

Лекция 6. Разработка на Flutter

Полезные ссылки:

- Официальный сайт Dart;
- Программа Visual Studio Code.

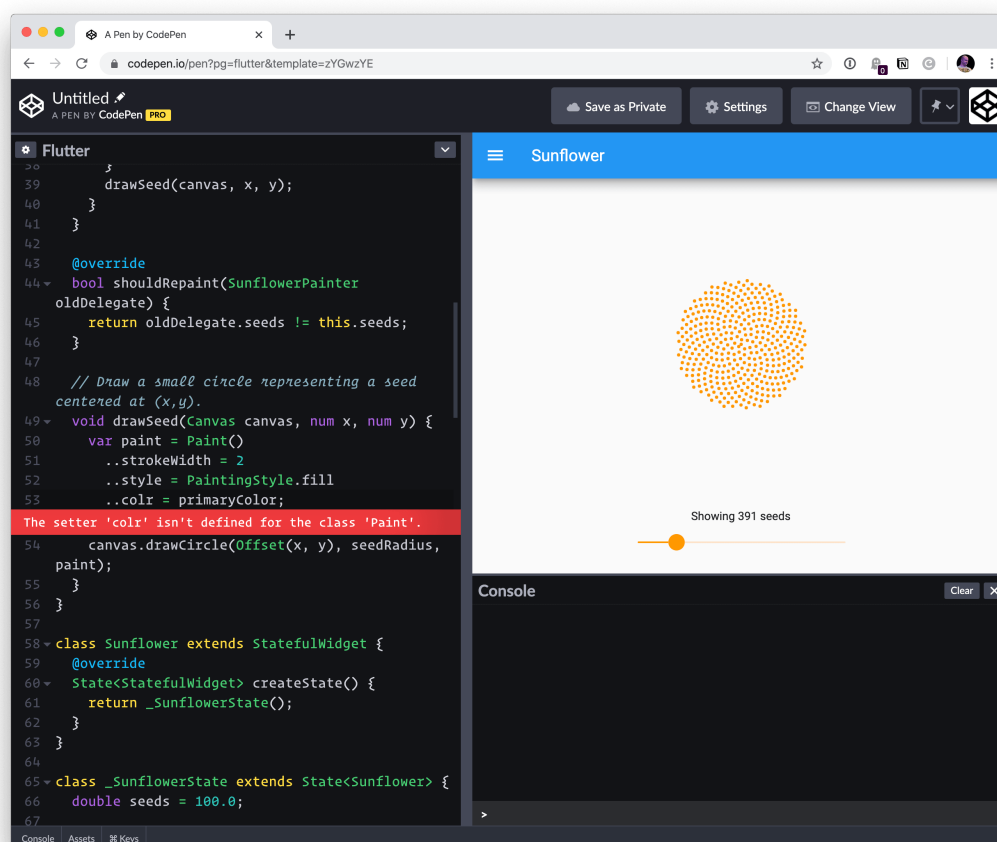
Flutter – что это такое?

Flutter – относительно новый фреймворк, что был разработан компанией Google. Его первое появление датируется 2015 годом. На основе Flutter любой разработчик может создать мобильное приложение как под Андроид, так и под iOS платформу. Помимо этого еще есть возможность создания веб приложений, но это уже совсем другая история.

В будущем планируется сделать поддержку разработки под компьютерные операционные системы: Windows, Mac, Linux и Google Fuchsia.

Что такое Dart?

Flutter – это лишь фреймворк, что в свою очередь написан на основе языков программирования C++ и Dart. Про язык Dart, вы, скорее всего, мало что знаете, так как язык появился лишь в 2011 году и был создан той же компанией Google. Основная цель данного языка – замена языка JavaScript. По этой причине Dart делает множество схожих действий с языком JavaScript, но делает это в более современном и верном стиле.



При разработке на Flutter весь код пишется на основе языка Dart. Если язык вам не знаком, то не переживайте, ведь мы будем его изучать в ходе курса и лишь после его изучения приступим к Flutter.

Удобство использования Dart в связке с Flutter также прослеживается при разработке проектов. Дело в том, что вы можете видеть все изменения по вашему проекту сразу же при сохранении проекта. Если говорить про стандартный способ разработки мобильных приложений, то чтобы увидеть изменения сперва нужно компилировать проект и запускать его на виртуальном устройстве.

Dart – язык программирования, что появился относительно недавно в 2011 году. Он был разработан компанией Google. Его предназначение стать альтернативой языку JavaScript. Dart выполняет множество действий, что в некоторой степени схожи с действиями из языка ДжаваСкрипт.

Основы языка

Любые программы на **Dart** состоят из нескольких компонентов:

- Подключение библиотек (по необходимости);
- Указание функции `main`, что ничего не возвращает.

Каждая строка должна закрываться точкой с запятой (;). В языке можно использовать все стандартные конструкции: переменные, списки, условия, циклы, функции и возможности ООП.

Переменные

Типы переменных в языке Дарт могут указываться перед названием переменной. Если это не сделать, то нужно прописывать ключевое слово `var`.

От конкретного значения будет зависеть тип данных переменной. Если мы укажем что переменная со значением 5, то тип данных будет указан для целых чисел. И наоборот, если указать тип данных для целых чисел `int`, то в неё мы не сможем записать строку или число с точкой.

В языке есть возможность создания динамических переменных, что смогут менять свой тип данных по ходу выполнения программы. Пример такой переменной:

```
dynamic some = 'text';  
some = 5;
```

Для создания констант необходимо прописывать ключевое слово **const** в начале переменной.

Прочие конструкции

Условия, списки, циклы и функции были рассмотрены в ходе видео урока. Можно заметить, что по большей части они не сильно отличаются от аналогичных конструкций из других языков программирования.

