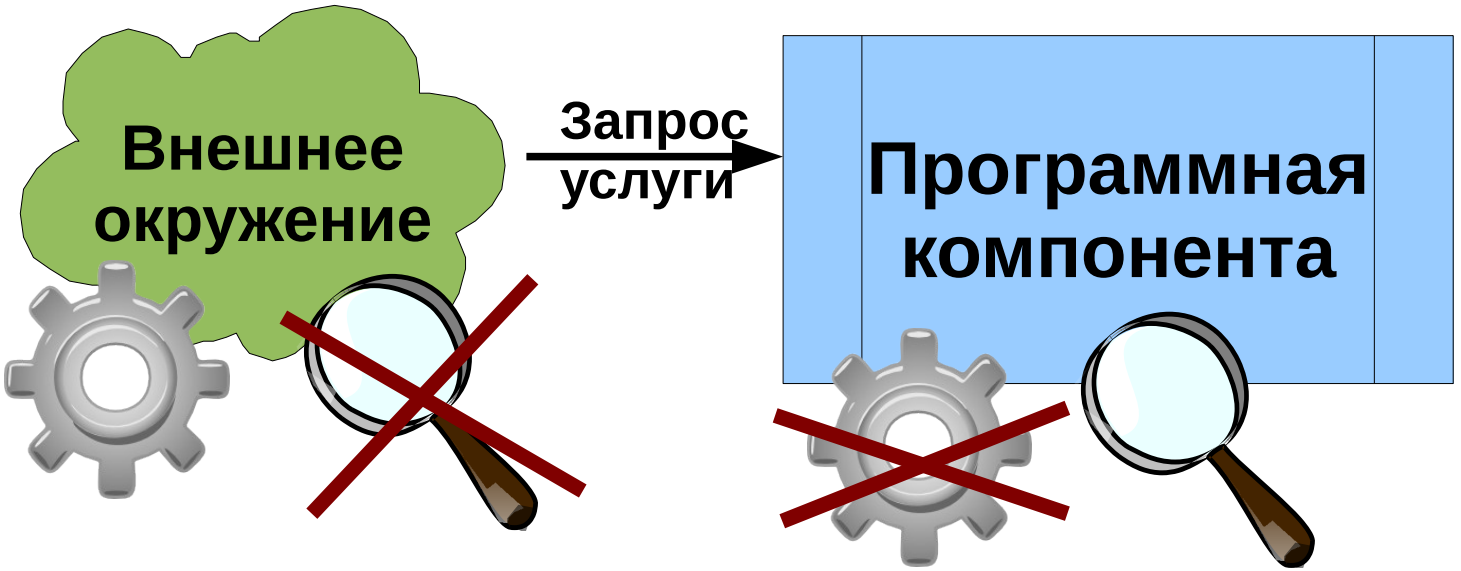
- ✔Деление на 0
- ✔Переполнение разрядной сетки
- ✔Отсутствие свободной памяти
- ✔Чтение недопустимого значения переменной из потока
- ✔Отсутствие файла при попытке его открытия
- ✔. . .



