



Передовые  
инженерные  
школы



МИНОБРНАУКИ  
РОССИИ



УНИВЕРСИТЕТ  
ИННОПОЛИС

# Занятие 8

## Отладка ошибок



# План занятия

1. Введение
2. Типы ошибок
3. Типы отладки
4. Пять шагов отладки
5. Методы отладки
6. Отладка в браузере
7. Отладка в IDE на примере VS Code

# Введение

**Отладка** – это процесс поиска и исправления ошибок или неполадок в исходном коде какого-либо программного обеспечения.

Дебаггинг – это процесс, который включает в себя выявление существующих или даже потенциальных ошибок в программе.



# Введение

Термин «баг» ранее использовался для описания механических неисправностей.

Первым человеком, который использовал его для обозначения проблем с вычислительной техникой, была адмирал Грейс Мюррей Хоппер в 1940-х годах.

Она работала в военно-морском флоте, была пионером компьютерной техники и работала над компьютерами Mark II и Mark III.

Мюррей Хоппер впервые использовала этот термин, когда она и ее команда обнаружили внутри своего компьютера моль, которая мешала его работе.

Таким образом, термин имеет очень буквальное происхождение!



# Типы ошибок

Существуют различные типы ошибок, которые могут привести к тому, что ваша программа не выполнит то, что ожидалось, или вообще не запустится.

Давайте узнаем:



## Синтаксические ошибки

Синтаксис относится к «правилам», которые определяют язык программирования.

Синтаксические ошибки, следовательно, являются результатом «нарушения» «правил» языка программирования.

К синтаксическим ошибкам относятся опечатки или пропуск точки с запятой.

## Логические ошибки

Логическая ошибка возникает, когда ваша программа работает не так, как должна. Тем не менее, программа может выполняться.



## Ошибки времени выполнения

Ошибка времени выполнения возникает, когда вы выполняете программу, но выполнение прекращается. Иногда ошибки времени выполнения могут быть вызваны средой, в которой выполняется программа.

Существует множество других типов ошибок, о которых вы узнаете по мере изучения кода. Еще лучше научиться «ловить» их в своем коде с помощью таких понятий, как «ошибки и обработка исключений».



# Типы отладки

Отладку можно в основном разделить на два типа:

- Реактивную
- проактивную.

Давайте копнем глубже и узнаем о них немного больше.



## Реактивная отладка

Выполняется после выявления ошибки. В предыдущем разделе мы рассмотрели различные типы ошибок. Все типы ошибок, с которыми мы сталкивались до сих пор, решаются с помощью реактивной отладки.



# Типы отладки

## Проактивная отладка

При проактивной отладке разработчики пишут код (как часть программы), который «следит» за ошибками. Этот «дополнительный» код не влияет на функциональность остальной части программы. Проактивную отладку также называют упреждающей отладкой.



# Пять шагов отладки

В стремлении действительно понять, что такое отладка, должен существовать способ сделать ее более управляемой и, кто знает, может быть, даже приятной.

пять простых шагов для этого:



# Пять шагов отладки

## Шаг 1: Соберите информацию об ошибке

Как начинающий программист, вы с большой вероятностью будете допускать синтаксические ошибки. Поэтому следить за ними должно быть одним из ваших приоритетов.

Еще одна вещь, которую вам необходимо делать, - это читать сообщения об ошибках и пытаться понять их.

Если вы работаете с какими-либо данными, например, пользовательскими, проверьте и их. Возможно, вы забыли выполнить проверку достоверности данных.

## Шаг 2: Изолировать ошибку

На этом этапе цель состоит в том, чтобы попытаться определить источник ошибки.

Возможно, вам придется «воссоздать» ошибку, чтобы узнать, когда она возникает. Можно также закомментировать часть кода и продолжать проверять, какая часть программы может быть причиной ошибки.