Если у вас уже есть навыки работы с другими языками программирования, то освоить Дарт вам не составит никакого труда.

Почему Flutter крут?

Предположим, вы решили разработать некий мобильный проект. Тут же возникает много вопросов. Какой язык изучить: Джава или же Swift, а может лучше и вовсе попробовать C++? А какой язык изучить, если нужно под несколько платформ разрабатывать?

И таких вопросов получается достаточно много. На все из них нужно ответить, прежде чем разработать мобильное приложение. Более того, ваше приложение, скорее всего, будет работать лишь под одну ОС.

Разработчики это понимают и стараются избавить других молодых разработчиков от этой головной боли. В качестве альтернативного решения они придумали технологию Flutter.

Flutter является полноценным современным решением для разработчика, что решил создать мобильное приложение.

- Во-первых, никаких сложных языков программирования изучать не придется. Язык Dart хоть и новый язык, но его синтаксис очень прост.
- Во-вторых, разработка будет происходить под каждую платформу сразу же. Теперь вы просто пишете один код и получаете приложение как под iOS, так и под Андроид.
- В-третьих, ваши мобильные проекты ничем не будут отличаться от стандартного подхода к разработке приложений.

Получается, что используя Flutter вы облегчаете себе процесс разработки, получаете сразу готовое решение под разные операционные системы, да и ко всему этому получаете массу готовых дополнительных библиотек, что можно использовать для быстрого добавления дополнительного функционала.

На что способен Flutter?

Flutter способен на многое. Начиная от создания простого проекта с одной страницей и до создания полноценного приложения, где у вас будет множество функций, красивый дизайн, безопасность и подключение к базе данных.

На Flutter уже было создано большое количество проектов. Можно выделить такие мобильные проекты:

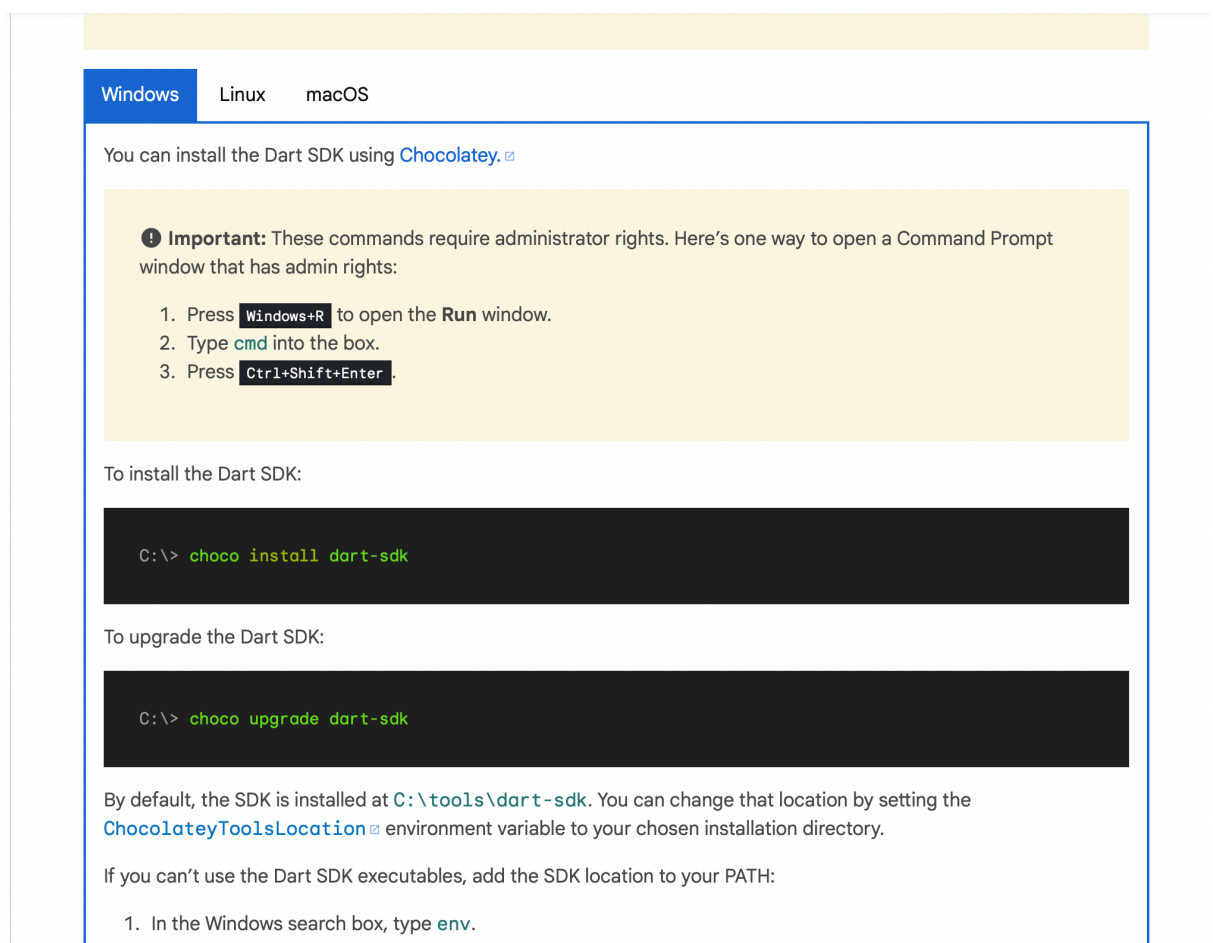
- Google Ads – сервис рекламы от Google;
- Alibaba – мобильное приложение всемирноизвестного магазина Alibaba;
- Reflectly – фитнес приложение;
- Birch Finance – финансовое приложение;
- Hamilton Musical – музыкальное приложение.