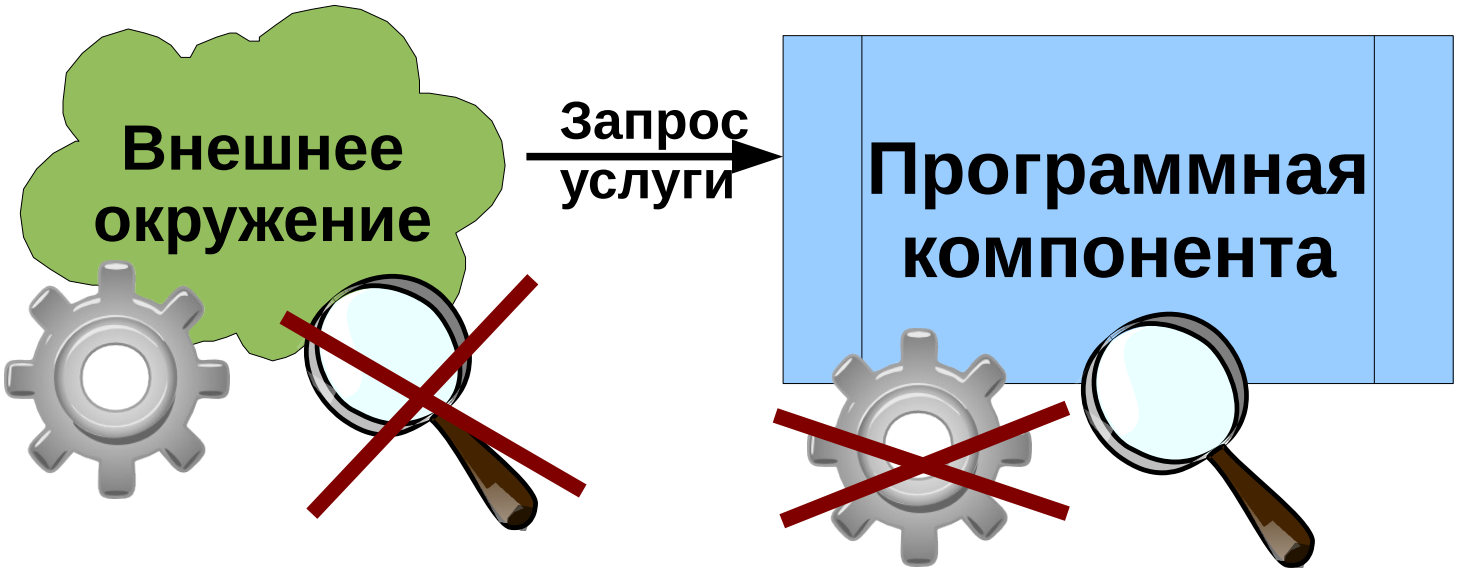
Обработка особых ситуаций



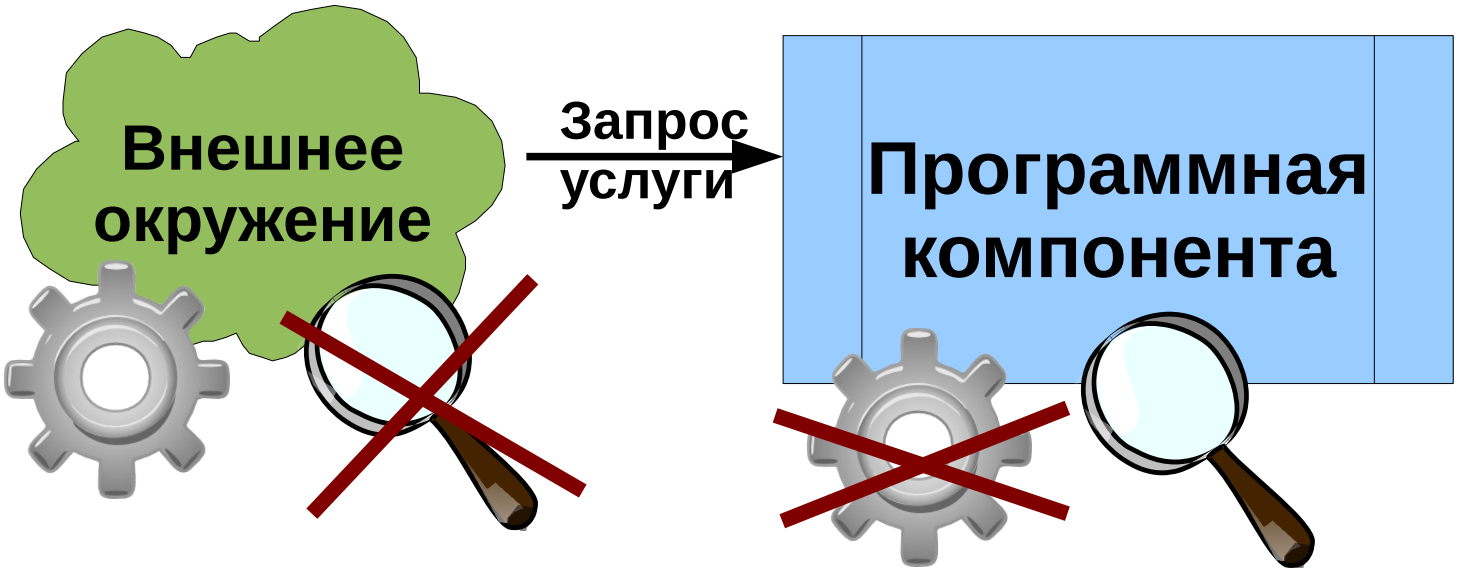
Обработка особых ситуаций



Обработка особых ситуаций

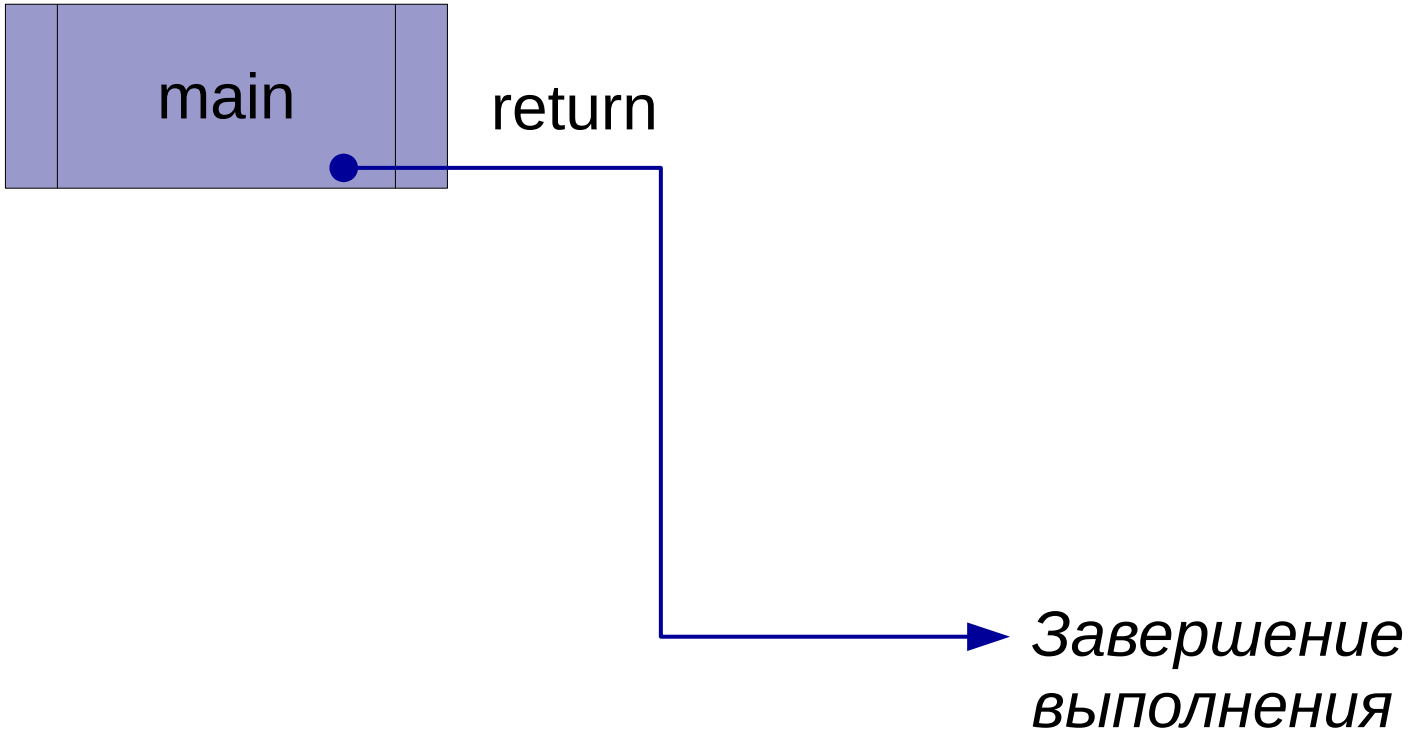