Вы также можете добавить в код операторы печати (`console.log` в JavaScript). Например, если вы можете распечатать вывод функции, то, скорее всего, ошибка вызвана не ею. Продолжайте делать это после каждых нескольких строк кода. В конце концов, вы увидите ту часть кода, которая вызывает ошибку.

# Пять шагов отладки

## Шаг 3: Определите ошибку

После того как вы сузили круг разделов программы, вызывающих ошибку, необходимо найти саму ошибку. Лучше всего начать с гипотезы. Например, если вы обнаружили, что ошибка возникает в поле вашей формы, ваша гипотеза может заключаться в том, что вы используете неправильный тип данных. Например, вы пытаетесь добавить строки и целые числа.

# Пять шагов отладки

## Шаг 4: Определите, как исправить ошибку

Когда у вас есть гипотеза, пришло время проверить ее. Например, вы можете добавить в программу логику преобразования данных и запустить ее, чтобы проверить, исчезла ли ошибка.

Если ошибка не исчезла, сформулируйте другую гипотезу и продолжайте ее проверять. Продолжайте делать это до тех пор, пока ошибка не будет устранена.

Иногда решение ошибки может быть неочевидным. Вы можете посетить такие сайты, как `StackOverflow`, и узнать, сталкивался ли кто-нибудь с такой же ошибкой и как он ее исправил.



## Шаг 5: Применить и протестировать

Вы можете повторно запустить код, чтобы проверить, исправлена ли ошибка. Тестирование – это отличный способ убедиться, что вы не создадите новых ошибок при исправлении тех, на которых вы сосредоточились!

Иногда вы не можете понять, как исправить ошибку, несмотря на все ваши усилия и даже усилия ваших коллег. В таких ситуациях вам придется довольствоваться обходным решением, которое позволит сохранить работоспособность программы, пока вы продолжаете искать способы полностью избавиться от ошибки.

Важная вещь, о которой вы можете забыть, – это документирование вашего решения. Да, исправление ошибки – это большое достижение. Но документирование того, как вы устранили ошибку, поможет другим членам вашей команды, которые могли столкнуться с этой ошибкой, или даже вам самим в будущем.

# Методы отладки

Отладка часть работы. Как часто? Ну, это может происходить каждые восемь минут и продолжаться от нескольких минут до .....



## Активная отладка

Эта техника заключается в уменьшении количества строк кода, которые вы собираетесь отлаживать. Один из способов сделать это - закомментировать части кода, пока вы не доберетесь до участка, вызывающего ошибку.



## Добавление операторов печати

Вы можете добавлять операторы печати через каждые несколько строк кода или даже через строку кода, чтобы попытаться сузить раздел, вызывающий ошибку.



## Использование отладчика

Не всегда нужно искать все ошибки самостоятельно. Вы можете прибегнуть к помощи отладчика (программное обеспечение, помогающее найти ошибки).

VS Code, например, имеет встроенный отладчик, который поможет вам найти ошибки в приложениях Node.js. Он также работает с JavaScript, TypeScript и другими языками, которые могут быть преобразованы в JavaScript. Для других языков вам потребуется установить расширения из VS Code Marketplace.

## Использование точек остановки

Это точки в коде, в которых выполнение останавливается – не потому, что есть ошибка, а потому, что вы «поставили паузу».

Это буквально «пауза» в коде, чтобы вы могли найти ошибки.



## Написание тестов

Написание тестов – это проактивная отладка.

При написании тестов разработчики пишут простую программу, которая имитирует действия более крупной программы. Эта программа и называется «тестом».

Затем они запускают ее (используя библиотеку типа Mocha или Jasmine для JavaScript), чтобы увидеть все ошибки, которые могут возникнуть. Затем ошибки исправляются до тех пор, пока код не заработает. По сути, вы создали «пример» кода, который вы должны написать.



Передовые  
инженерные  
школы



МИНОБРНАУКИ  
РОССИИ



УНИВЕРСИТЕТ  
ИННОПОЛИС

# Спасибо за внимание