Flutter действительно является полноценным фреймворком для создания мобильных приложений, поэтому если вам нужно разработать проект, то смело можете выбирать данный фреймворк.

Сравнение с React Native

Сравнение платформ – дело сложное, ведь у каждой платформы есть свои фанаты и разработчики. Сказать что является более хорошей платформой очень сложно, ведь обе делают свое дело очень хорошо. **React Native** делает все те же функции, вот только он разработан компанией Facebook, а Flutter компанией Google. Сказать что из них лучше или хуже – очень сложно.

1) Install - <https://dart.dev/get-dart>



The screenshot shows the 'Windows' tab of the Dart SDK installation guide. It includes instructions on how to install the SDK using Chocolatey, with a note about requiring administrator rights. It provides a list of steps to open a Command Prompt with admin rights, followed by the command to install the SDK: `C:\> choco install dart-sdk`. It also shows the command to upgrade the SDK: `C:\> choco upgrade dart-sdk`. Finally, it mentions the default installation path and how to set the `ChocolateyToolsLocation` environment variable.

Windows Linux macOS

You can install the Dart SDK using [Chocolatey](#).

Important: These commands require administrator rights. Here's one way to open a Command Prompt window that has admin rights:

1. Press **Windows+R** to open the **Run** window.
2. Type `cmd` into the box.
3. Press **Ctrl+Shift+Enter**.

To install the Dart SDK:

```
C:\> choco install dart-sdk
```

To upgrade the Dart SDK:

```
C:\> choco upgrade dart-sdk
```

By default, the SDK is installed at `C:\tools\dart-sdk`. You can change that location by setting the [ChocolateyToolsLocation](#) environment variable to your chosen installation directory.

If you can't use the Dart SDK executables, add the SDK location to your PATH:

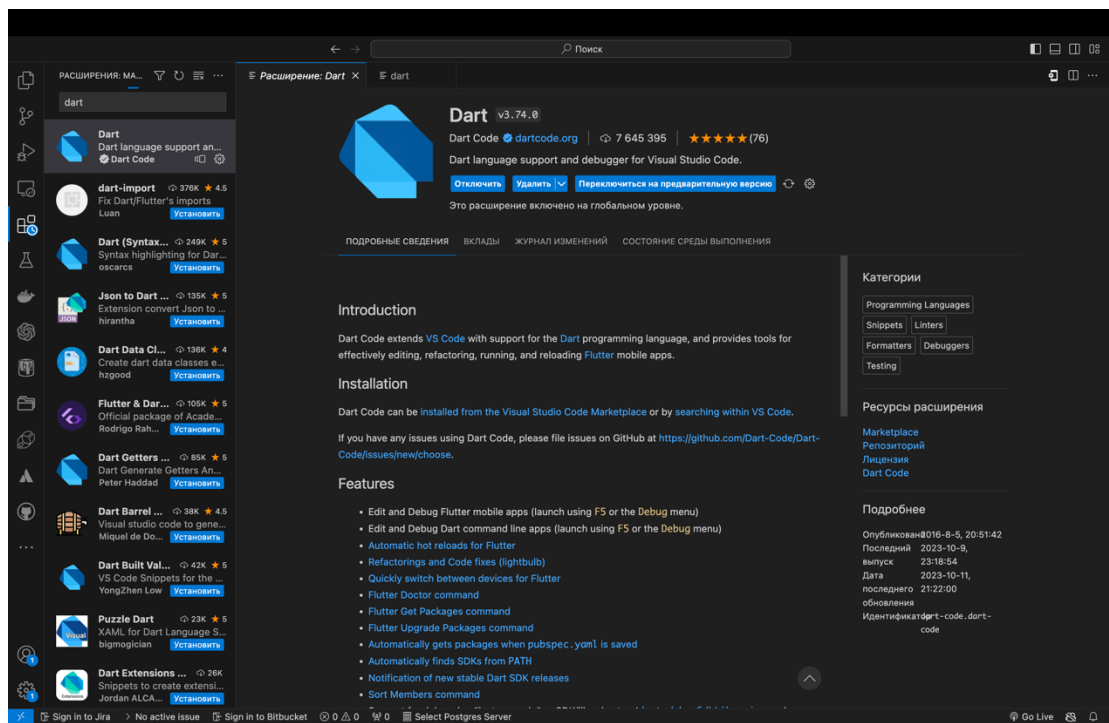
1. In the Windows search box, type `env`.

