



Ву! ИИ готовит к ЕГЭ по информатике [Попробовать](#)

Урок Спринт 2.3

Debugger

Go Delve — это мощный инструмент для отладки программ на Go. Он предоставляет расширенные возможности для контроля выполнения программы, анализа переменных и стека вызовов.

Вот некоторые особенности и возможности Go Delve:

- **Точки останова:** Вы можете установить точки останова в своем коде, чтобы остановить выполнение программы и исследовать состояние переменных и стек вызовов на этой точке.
- **Шаги выполнения:** Go Delve предоставляет возможность выполнять шаги программы по одной инструкции или по функциям, чтобы следить за ее выполнением и исследовать различные состояния.
- **Анализ переменных:** Вы можете просматривать значения переменных в текущем контексте выполнения программы. Go Delve также позволяет изменять значения переменных во время выполнения для проверки различных сценариев.
- **Стек вызовов:** Вы можете просматривать стек вызовов, чтобы увидеть, какие функции были вызваны и в каком порядке. Это может быть полезно для отслеживания пути выполнения программы.
- **Управление горутинами:** Go Delve позволяет управлять горутинami в вашей программе. Вы можете переключаться между горутинami, останавливать и возобновлять их выполнение, а также анализировать их состояние и стеки вызовов.

Для использования Go Delve вам понадобится его установить. Выполните следующие шаги:

1. Установите Go Delve, выполнив следующую команду в командной строке:

```
go get github.com/go-delve/delve/cmd/dlv
```

1. После успешной установки Go Delve вы можете использовать его для отладки ваших программ на Go. Например, чтобы запустить программу в режиме отладки, выполните следующую команду:

```
dlv debug имя_файла.go
```

1. После запуска программы в режиме отладки вы сможете использовать команды Go Delve для управления выполнением программы и анализа ее состояния.

Некоторые полезные команды Go Delve:

- `break` — установить точку останова
- `continue` — продолжить выполнение программы
- `next` — выполнить следующую инструкцию
- `step` — выполнить шаг внутри функции
- `print` — вывести значение переменной
- `stack` — просмотреть стек вызовов
- `goroutines` — просмотреть информацию о горутинах
- `quit` — выйти из режима отладки

Можете изучить полный список команд и их синтаксис в [официальной документации Go Delve](#).

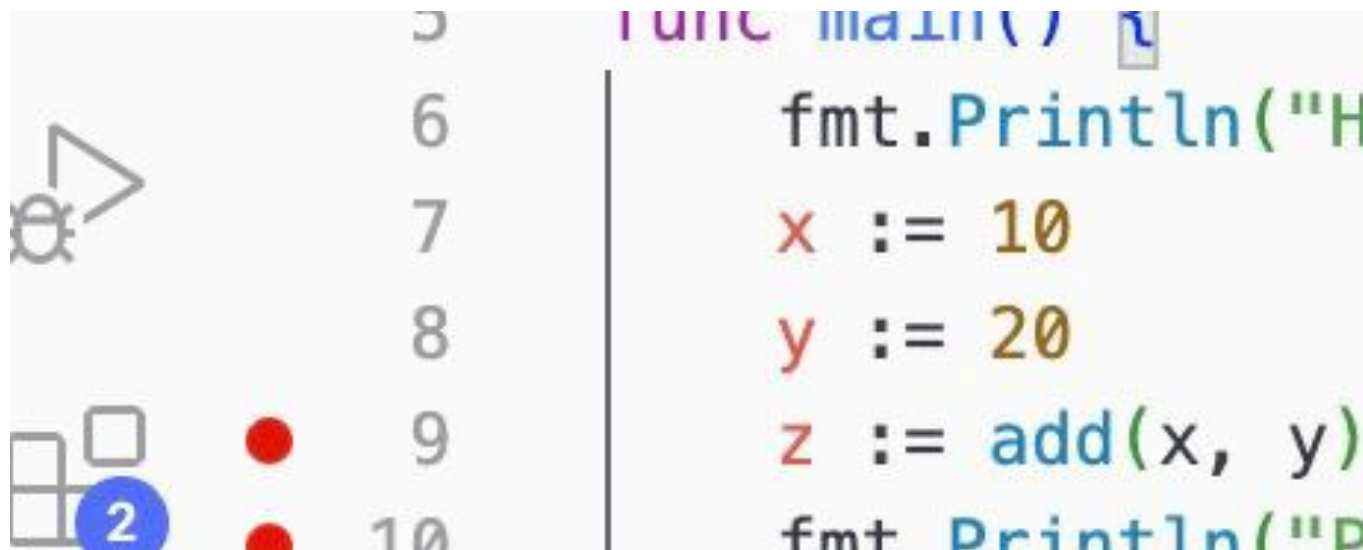
Так же мы можем запустить его прямо из среды разработки.

