Dart Básico - Aula 4

Fernando Duwe

v 1.0

Uma variável tem o valor null quando nenhum valor foi atribuído à ela, ou seja, é literalmente nada. Por exemplo:

```
String nome; // variável declarada, mas ainda não populada com valores
```

O que acontecerá se logo após eu declarar a variável acima, eu executar o código abaixo?

```
print(nome + " é o nome da pessoa");
```

Ver mais

O Null é a indicação que nenhum valor ainda foi atribuído à variável, ou seja, que a variável ainda não possui valor.

Porém, a utilização de variáveis ainda não inicializadas, ou com valores nulos, é o principal causador de bugs na indústria de software, isso porque é muito difícil detectar essas possíveis falhas durante a fase de desenvolvimento.

Normalmente, podemos dividir os erros de programação em duas categorias distintas:

- Erros na escrita do código (Sintaxe e Semântica);
- Erros de lógica;

Ao compilar o meu código (em linguagens compiladas) ou ao executar (em linguagens interpretadas) os erros na escrita do código são rapidamente detectados, bastando ao desenvolvedor corrigi-los.

Dentro da programação, sintaxe e semântica são termos relacionados a qualquer linguagem:

- Erro de sintaxe: Um erro na montagem do código, onde um código não compatível com a especificação da linguagem é escrito;
- Erro semântico: O código escrito é válido, mas não corresponde ao que o desenvolvedor deseja de fato fazer;

Ambos os erros acima são facilmente detectados e de fácil correção.

Os erros de programação, são aqueles erros onde o código foi escrito conforme a linguagem específica, mas erros podem ocorrer em determinadas situações, onde o programador não pensou em determinadas situações. Um exemplo é o exercício de idade que criamos, quando o usuário informava uma idade negativa, quebrando a lógica proposta.

Os erros de acesso a variáveis nulas se encaixa nesse tipo de erro, quando vou realizar uma operação em uma variável que ainda não teve seu valor determinado, seja por erro de programação ou por algum outro problema.

Null vs Undefined

Ver mais

Durante as aulas surgiu um debate sobre a diferença entre null e undefined.

- Null: Uma variável não tem valor definido, e se tem, ainda não temos como saber. É uma indicação de que não temos certeza qual é o seu valor correspondente.

  Utilizado quando uma variável não teve valor atribuído;

- Undefined: Indica que ou a variável ainda não existe ou se existe, não tem valor definido;

Vale lembrar que o conceito de undefined não é presente no Dart, mas sim, vindo do Javascript.

Null safety

Ver mais

Uma linguagem chamada de null safety é uma linguagem que possui mecanismos para prevenir problemas relacionados a variáveis e atributos que sejam utilizados quando ainda não possuem valor definido.

Desde a versão 2.12, o Dart possui a compatibilidade com Null Safety, podendo essa questão ser configurada. A partir da versão 3 do Dart, o Null Safety é padrão na linguagem.

Toda variável declarada, por padrão, não poderá ser nula

```
int idade;
```

Caso eu ainda não saiba o valor de uma variável, posso utilizar o operador ? para indicar que ela pode ser nula

```
int? idade;

if (idade != null)
  if (idade >= 18)
    print("Maior de idade");

print(idade ?? 1899);
```

Exemplo em funções

```
void saudarUsuario(String nomeSistema, String? nomeUsuario)
{
  if (nomeUsuario == null) {
    print("Olá, usuário anônimo");
  } else {
    print("Bem vindo, $nomeUsuario");
  }

  print("Você está logado atualmente em: $nomeSistema");
}

String? getUsuarioAtual() {
  return null;
}
```

Operador !

O operador !, ao lado de uma variável que pode ser nula especifica que essa variável não é nula no momento

```
void main() {
  int? idade;

  if (1 == 2)
    idade = 18;

  if (1 == 2)
    print("Minha idade é ${idade! + 10}");
}
```

Parâmetros opcionais em funções

Adicionando [] entre os parâmetros, indicamos que esses parâmetros são opcionais

```
int incrementarValor(int valor, [int? incremento]) {
  if (incremento == null)
    return valor + 1;

  return valor + incremento;
}
```

Parâmetros opcionais em funções

Podemos também adicionar um valor padrão ao parâmetro opcional

```
int incrementarValor(int valor, [int incremento = 1]) {
  return valor + incremento;
}
```