Обработка особых ситуаций

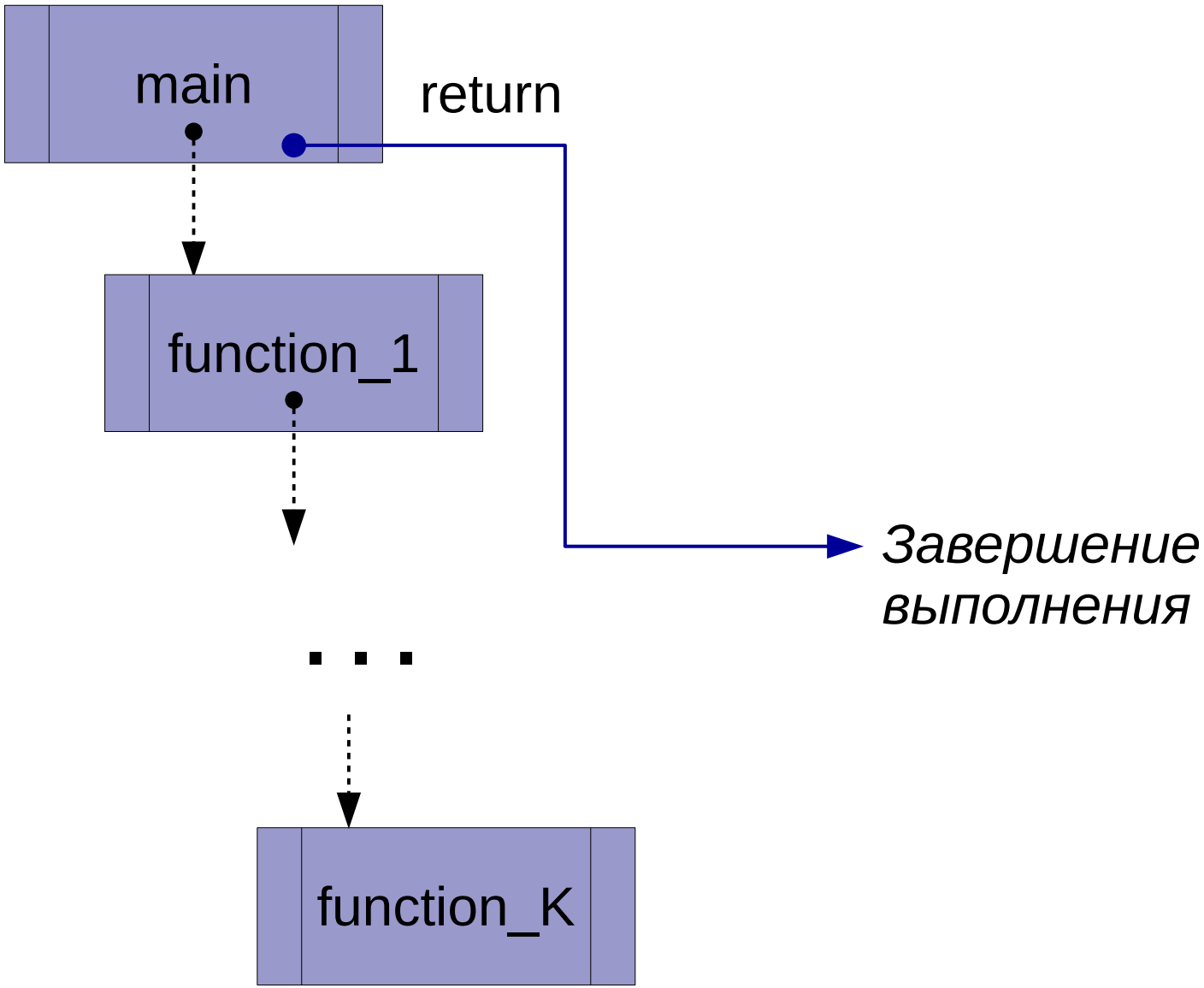


Прекратить выполнение программы

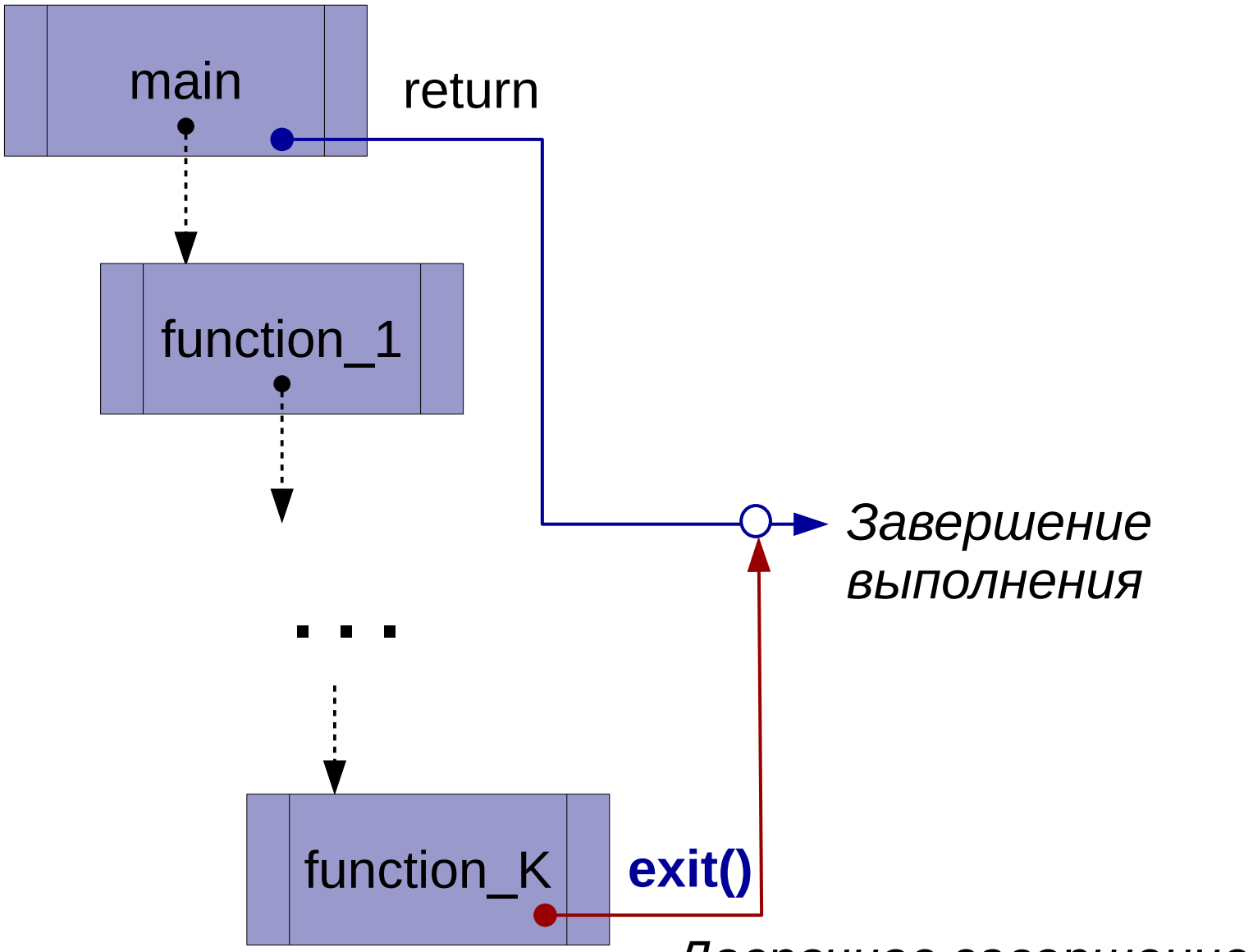
Прекратить выполнение программы



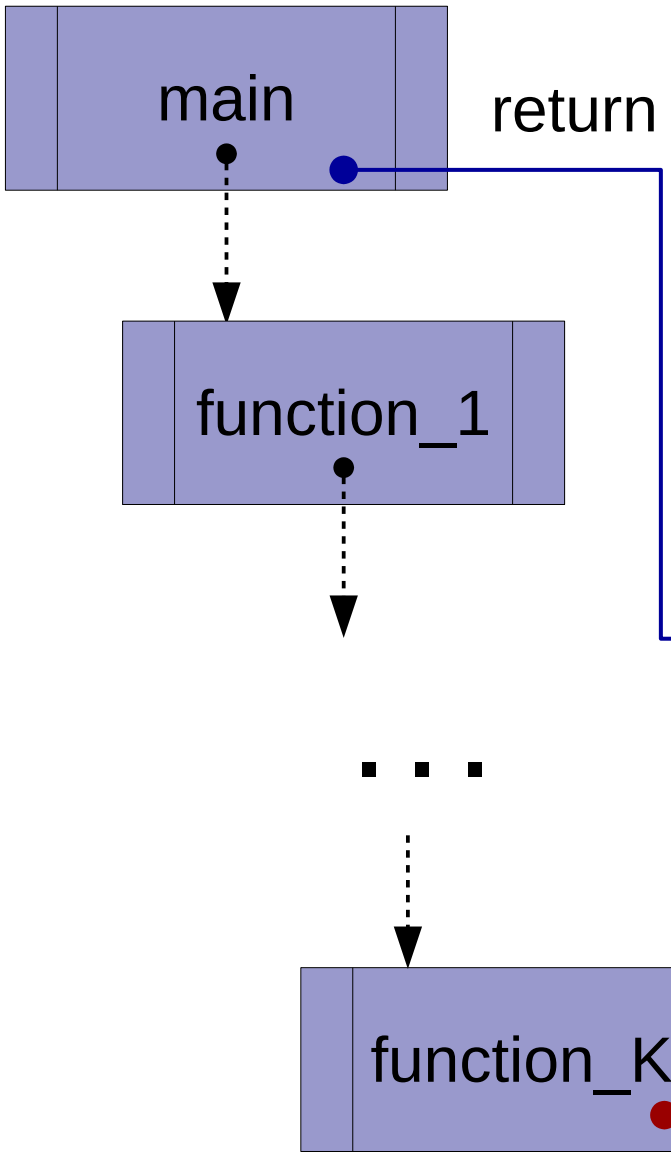