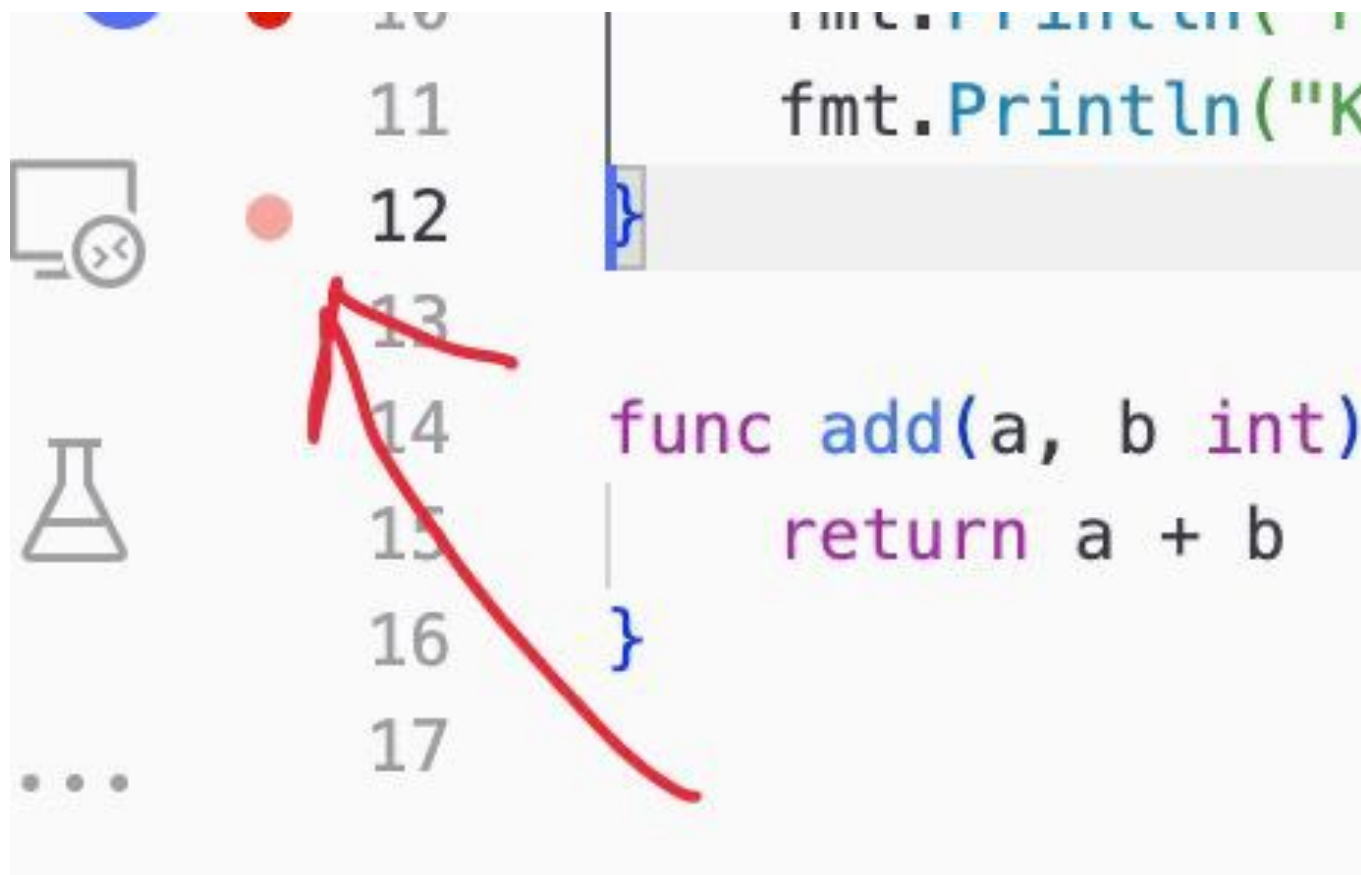
Для запуска Go Delve из Visual Studio Code (VS Code) вам потребуется установить и настроить несколько расширений. Вот пошаговая инструкция

Запустите VS Code в корневой папке вашего проекта Go.

