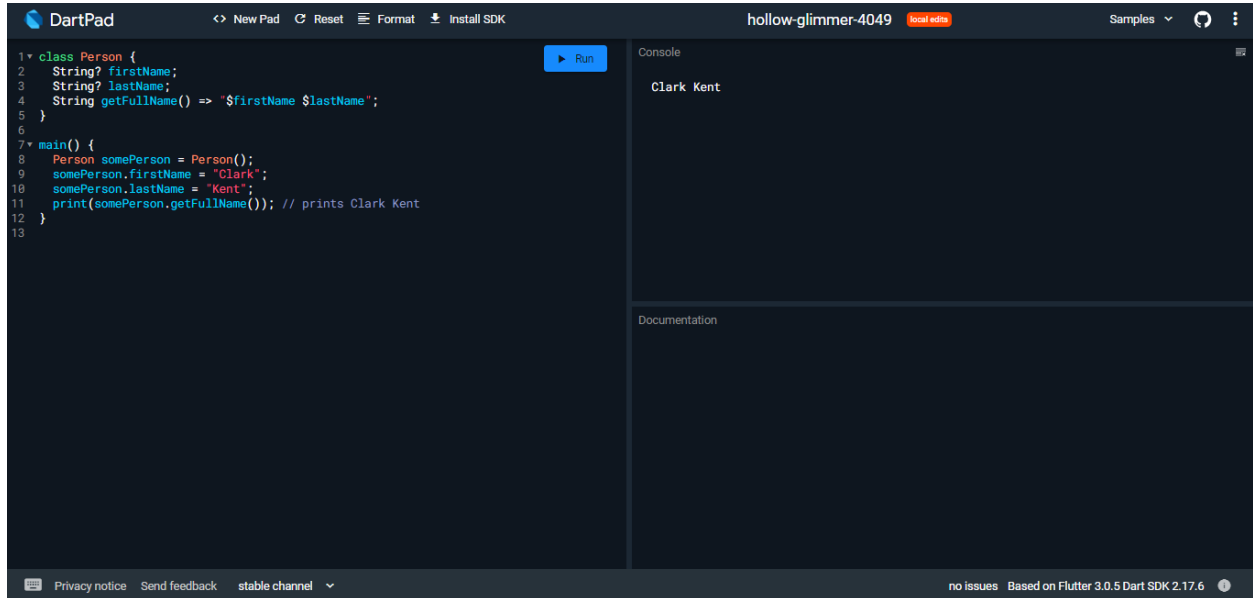


Nama : Dhika Ainul Luthfi

Kelas : TSA MOBILE

Struktur Kelas

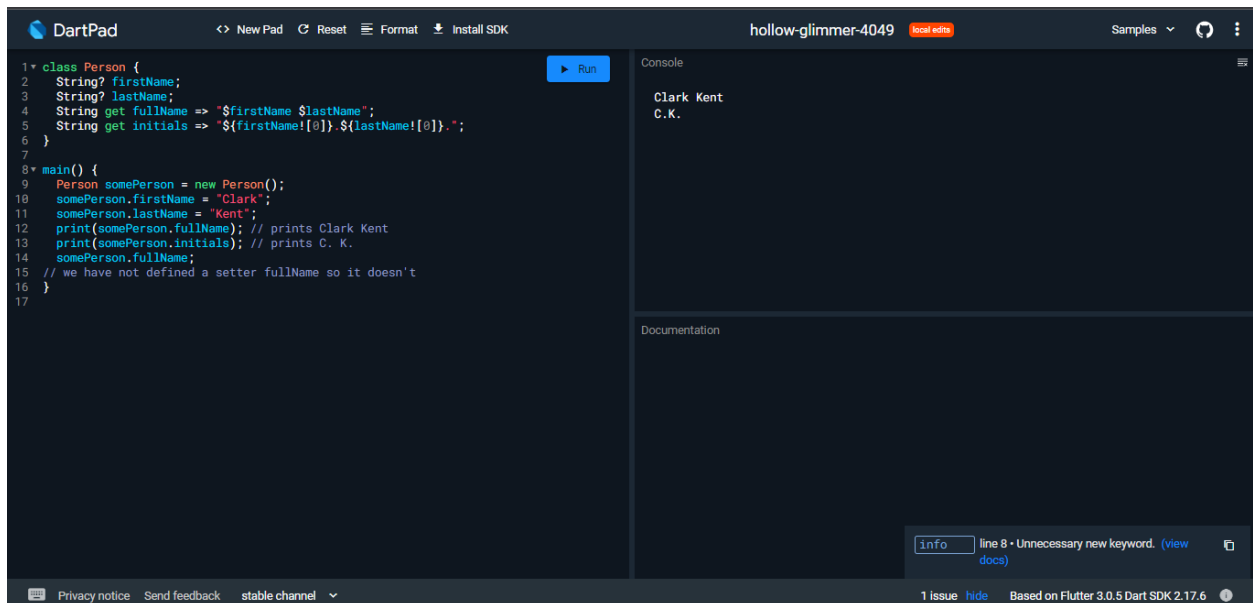


The screenshot shows the DartPad interface with a Dart class named `Person`. The class has two private fields, `firstName` and `lastName`, and a public method `getFullName` that concatenates them. The `main` function creates an instance of `Person` with the values "Clark" and "Kent", and prints the result of `getFullName`.

```
1 class Person {
2   String? firstName;
3   String? lastName;
4   String getFullName() => "$firstName $lastName";
5 }
6
7 main() {
8   Person somePerson = Person();
9   somePerson.firstName = "Clark";
10  somePerson.lastName = "Kent";
11  print(somePerson.getFullName()); // prints Clark Kent
12 }
13
```

The console output shows "Clark Kent".

Code Setter dan Getter



The screenshot shows the DartPad interface with a Dart class named `Person`. The class has two private fields, `firstName` and `lastName`, and two public methods, `fullName` and `initials`. The `main` function creates an instance of `Person` with the values "Clark" and "Kent", and prints the result of `fullName` and `initials`. There is a comment indicating that a setter for `fullName` has not been defined.

```
1 class Person {
2   String? firstName;
3   String? lastName;
4   String get fullName => "$firstName $lastName";
5   String get initials => "${firstName![0]} ${lastName![0]}. ";
6 }
7
8 main() {
9   Person somePerson = new Person();
10  somePerson.firstName = "Clark";
11  somePerson.lastName = "Kent";
12  print(somePerson.fullName); // prints Clark Kent
13  print(somePerson.initials); // prints C. K.
14  somePerson.fullName;
15  // we have not defined a setter fullName so it doesn't
16 }
17
```

The console output shows "Clark Kent" and "C.K.". An error message is visible at the bottom: "line 8 • Unnecessary new keyword. (view docs)".

Static Fields and Methods

The screenshot shows the DartPad interface with a Dart class named `Person`. The class has two static fields, `firstName` and `lastName`, both of type `String?`. It also has two static methods, `fullName` and `initials`, both returning `String`. The `main` function creates two `Person` objects, `somePerson` and `anotherPerson`, and prints their `fullName` and `initials` properties. The console output shows the results of these operations.

```
1 class Person {
2   String? firstName;
3   String? lastName;