Прекратить выполнение программы



Прекратить выполнение программы



Прекратить выполнение программы

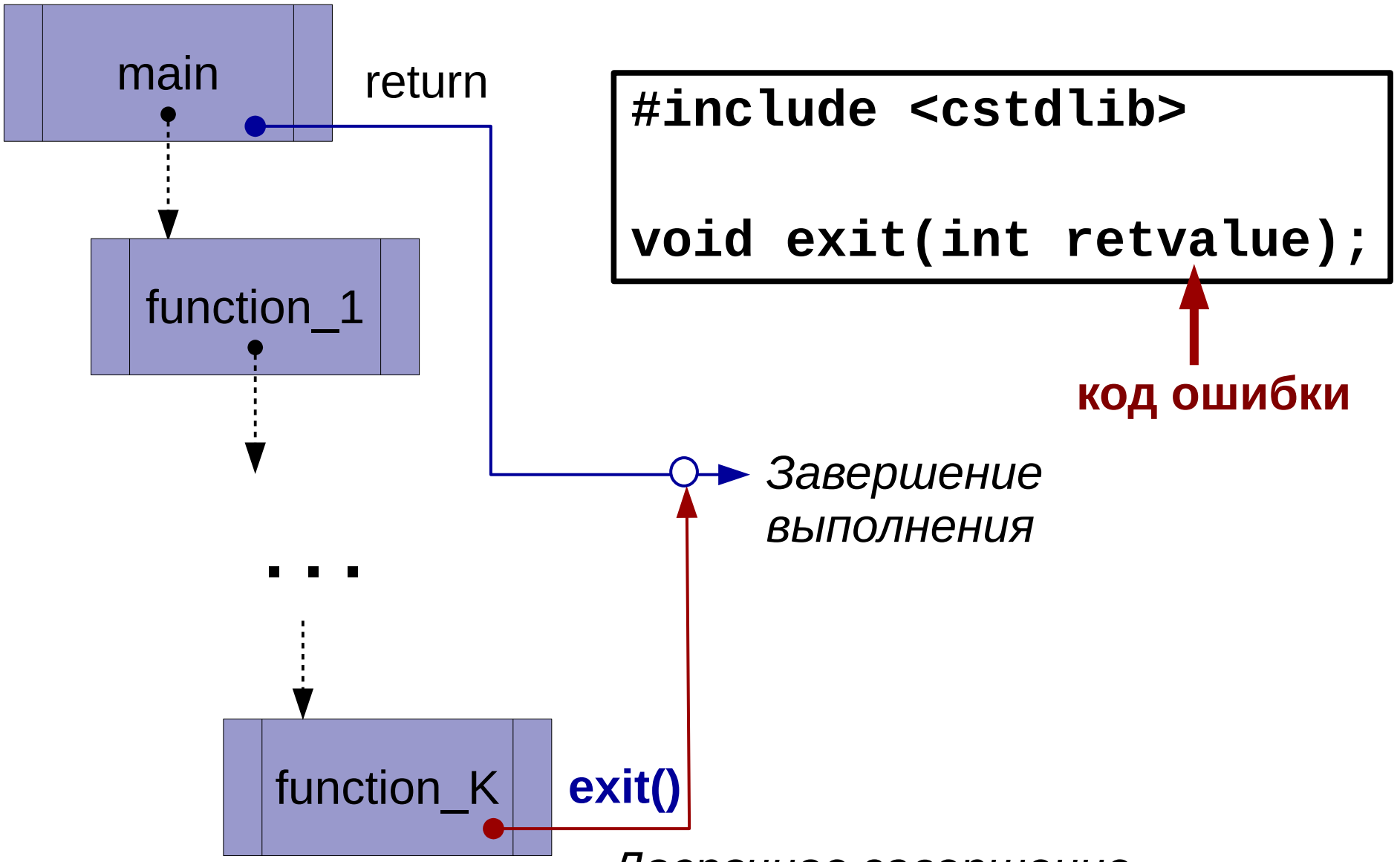



```
#include <cstdlib>

void exit(int retvalue);
```