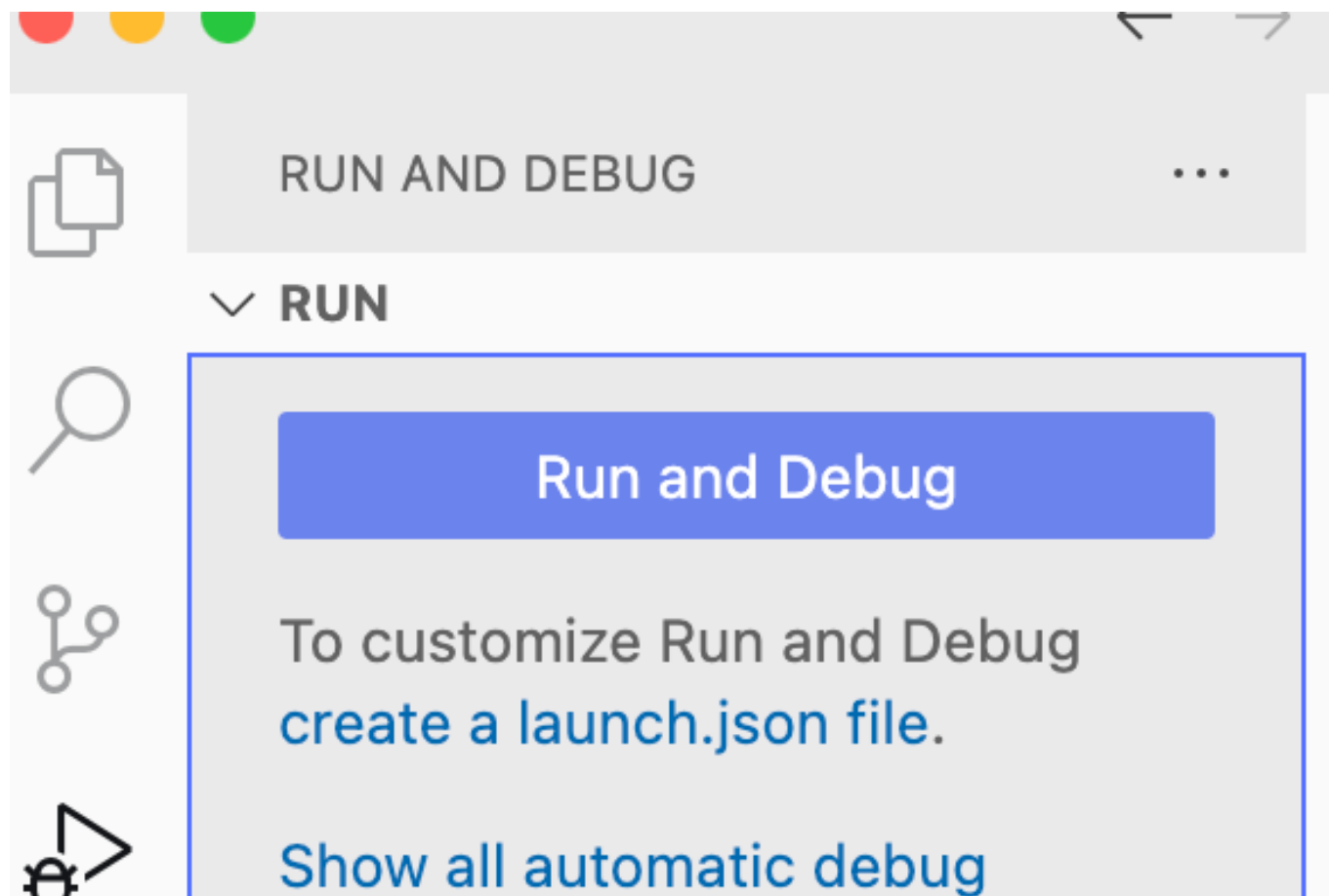
Откройте файл с кодом Go, который вы хотите отладить.