4   String get fullName => "$firstName $lastName";
5   String get initials => "${firstName![0]} ${lastName![0]}.";
6 }
7
8 main() {
9   Person somePerson = Person();
10  somePerson.firstName = "Clark";
11  somePerson.lastName = "Kent";
12  Person anotherPerson = Person();
13  anotherPerson.firstName = "Peter";
14  anotherPerson.lastName = "Parker";
15  print(somePerson.fullName);
16  // prints Person name: Clark kent
17
18  print(anotherPerson.fullName); // prints Person name:Peter Parker
19  // Person.personLabel = "name:";
20  print(somePerson.fullName); // prints name: Clark Kent
21  print(anotherPerson.fullName); // prints name: Peter Parker
22 }
23
```

Console output:

```
Clark Kent
Peter Parker
Clark Kent
Peter Parker
```

Documentation

no issues Based on Flutter 3.0.5 Dart SDK 2.17.6

Konstruktor

The screenshot shows the DartPad interface with a Dart class named `Person`. The class has two static fields, `firstName` and `lastName`, both of type `late String`. It also has a static method `fullName` returning `String`. The `main` function creates a `Person` object, `somePerson`, and prints its `fullName` property. The console output shows the result of this operation.

```
1 class Person {
2   late String firstName;
3   late String lastName;
4   Person(String firstName, String lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7   }
8   String get fullName() => "$firstName $lastName";
9 }
10
11 void main() {
12  // Person somePerson = Person(); No longer compiles
13  Person somePerson = Person("Clark", "Kent");
14  print(somePerson.fullName());
15 }
16
```


Console output:

```
Clark Kent
```

Documentation

2 issues hide Based on Flutter 3.0.5 Dart SDK 2.17.6

Factory Constructor

 DartPad

<> New Pad

↺ Reset


≡ Format

⬇️ Install SDK

▶ Run

```
1 class Person {
2   late String firstName;
3   late String lastName;
4   Person(String firstName, String lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7   }
8   String getFullName() => "$firstName $lastName";
9   factory Person.fromCache(String firstName, String lastName) {
10    if (_cacheService.containsPerson(firstName, lastName)) {
11      return _cacheService.getPerson(firstName, lastName);
12    } else {
13      return Person(firstName, lastName);
14    }
15  }
16 }
17
18 void main() {
19   // Person somePerson = Person(); No longer compiles
20   Person somePerson = Person("Clark", "Kent");
21   print(somePerson.getFullName());
22 }
23
```