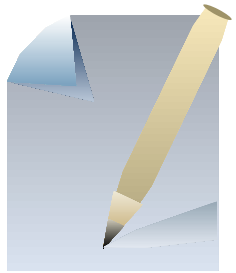
Завершение выполнения

Досрочное завершение выполнения

Прекратить выполнение программы

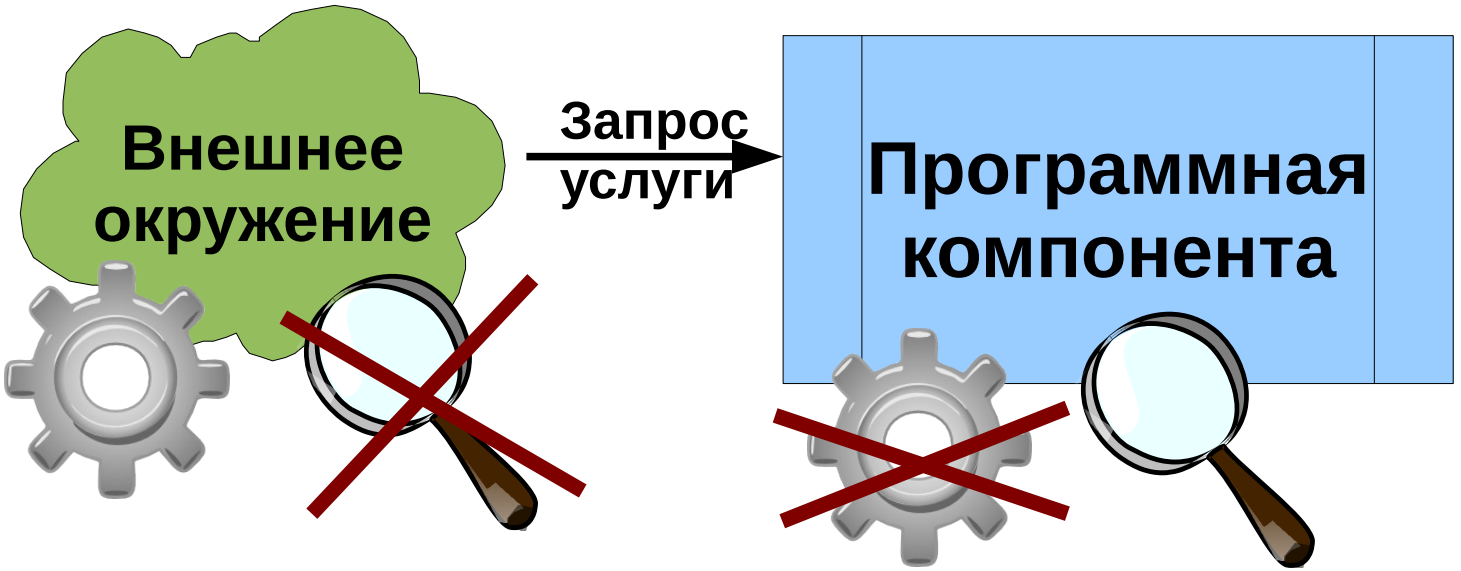


Пример



```
double svScalar(const SpatialVector& first,
                const  SpatialVector& second) {

    if( first.size() != second.size() )
       exit(1);

    //...
}
```

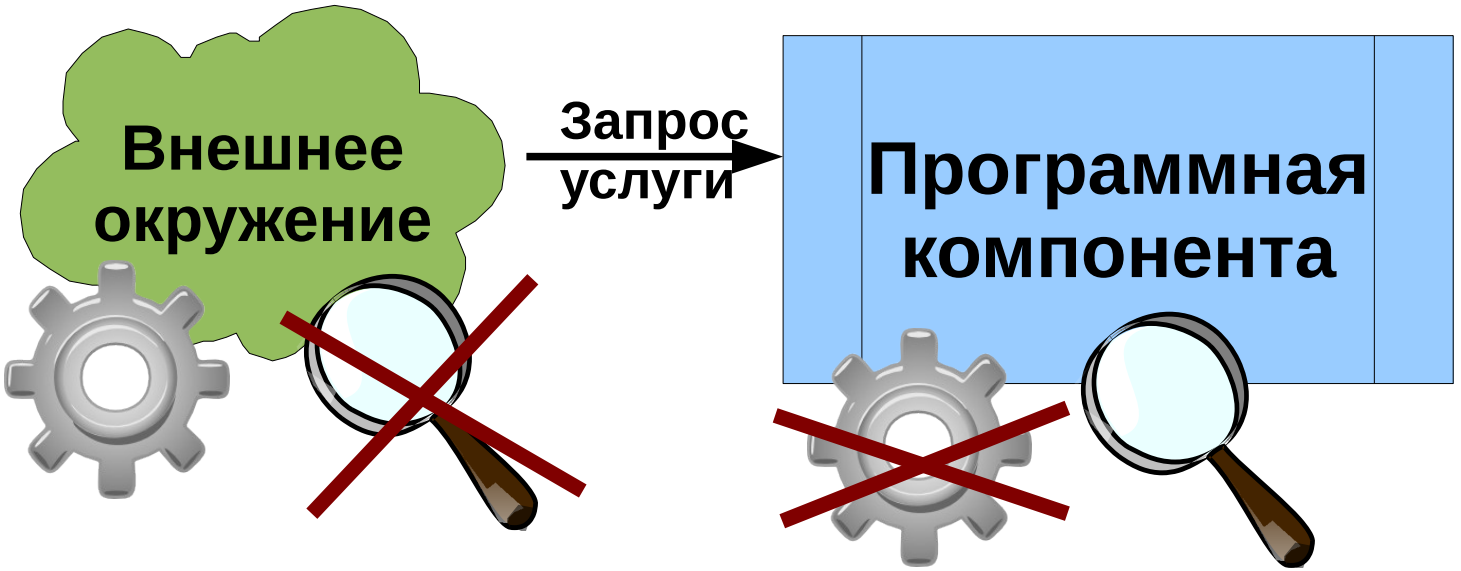
Обработка особых ситуаций



Прекратить выполнение программы

«Надеяться на лучшее»