Установите точку остановки в нужном месте вашего кода. Чтобы установить точку останова, щелкните левее строки кода, где хотите установить точку останова. Вы увидите красный кружок, указывающий на установленную точку останова.

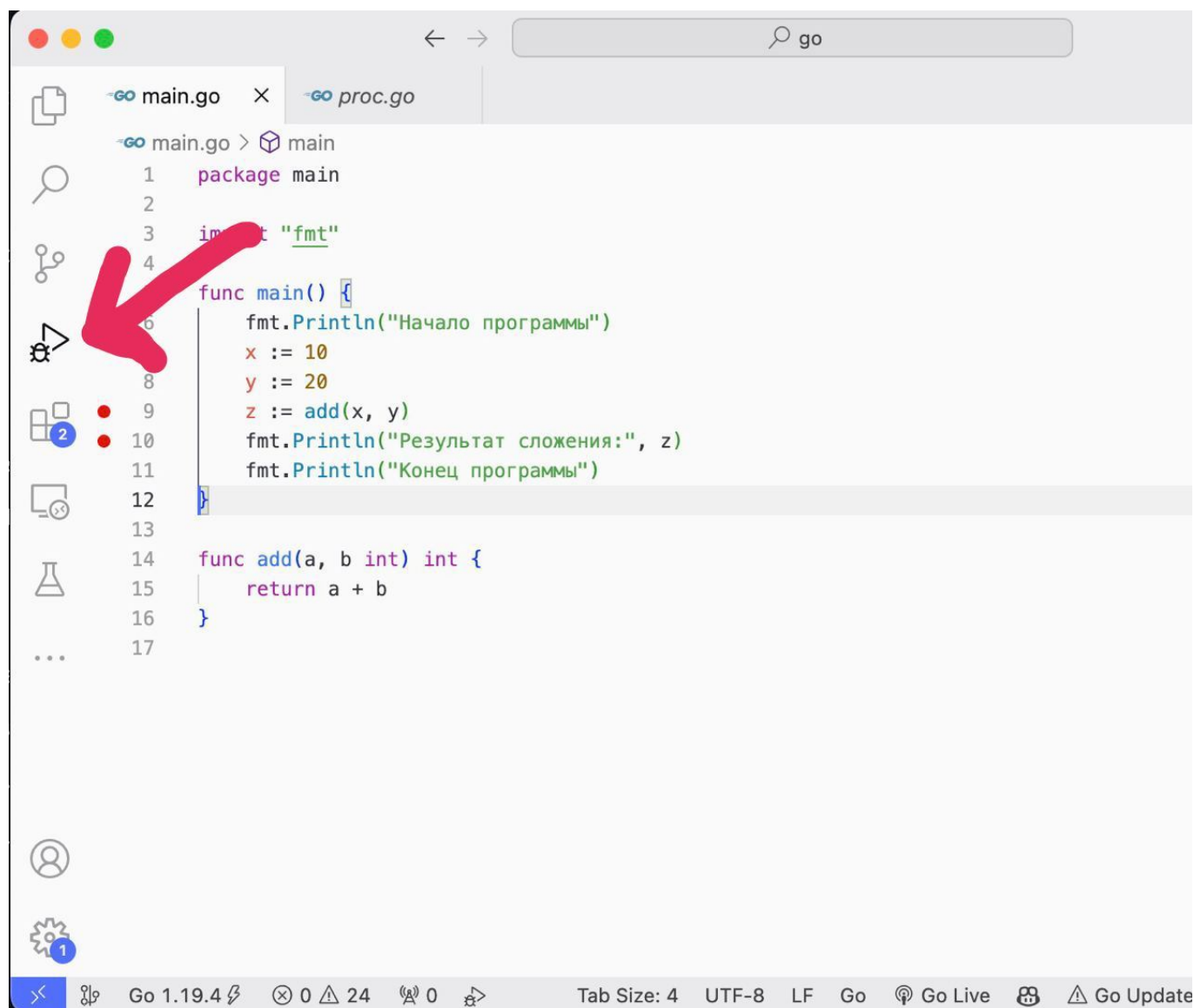




Нажмите кнопку "Debug" на левой панели VS Code (или нажмите `Ctrl+Shift+D`). Откроется панель отладки.