Parâmetros nomeados

Parâmetros nomeados são uma forma interessante de passarmos informação para a nossa função

```
double sacar({String? numeroConta, double? valor}) {
  return 0;
}

void main() {
  sacar(
    valor: 544.90,
    numeroConta: "123-24-1241"
  );
}
```

Parâmetros nomeados

Parâmetros nomeados são sempre opcionais, a não ser que você especifique nos parâmetros que ele é obrigatório, utilizando a palavra reservada required.

```
String? pesquisarUsuario({required String textoPesquisa) {

  if (textoPesquisa == "")
    return null;

}
```

O modificador late

Possui duas funções principais:

- Utilizado para indicar que uma variáveis não nula receberá valor mais tarde
- Utilizado para instanciar as variáveis no método lazily

```
void main() {
  late String descricao;
}
```

Os modificadores final e const

Indicam que a variável não poderá ter seu valor alterado.

- const: A variável será fixada em tempo de compilação
- final: A variável poderá ter seu valor setado somente uma vez

```
void main() {
  const double valorPi = 3.14;

  final String userId = getUserId();
}
```

Ver mais

Utilizado para representar um conjunto fixo de valores em um grupo

```
enum Operacao { soma, subtracao, divisao, multiplicacao }

void main() {
  Operacao operacaoAtual = Operacao.soma;

  if (operacaoAtual == Operacao.multiplicacao)
    print("Multiplicação");
}
```

Enums - cont

Ver mais

```
enum Operacao { soma, subtracao, divisao, multiplicacao }

double calcular(double valor1, double valor2, Operacao operacao) {
  double resultado = 0;

  switch(operacao) {
    case Operacao.soma : resultado = valor1 + valor2;
                         break;

    case Operacao.subtracao : resultado = valor1 - valor2;
                              break;

    case Operacao.divisao : resultado = valor1 / valor2;
                            break;

    case Operacao.multiplicacao : resultado = valor1 * valor2;
                                  break;
  }

  return resultado;
}

void main() {
  print(calcular(2, 2, Operacao.soma));
}
```

Exercício 19

Crie uma rotina que calcule o IMC de uma pessoa. A rotina deve possuir dois parâmetros:

- Peso (obrigatório)
- Altura (opcional): Caso não preenchido, considerar a altura de 170cm

O retorno da rotina será um enum, com as seguintes opções:

AbaixoPeso, PesoSaudavel, SobrePeso, ObesidadeGrau1, ObesidadeGrau2, ObesidadeGrau3

Fórmula IMC:

IMC = peso / (altura * altura)

Obs:

peso (em kg)
altura (em m)

CLASSIFICAÇÃO	IMC
Abaixo do Peso	Abaixo 18,5
Peso Normal	18,5 - 24,9
Sobrepeso	25 - 29,9
Obesidade Grau I	30 - 34,9
Obesidade Grau II	35 - 39,9
Obesidade Grau III ou Mórbida	Maior ou Igual 40

Exercício 20

Crie uma rotina que calcule o valor de uma corrida de taxi. O cálculo do valor deve levar em consideração os seguintes itens:

- O valor mínimo da corrida é R$ 3,50
- O valor cobrado por quilometro percorrido é de R$ 1,25
- A partir da bandeira 2, é adicionado um valor de R$ 0,35 a cada bandeira adicional
  Ex: Bandeira 2: R$ 0,35, Bandeira 3 R$ 0,70, Bandeira 4 R$ 1,05...
- O passageiro poderá ter um convênio, caso possuir, recebe um desconto de 10% em sua corrida
- Caso a distância percorrida for igual a 0 (zero), gerar uma exceção, pois a corrida é inválida

Sobre os dados que a rotina espera:

- A bandeira é opcional, se não informada, pode ser considerada a bandeira 1
- A quilometragem percorrida é obrigatória
- A informação de convênio é igual a Sim ou Não e seu preenchimento é opcional. Caso não preenchido, considerar Não

Essa rotina devolverá o valor (em R$) a ser cobrado do passageiro

Exercício 21

Escreva uma rotina para contar, em uma string, quantas vezes cada vogal (A, E, I, O, U) é repetida.

Exercício 22

Escreva uma rotina que verifique se uma determinada palavra é palíndroma. Uma palavra ou expressão palíndroma é quando essa palavra ou expressão poderá ser lida igualmente tanto da esquerda para a direita, quanto da direita para a esquerda.

Exemplo de frases palíndromas:

- Otimo, so eu, que os omito
- Socorram-me, subi no onibus em Marrocos
- O romano acata amores a damas amadas e Roma ataca o namoro