Обработка особых ситуаций

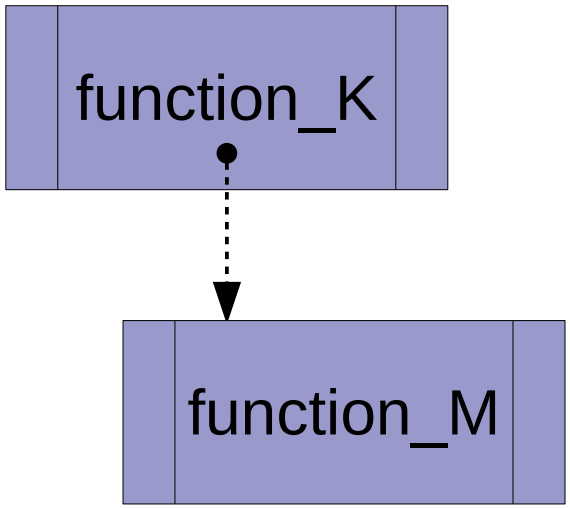


Прекратить выполнение программы

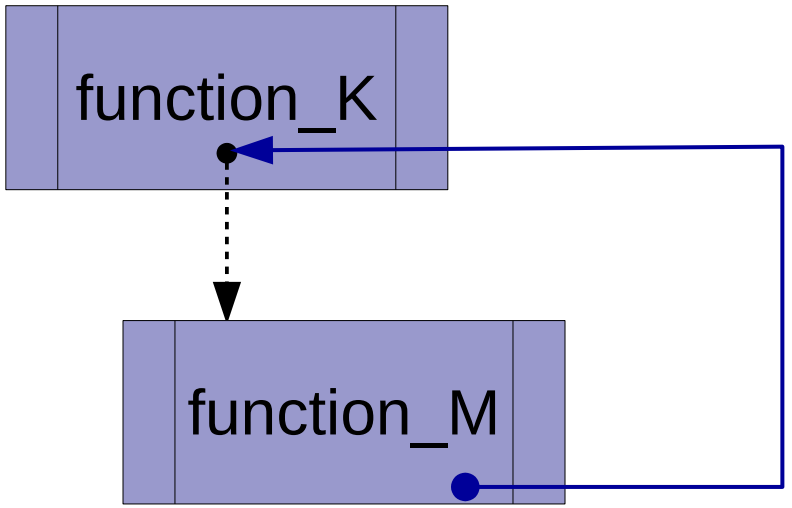
«Надеяться на лучшее»

Вернуть признак ошибки

Вернуть признак ошибки

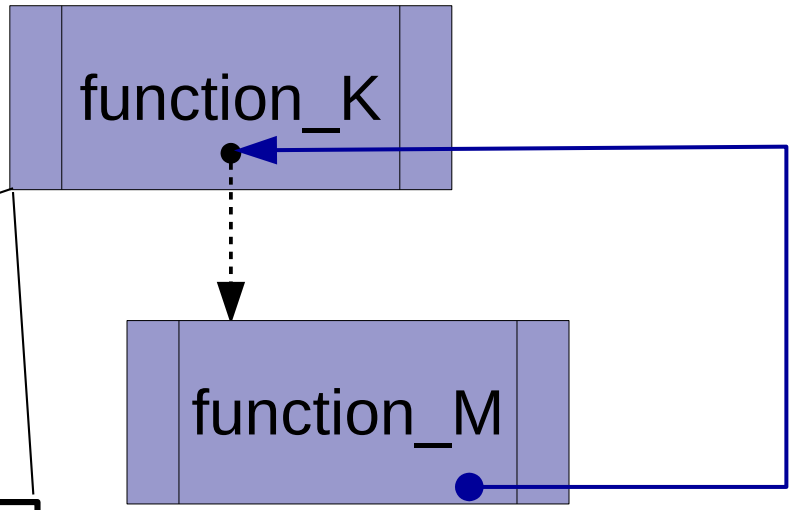


Вернуть признак ошибки



return код_ошибки

Вернуть признак ошибки

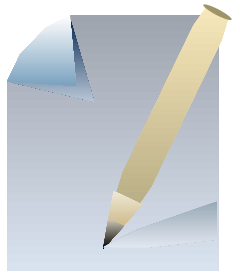


return код_ошибки

```
int errcode =
function_M(...);
switch (errcode) {
   case 0: break;
   case err_1:
      /*некоторые действия*/
      break;
   case err_2:
      /*некоторые действия*/
      break;
   . . .
}
```

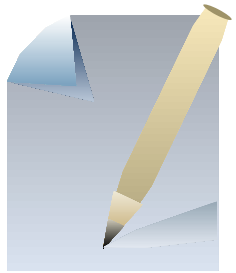
Пример (1)

```
int vectorScalar(const Vector& first,
    const Vector& second, double& result)
{
    if( first.coordinates.size() !=
        second.coordinates.size() )
       return -1;
    result = 0;
    //...
    return 0;
}
```



Пример (1)

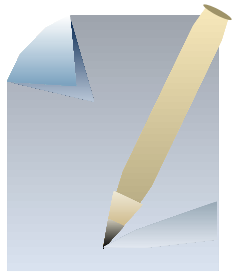
```
int vectorScalar(const Vector& first,
    const Vector& second, double& result)
{
    if( first.coordinates.size() !=
        second.coordinates.size() )
       return -1;
    result = 0;
    //...
    return 0;
}
```



```
static int appScalarProduct(Application& app, double& result)
{
   return vectorScalar(app.first,app.second,result);
}
```

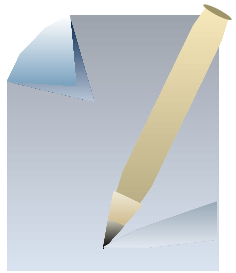
Пример (2)

```
int appRun(Application& app)
{
    //...
    double scalar;
```

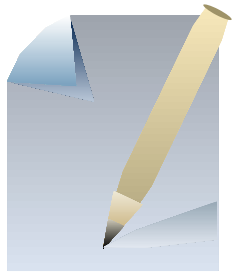


Пример (2)

```
int appRun(Application& app)
{
    //...
    double scalar;
    int errcode = appScalarProduct(app,scalar);
```

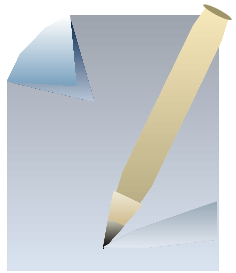


Пример (2)