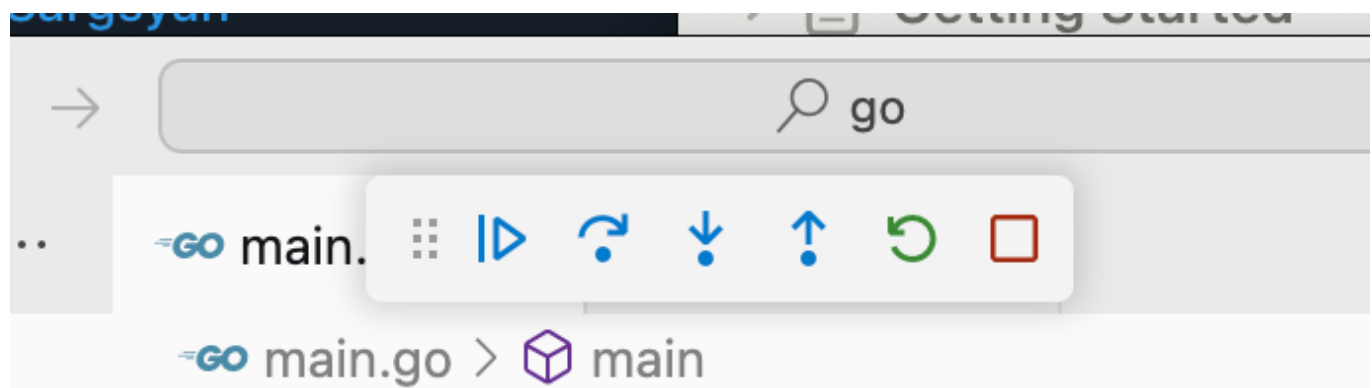


В выпадающем списке конфигураций отладки выберите "Go" и нажмите кнопку "Run" (зеленая стрелка).



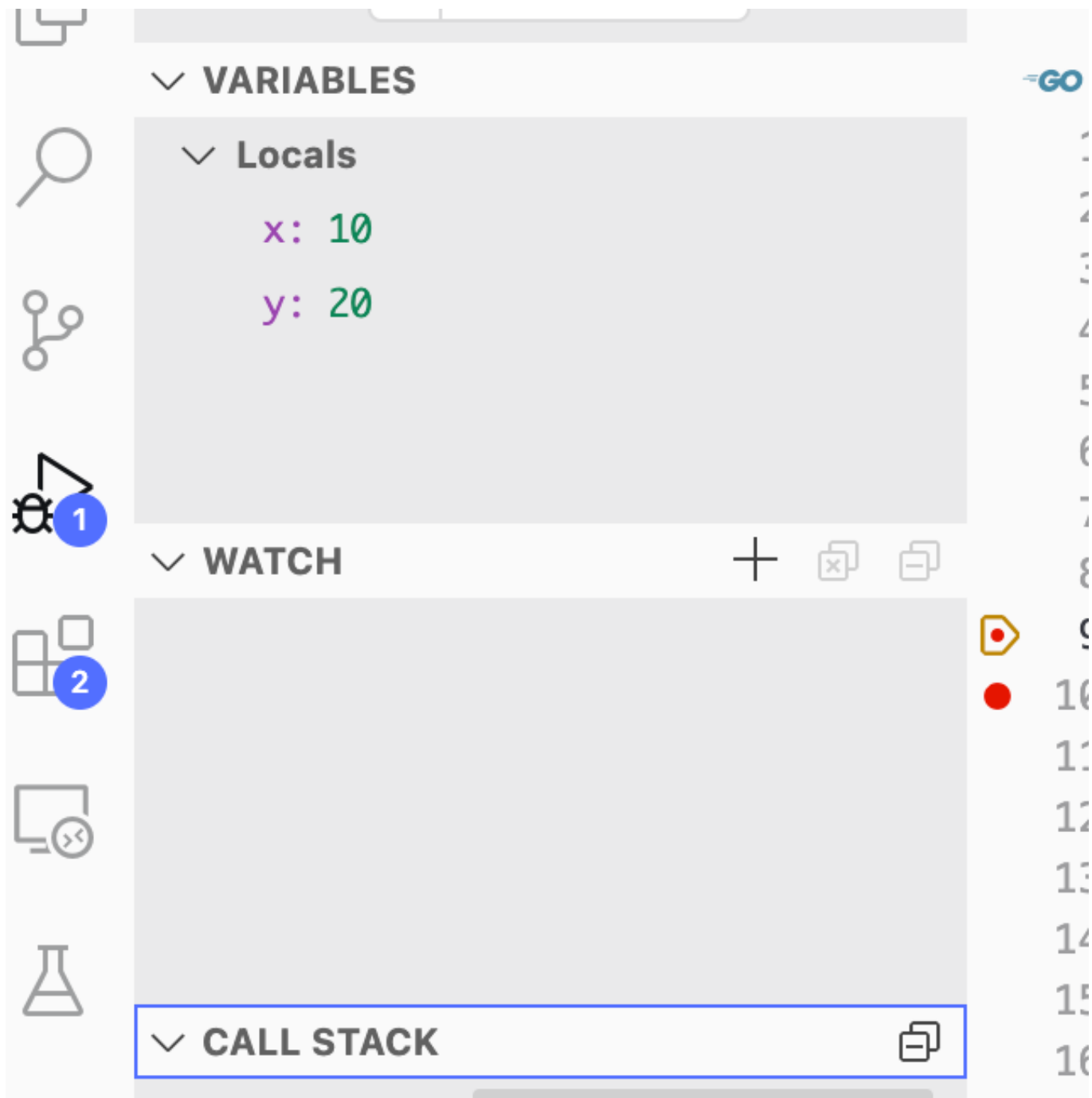
Ваша программа Go будет запущена в режиме отладки, и выполнение остановится на установленной точке останова.