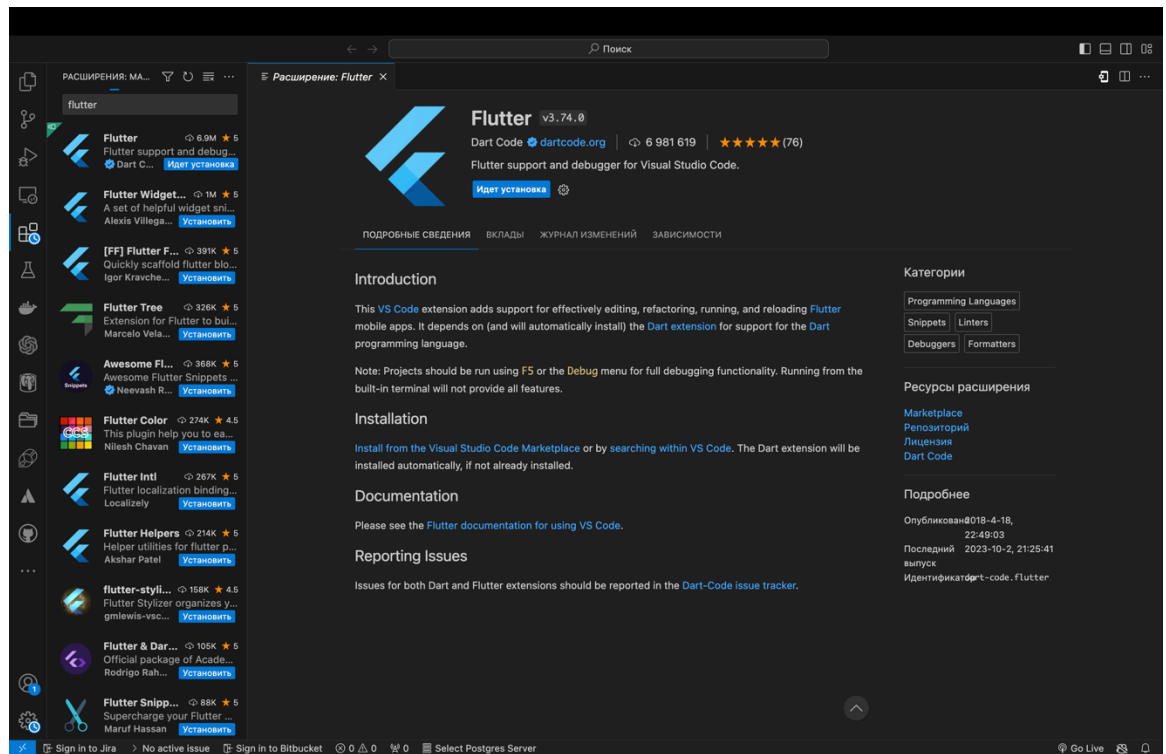
Mac OS Install:

```
ПРОБЛЕМЫ  ВЫХОДНЫЕ ДАННЫЕ  КОНСОЛЬ ОТЛАДКИ  ТЕРМИНАЛ  ПОРТЫ  КОММЕНТАРИИ
$ brew install dart
zsh: command not found: $
zsh: command not found: $
$ brew tap dart-lang/dart
Running 'brew update --auto-update'...
==> Auto-updated Homebrew!
Updated 4 taps (hashicorp/tap, homebrew/services, homebrew/core and homebrew/cask).
==> New Formulae
bashunit          incus          jupyter-r          netlistsvg          python-psutil
==> New Casks
kuaitie
low-profile
You have 37 outdated formulae installed.

==> Tapping dart-lang/dart
Cloning into '/opt/homebrew/Library/Taps/dart-lang/homebrew-dart'...
remote: Enumerating objects: 3784, done.
remote: Counting objects: 100% (1538/1538), done.
remote: Compressing objects: 100% (415/415), done.
remote: Total 3784 (delta 1282), reused 1326 (delta 1117), pack-reused 2246
Receiving objects: 100% (3784/3784), 682.42 KiB | 1.45 MiB/s, done.
Resolving deltas: 100% (2558/2558), done.
Tapped 22 formulae (54 files, 886.0KB).
$ brew install dart
==> Fetching dart-lang/dart/dart
==> Downloading https://storage.googleapis.com/dart-archive/channels/stable/release/3.1.3/sdk/dart-sdk-macos-arm64-release.zip
###
2.9%
```