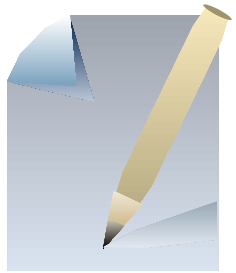
```
int appRun(Application& app)
{
    //...
    double scalar;
    int errcode = appScalarProduct(app,scalar);
    switch ( errcode ) {
```

Пример (2)



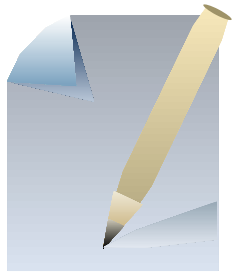
```
int appRun(Application& app)
{
    //...
    double scalar;
    int errcode = appScalarProduct(app,scalar);
    switch ( errcode ) {
       case 0:
          appOutputResult(app,scalar);
          break;
```

Пример (2)



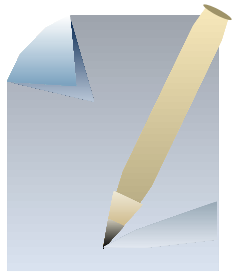
```
int appRun(Application& app)
{
    //...
    double scalar;
    int errcode = appScalarProduct(app,scalar);
    switch ( errcode ) {
       case 0:
          appOutputResult(app,scalar);
          break;
       case -1:
          cout << "несогласованные размеры векторов" << endl;
          break;
```

Пример (2)



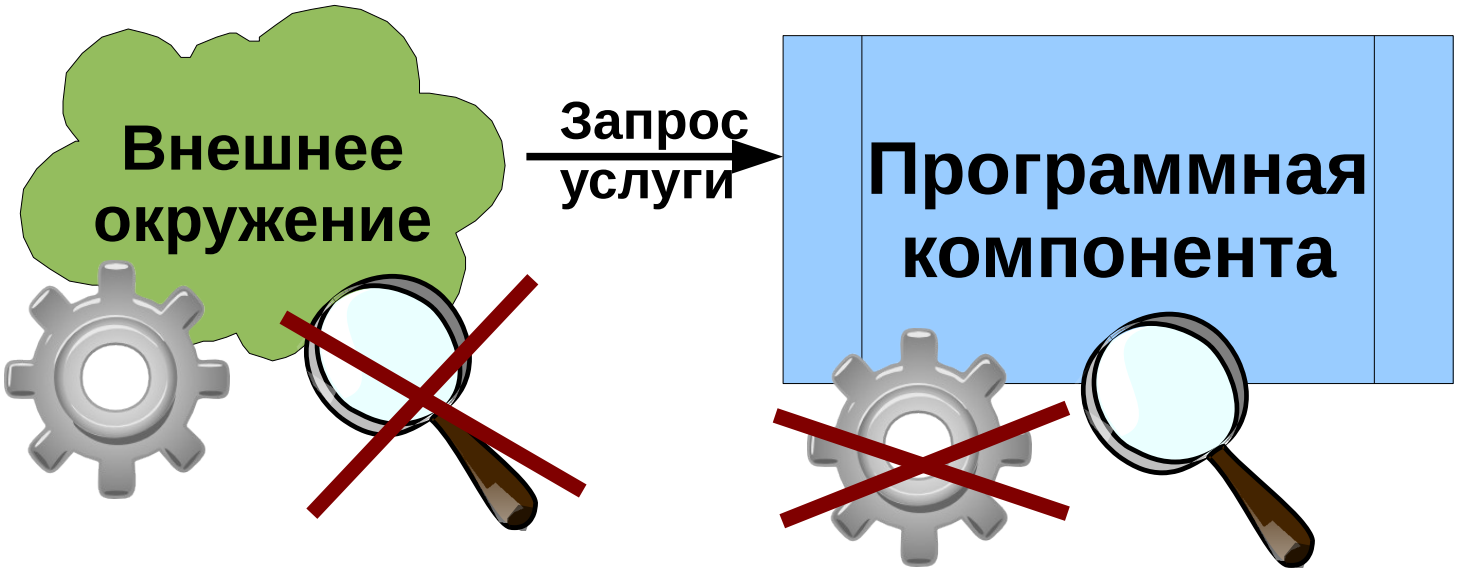
```
int appRun(Application& app)
{
    //...
    double scalar;
    int errcode = appScalarProduct(app,scalar);
    switch ( errcode ) {
       case 0:
          appOutputResult(app,scalar);
          break;
       case -1:
          cout << "несогласованные размеры векторов" << endl;
          break;
       default:
          cout << "неизвестная ошибка" << endl;
    }
```

Пример (2)



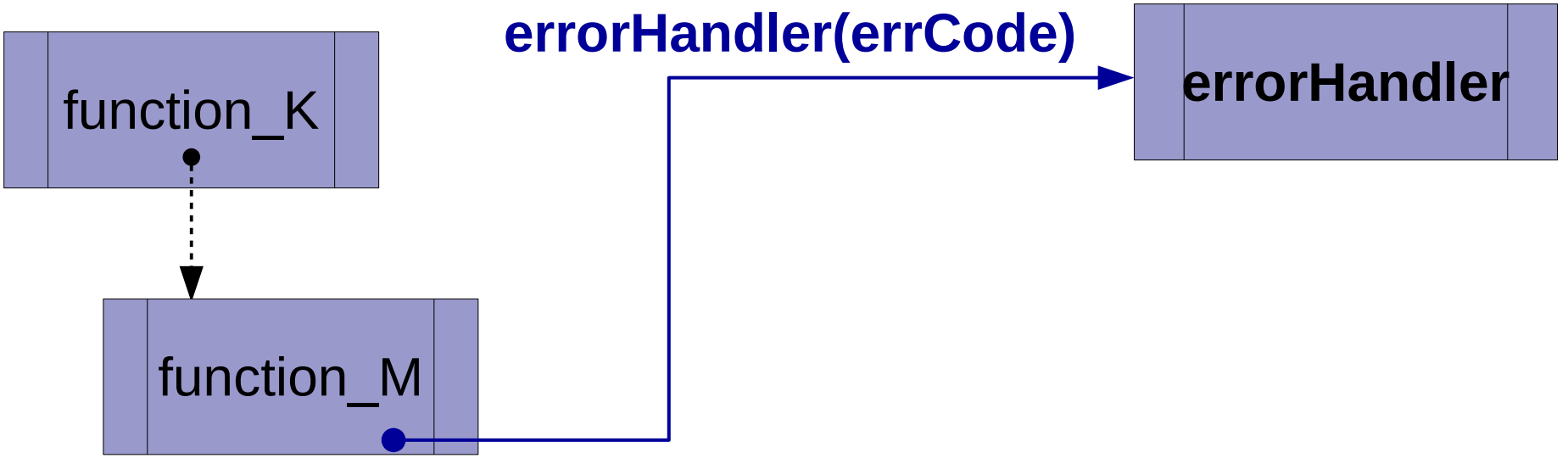
```
int appRun(Application& app)
{
    //...
    double scalar;
    int errcode = appScalarProduct(app,scalar);
    switch ( errcode ) {
       case 0:
          appOutputResult(app,scalar);
          break;
       case -1:
          cout << "несогласованные размеры векторов" << endl;
          break;
       default:
          cout << "неизвестная ошибка" << endl;
    }
    return 0;
}
```