Используйте кнопки управления на панели отладки (перезапуск, продолжение выполнения, шаги выполнения и т.д.) для управления выполнением программы и анализа ее состояния.



```
1 package main
2
3 import "fmt"
4
5 func main() {
```

В окне "Variables" вы можете посмотреть значения переменных в текущем контексте выполнения программы. Если у вас есть точка останова на этой строке кода, вы увидите значения переменных в этой точке останова.





В ходе данного урока мы изучили несколько полезных инструментов для отладки и анализа программ на Go. Один из них — дампы горутин, который помогает нам разобраться в том, какие горутинны активны в данный момент и как они взаимодействуют друг с другом. Мы рассмотрели пример кода, который позволяет получить дампы горутин в Go.

Другой важный инструмент — Race Detector или детектор гонок. Гонки данных возникают, когда несколько горутин обращаются к общим данным без синхронизации, что может привести к неопределенному поведению программы. Race Detector позволяет обнаруживать гонки данных и выводить предупреждения о них. Мы рассмотрели пример кода, который триггерит Race Detector и изучили вывод, который мы получаем при его запуске.

Наконец, мы ознакомились с инструментом Go Delve — мощным инструментом для отладки программ на Go. Он предоставляет расширенные возможности для контроля выполнения программы, анализа переменных и стека вызовов. Мы изучили основные функции Go Delve, такие как установка точек останова, выполнение шагов программы, анализ переменных и стека вызовов, а также управление горутинными.

В результате урока мы поняли, что использование этих инструментов может значительно облегчить процесс отладки и анализа программ на Go. Дампы горутин позволяют лучше понять, как работают горутинны в программе, а Race Detector помогает выявлять и исправлять проблемы с гонками данных. Go Delve предоставляет мощные возможности для более детального анализа и контроля выполнения программы.

Справка

Исключительное право на учебную программу и все сопутствующие ей учебные материалы, доступные в рамках сервиса, принадлежат АНО ДПО «Образовательные технологии Яндекса». Воспроизведение, копирование, распространение и иное использование программы и материалов допустимо только с предварительного письменного согласия АНО ДПО «Образовательные технологии Яндекса».

[Пользовательское соглашение.](#)

© 2018 – 2024 ООО «Яндекс»