Class Inheritance

 DartPad

<> New Pad

↺ Reset

≡ Format

⬇️ Install SDK

hollow-glimmer-4049 local edit

Samples

🔍

⋮

▶ Run

```
1 class Person {
2   late String firstName;
3   late String lastName;
4   Person(String firstName, String lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7   }
8   String getFullName() => "$firstName $lastName";
9   String get fullName => "$firstName $lastName";
10 }
11 class Student extends Person {
12   late String nickname;
13   Student(
14     String firstName,
15     String lastName,
16     this.nickname,
17   ) : super(firstName, lastName);
18 }
19 @override
20 String toString() => "$fullName, aka $nickname";
21 }
22 main() {
23   Student s = Student("Clark", "Kent", "Kal-El");
24   print(s);
25 }
26
```

Console

Clark Kent, aka Kal-El

Documentation

info

 line 5 • Use initializing formals when possible. (view docs)

info

 line 6 • Use initializing formals when possible. (view docs)

Privacy notice

Send feedback

stable channel

2 issues [hide](#)

Based on Flutter 3.0.5 Dart SDK 2.17.6

Interfaces

DartPad interface showing a Dart class hierarchy. The code defines a `Person` class with `firstName` and `lastName` properties, and a `Student` class that implements `Person` and adds a `nickName` property. The `main` function creates a `Student` instance and prints its `fullName`.

```
1 class Person {
2   late String firstName;
3   late String lastName;
4   Person(String firstName, String lastName) {
5     this.firstName = firstName;
6     this.lastName = lastName;
7   }
8   // String getFullName() => "$firstName $lastName";
9   String get fullName => "$firstName $lastName";
10 }
11 class Student implements Person {
12   String nickName;
13   @override
14   String firstName;
15   @override
16   String lastName;
17   Student(this.firstName, this.lastName, this.nickName);
18   @override
19   String get fullName => "$firstName $lastName";
20   @override
21   String toString() => "$fullName, also known as $nickName";
22 }
23
24 main() {
25   Student s = Student("Clark", "Kent", "Kal-El");
26   print(s);
27 }
28
```

Console output: Clark Kent, also known as Kal-El

Documentation: dynamic \$nickName

Issues: 2 issues (hide) Based on Flutter 3.0.5 Dart SDK 2.17.6

Abstract

DartPad interface showing a Dart class hierarchy. The code defines an abstract `Person` class with `firstName` and `lastName` properties, and a `Student` class that extends `Person` and adds a `nickName` property. The `main` function creates a `Student` instance and prints its `fullName`.

```
1 abstract class Person {
2   String firstName;
3   String lastName;
4   Person(this.firstName, this.lastName);
5   String get fullName;
6 }
7
8 class Student extends Person {
9   late String nickName;
10   Student(
11     String firstName,
12     String lastName,
13     this.nickName,
14   ) : super(firstName, lastName);
15
16   @override
17   String get fullName => "$firstName, $lastName, $nickName";
18 }
19
20 main() {
21   Person student = new Student("Clark", "Kent", "Kal-El");
22   print(student.fullName);
23 }
24
```

Console output: Clark, Kent, Kal-El

Documentation: line 19 - Unnecessary new keyword. (view docs)

Issues: 1 issue (hide) Based on Flutter 3.0.5 Dart SDK 2.17.6

Mixins

DartPad

<> New Pad ⌵ Reset ⌵ Format ⌵ Install SDK

hollow-glimmer-4049 local edit

Samples ↕ 🔍 ⋮

▶ Run

```
1 class Person {
2   String firstName;
3   String lastName;
4   Person(this.firstName, this.lastName);
5   String get fullName => "$firstName $lastName";
6 }
7
8 class ProgrammingSkills {
9   coding() {
10    print("writing code...");
11  }
12 }
13
14 class ManagementSkills {
15   manage() {
16    print("managing project...");
17  }
18 }
19
20 class SeniorDeveloper extends Person with ProgrammingSkills, ManagementSkills {
21   SeniorDeveloper(String firstName, String lastName)
22     : super(firstName, lastName);
23 }
24
25 class JuniorDeveloper extends Person with ProgrammingSkills {
26   JuniorDeveloper(String firstName, String lastName)
27     : super(firstName, lastName);
28 }
29
30 main() {
31   SeniorDeveloper sd = SeniorDeveloper("Clark", "Kent");
32   print(sd.fullName);
33   JuniorDeveloper jd = JuniorDeveloper("Peter", "Parker");
34   print(jd.fullName);
35 }
```

Console

Clark Kent
Peter Parker

Documentation

Privacy notice Send feedback stable channel ⌵

no issues Based on Flutter 3.0.5 Dart SDK 2.17.6 ⓘ