Установка плагинов Dart и Flutter в VSCode



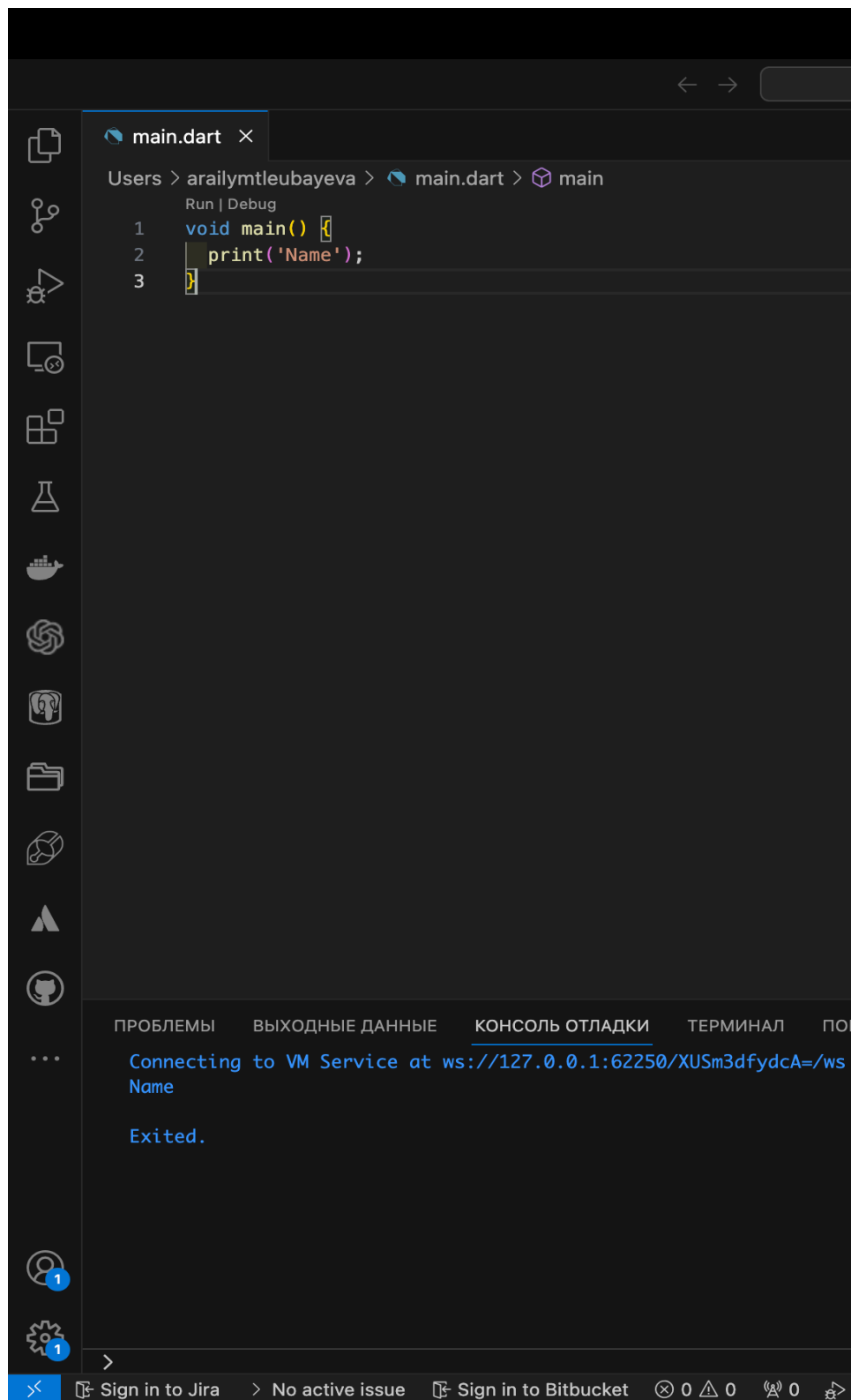


Изучение языка Dart. Основные концепции

Примеры кода Dart:

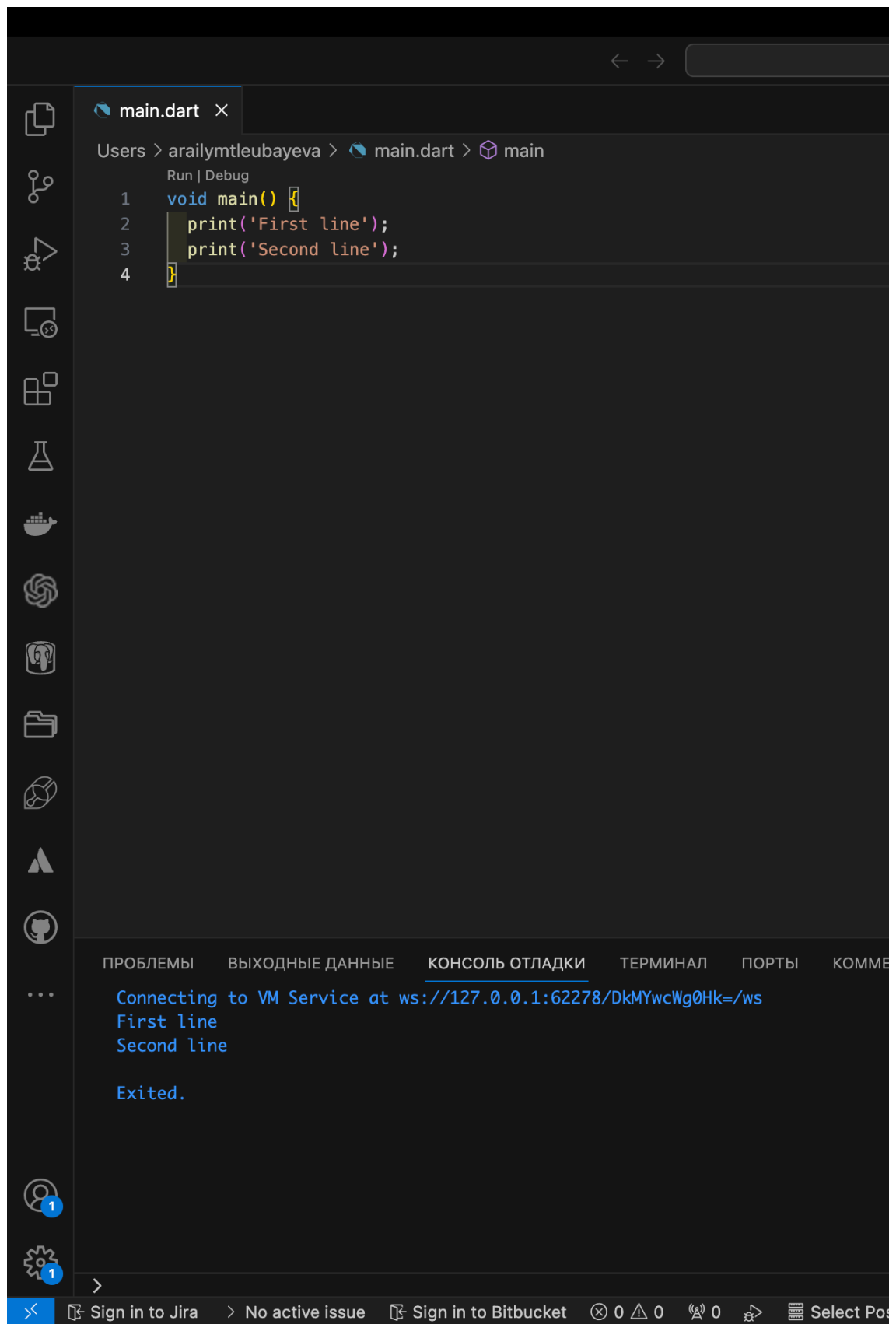
1) Вывод информации

Напишите программу, что выведет ваше имя в консоль.