Обработка особых ситуаций

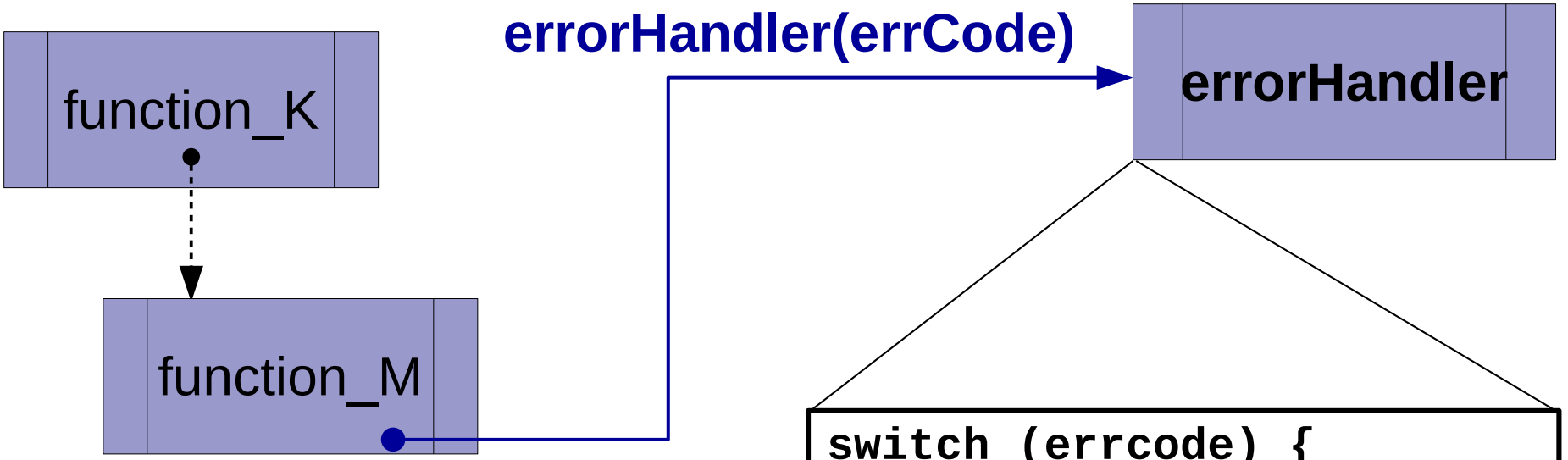


- Прекратить выполнение программы
- Вернуть признак ошибки
- «Надеяться на лучшее»
- Вызов функции обработки ошибок

Вызов функции обработки ошибок

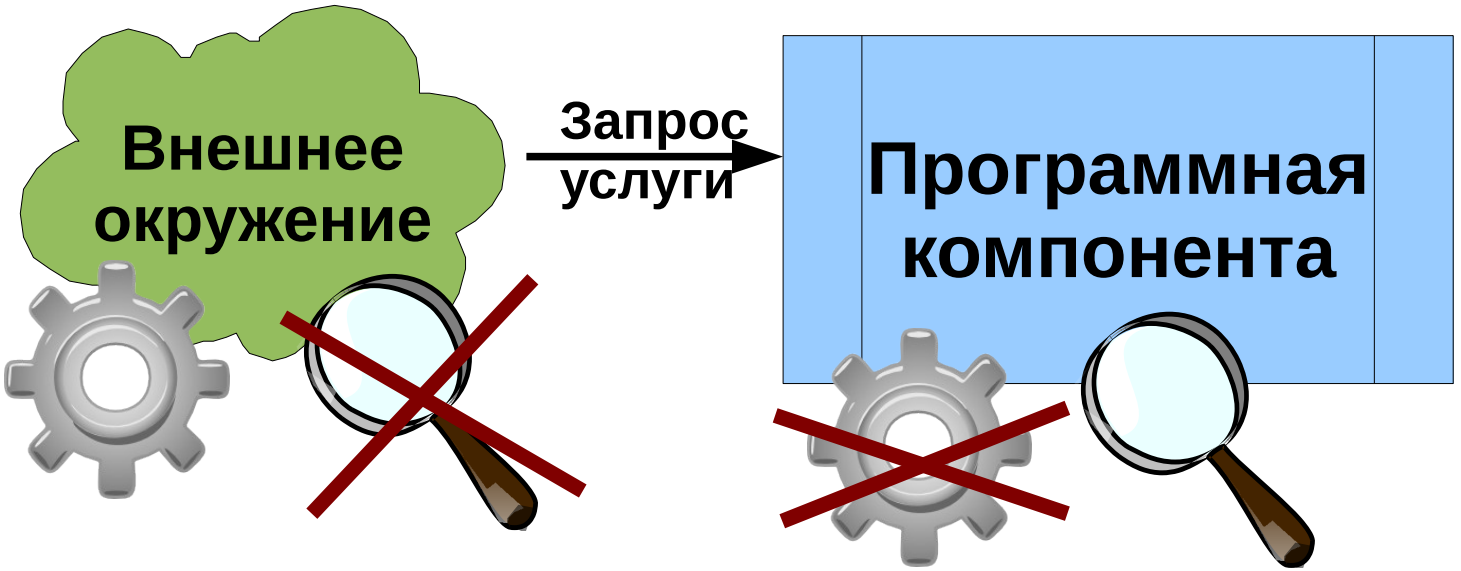


Вызов функции обработки ошибок




```
switch (errcode) {
   case 0: break;
   case err_1:
      /*некоторые
действия*/
      break;
   case err_2:
      /*некоторые
действия*/
      break;
   . . .
}
```

Обработка особых ситуаций



- Прекратить выполнение программы
- Вернуть признак ошибки
- «Надеяться на лучшее»
- Вызов функции обработки ошибок
- Возбудить исключительную ситуацию

Защитное программирование