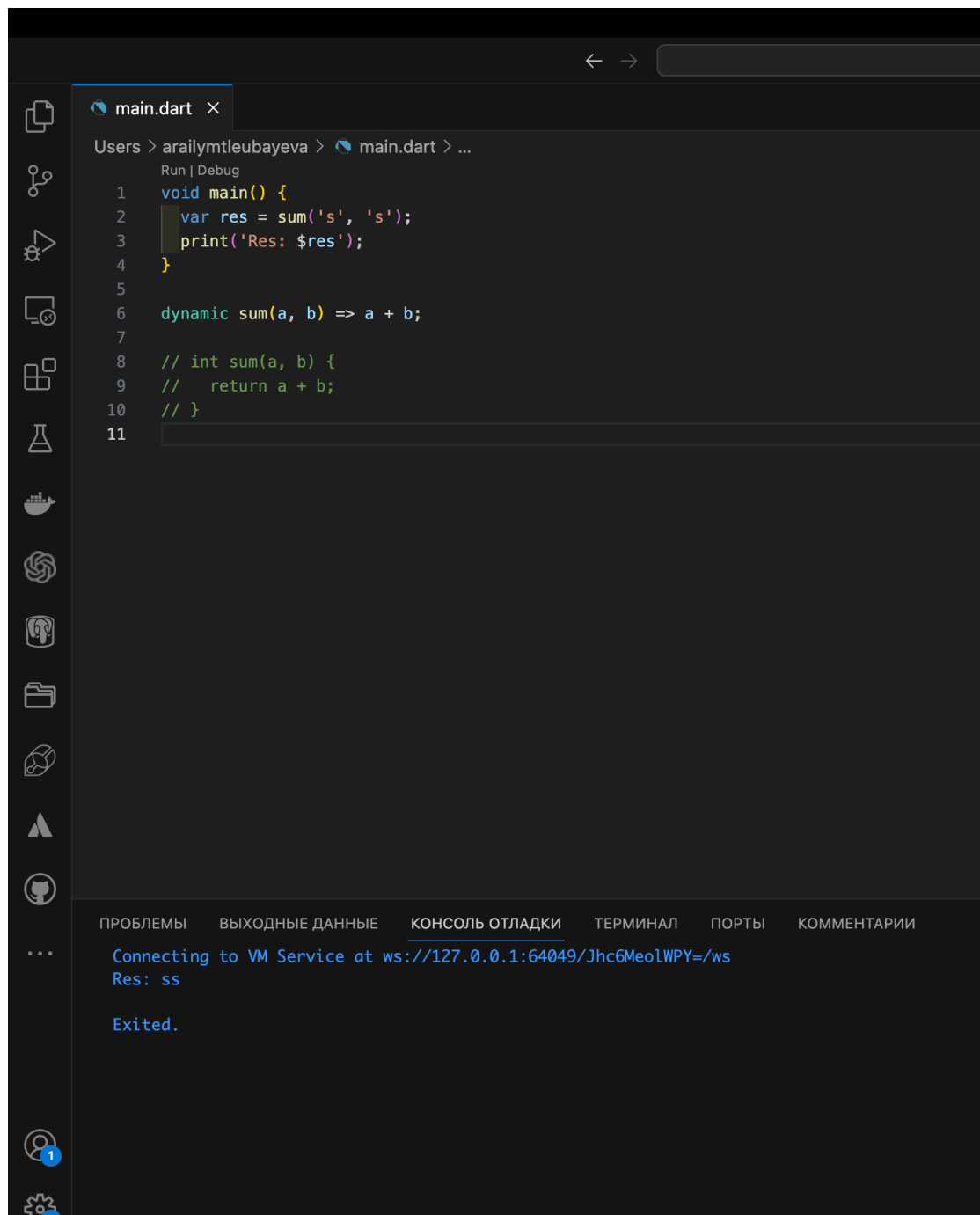


2) Несколько строк

Выполните вывод информации в нескольких строках.



3) Функции



4) Переменные

The screenshot shows an IDE window with a file named `main.dart`. The code defines a `main` function with various variable declarations and arithmetic operations. The console output shows the results of these operations, including a connection message to a VM Service.

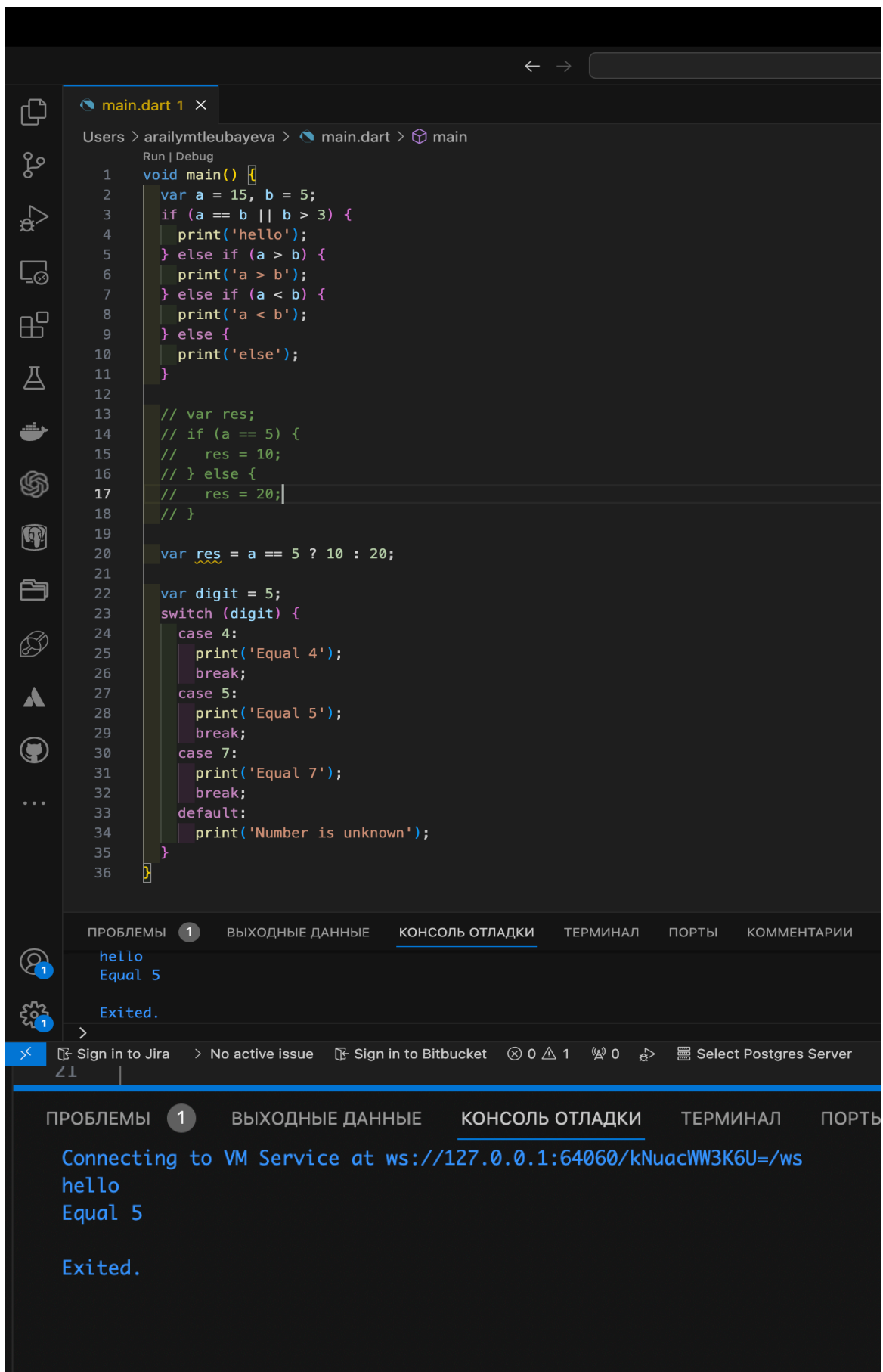
```
1  const num = 10;
2  var word = 'Hello';
3
4  void main() {
5    // print('Someone\'s\n\\');
6
7    var point = 5.5;
8    // point = point + 1;
9    // point++;
10
11    // int num = 0;
12    String some;
13    some = 'Hello';
14
15    bool isHappy;
16    isHappy = true;
17
18    print('Переменная: ${point + 0.5}. Еще что-то: ${some.toUpperCase()}');
19
20    dynamic dynamicData = 5;
21    dynamicData = 'word';
22    dynamicData = false;
23    print(dynamicData);
24
25    var a = 15, b = 5;
26    print('+ : ${a + b}');
27    print('- : ${a - b}');
28    print('* : ${a * b}');
29    print('/ : ${(a / b).toInt()}');
30    print('~/: ${a ~/ b}');
31    print('% : ${a % b}');
32
33    // Сокращения
34    a += 5; // Тоже что: a = a + 5;
35    a++; // То же что: a += 1;
36  }
```

Console Output:

```
Connecting to VM Service at ws://127.0.0.1:64056/kpMlyhBqc7Y=/ws
Переменная: 6.0. Еще что-то: HELLO
false
+ : 20
- : 10
* : 75
/ : 3
~/ : 3
% : 0

Exited.
```

5) Условные конструкции



6) Списки данных

main.dart 2 x

Users > arailymtleubayeva > main.dart > main

Run | Debug

```
1 void main() {
2   var nums = [5, 7, 8, 'hello', true]; // List()
3   nums.add(7);
4   nums.addAll([5, 7, 3, 2]);
5
6   nums.remove(8);
7   nums.removeAt(0);
8   // nums.removeWhere((element) => element >= 5);
9
10  // List<int> nums2;
11  // nums2 = [4, 6, 7];
12
13  var digits = {5, 6, 7, 2, 5, 3, 6}; // Set<int>
14  print(digits);
15
16  print('First: ${nums.first}. Last: ${nums.last}. Length: ${nums.length}');
17 }
```

ПРОБЛЕМЫ 2

ВЫХОДНЫЕ ДАННЫЕ

КОНСОЛЬ ОТЛАДКИ

ТЕРМИНАЛ

ПОРТЫ

КОММЕНТАРИИ

Connecting to VM Service at ws://127.0.0.1:64083/Ae0M910K8Kc=/ws
{5, 6, 7, 2, 3}
First: 7. Last: 2. Length: 8

Exited.

Sign in to Jira

No active issue

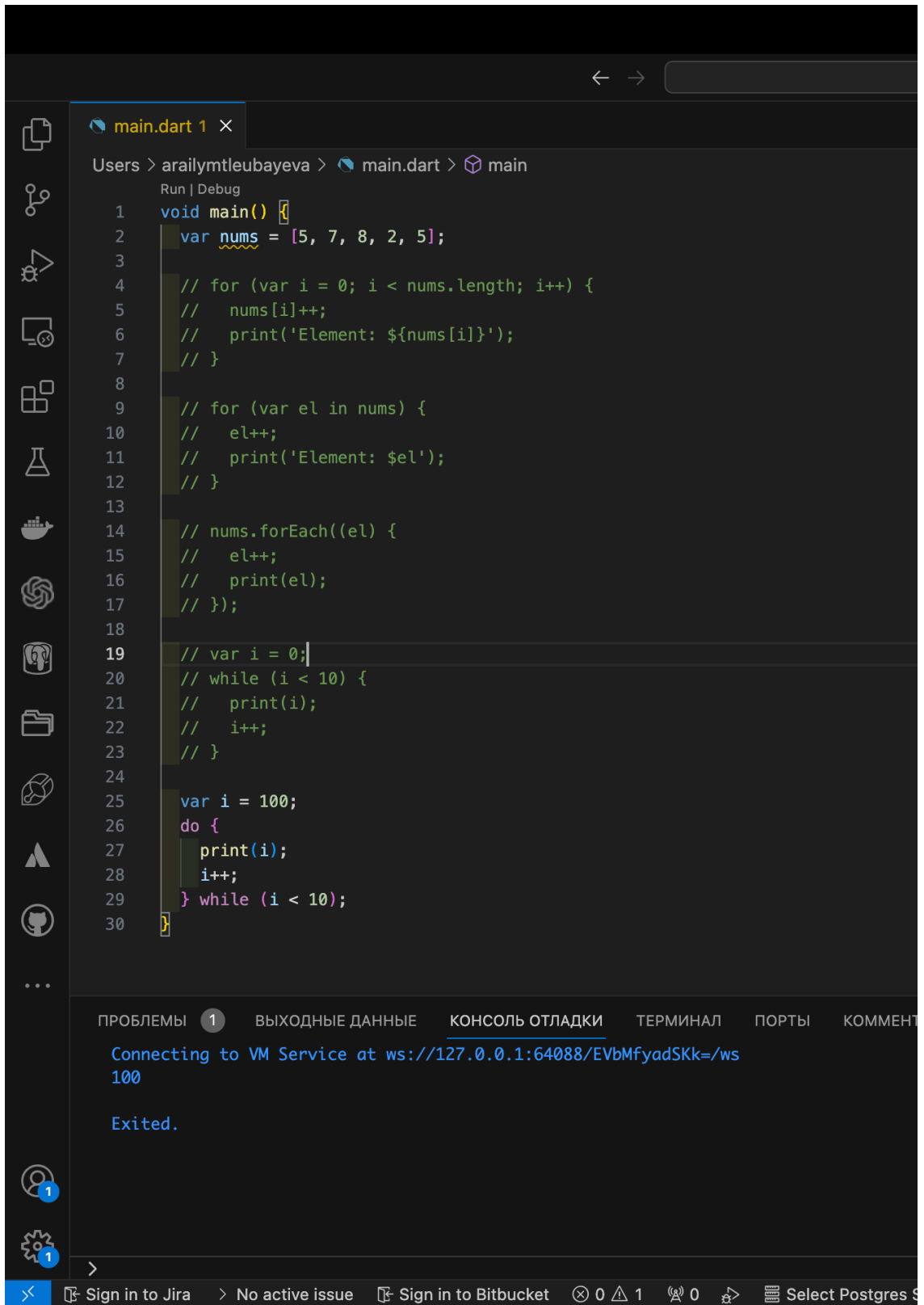
Sign in to Bitbucket

0 2

0

Select Postgres Server

7) Циклы



The screenshot shows an IDE window with a Dart file named `main.dart`. The code defines a `main` function containing several loop examples, all commented out. The examples include a `for` loop with an index, a `for` loop using `in` iteration, a `forEach` method call, and a `do-while` loop. The debug console at the bottom shows the output of the program: "Connecting to VM Service at ws://127.0.0.1:64088/EVbMfyadSKk=/ws" followed by "100" and "Exited.".

```
1 void main() {}
2   var nums = [5, 7, 8, 2, 5];
3
4   // for (var i = 0; i < nums.length; i++) {
5   //   nums[i]++;
6   //   print('Element: ${nums[i]}');
7   // }
8
9   // for (var el in nums) {
10  //   el++;
11  //   print('Element: $el');
12  // }
13
14  // nums.forEach((el) {
15  //   el++;
16  //   print(el);
17  // });
18
19  // var i = 0;
20  // while (i < 10) {
21  //   print(i);
22  //   i++;
23  // }
24
25  var i = 100;
26  do {
27    print(i);
28    i++;
29  } while (i < 10);
30 }
```

PROБЛЕМЫ 1 Выходные данные КОНСОЛЬ ОТЛАДКИ ТЕРМИНАЛ ПОРТЫ КОММЕНТЫ

Connecting to VM Service at ws://127.0.0.1:64088/EVbMfyadSKk=/ws
100
Exited.

Sign in to Jira No active issue Sign in to Bitbucket 0 1 0 Select Postgres S

Самостоятельное задание 1

1) Создание переменных

Создайте несколько переменных для хранения таких значений, как: -34, 4, 'R', 23.093433, true. Продумайте типы данных для переменных, чтобы они максимально подходили под каждое значение.

Важно: не используйте тип данных var для создания переменных!

2) Вывести четное число

Из двух чисел с разной четностью вывести на экран четное число. a, b - данные числа.

3) Создание калькулятора

Создайте две переменные со значением 5 и 19. Создайте еще одну переменную со значением одного арифметического действия: +, -, *, /.

В зависимости от символа, который будет во второй переменной, выполните математические действия над числами.

Для проверки данных используйте оператор switch case

4) Сумма чисел

Найдите сумму $1+2+3+\dots+n$, где число n записано в отдельной переменной.

5) Сумма чисел

Найдите сумму отрицательных элементов массива.

Массив:

[-24, 34, -4, 4, 5, -1]

6) Сравнение чисел

У вас есть два числа: 5 и 8.

Создайте функцию, что сравнивает числа и возвращает результат в виде знаков >, < или =.

Посмотреть ответ

Подготовьте отчет(Word файл) по выполнению примеров и самостоятельной работе с скриншотами исходного кода и результата компиляции.

На скриншоте пусть будет видно что это ваша работа (Создайте файл dart с вашим именем).

Исходные файлы с кодом загрузите в репозиторий GitHub и оставьте ссылку в отчете.