
Comentando provas
FUNDAÇÃO
cesgranrio
BB AGENTE DE TECNOLOGIA
ROGERAOARAUJO
www.rogeraoaraujo.com.br
ROGERAOARAUJO
www.rogeraoaraujo.com.br

1


Java
ROGERAOARAUJO
www.rogeraoaraujo.com.br
2

2

Blocos de inicialização estáticos

São blocos de código

Precedidos pela palavra-chave static

São executados apenas uma vez

No momento em que a classe é carregada na memória pela JVM

Características

São executados apenas uma vez

Não podem acessar diretamente membros de instância

Quando a classe é carregada

Porque ainda não há nenhum objeto criado no momento de sua execução

Características

São usados para inicializar atributos estáticos

São usados também para realizar operações que precisam ser executadas apenas uma vez

Como

Configuração de uma biblioteca

Conexão com um banco de dados

ROGERAOARAUJO
www.rogeraoaraujo.com.br

5

Sintaxe

```
static {
    // Código a ser executado quando a classe for carregada
}
```

ROGERAOARAUJO
www.rogeraoaraujo.com.br

6

Exemplo

- Código:
  - public class Configuracao {
    - private **static** String configuracao;
    - // Bloco estático
    - **static** {
      - configuracao = "Configuração inicializada";
      - System.out.println("Bloco estático executado");
    - }
    - ...

Exemplo (continuação)

- Código (continuação) :
    - ...
    - public **static** String getConfiguracao() {
      - return configuracao;
    - }
    - public **static** void main(String[] args) {
      - System.out.println(Configuracao.getConfiguracao());
    - }
  - }

Exemplo (continuação)

- Resultado da execução:
  - Bloco estático executado
  - Configuração inicializada

Blocos de inicialização de instância

São blocos de código

Não precedidos pela palavra-chave static

São executados sempre que um objeto é criado

Antes do construtor da classe

Características

São executados	São usados para		Podem acessar	
Toda vez que	Inicializar	Realizar	Atributos	
Um objeto é criado	Atributos de instância	Operações que devem ser feitas para cada objeto	Estáticos	De instância

ROGERAOARAUJO www.rogeraoaraujo.com.br

11

11

Sintaxe

```
{
```

- // Código a ser executado quando um objeto for criado

```
}
```

ROGERAOARAUJO www.rogeraoaraujo.com.br

12

12

Exemplo

- Código:
  - public class Pessoa {
    - private String nome;
    - // Bloco de instância
    - {
      - System.out.println("Bloco não estático executado");
      - nome = "Nome padrão";
    - }
    - public Pessoa(String nome) {
      - this.nome = nome;
    - }
    - public String getNome() {
      - return nome;
    - }
    - ...

Exemplo (continuação)

- Código (continuação) :
  - ...
  - public static void main(String[] args) {
    - Pessoa joao = new Pessoa("João");
    - System.out.println("Nome: " + joao.getNome());
    - Pessoa maria = new Pessoa("Maria");
    - System.out.println("Nome: " + maria.getNome());
  - }
- }

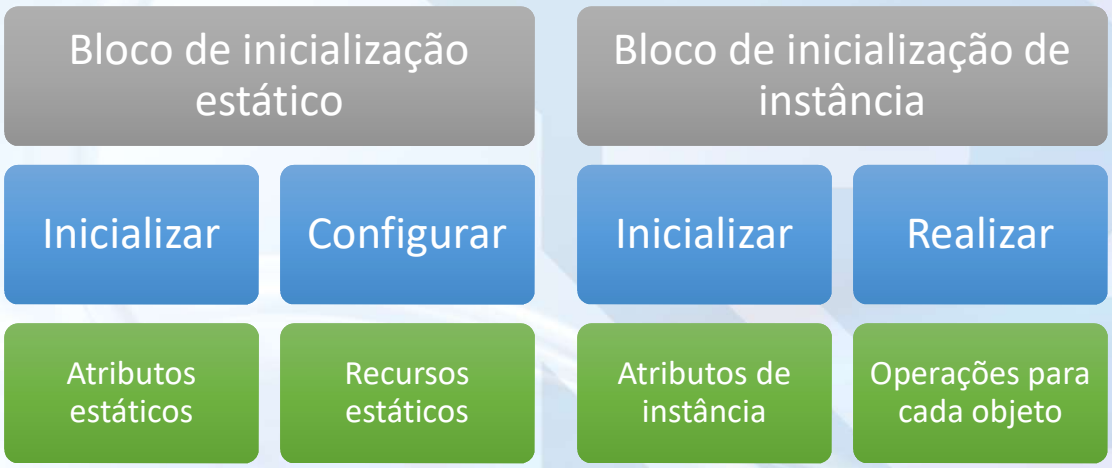
Exemplo (continuação)

- Resultado da execução:
  - Bloco não estático executado
  - Nome: João
  - Bloco não estático executado
  - Nome: Maria

Diferenças dos blocos

Aspecto	Bloco de inicialização estático	Bloco de inicialização de instância
Execução	Uma vez quando a classe é carregada	Toda vez que um objeto é criado
Palavra-chave static	Usa	Não usa
Acesso a atributos	Pode acessar apenas atributos estáticos	Pode acessar atributos estáticos e de instância

Uso principal dos blocos



ROGERAOARAUJO
www.rogeraoaraujo.com.br

17

17

Exemplo

- Casse Exemplo:
  - public class Exemplo {
    - private static int contadorEstatico;
    - private int contadorInstancia;
    - // Bloco estático
    - **static {**
      - **contadorEstatico = 0;**
      - **System.out.println("Bloco estático executado: " + contadorEstatico);**
    - **}**
    - // Bloco não estático
    - **{**
      - **contadorInstancia = ++contadorEstatico;**
      - **System.out.println("Bloco de instância executado: " + contadorInstancia);**
    - **}**
  - }

ROGERAOARAUJO
www.rogeraoaraujo.com.br

18

18

Exemplo (continuação)

- Código:
  - public class Main {
    - public static void main(String[] args) {
      - Exemplo obj1 = new Exemplo();
      - Exemplo obj2 = new Exemplo();
    - }
  - }

- Resultado da execução:
  - Bloco estático executado: 0
  - Bloco de instância executado: 1
  - Bloco de instância executado: 2

Ordem de inicialização de informações de um objeto

1. Atributos e blocos de inicialização estáticos
   - Na ordem em que aparecem
   - É feito apenas uma vez quando a classe é carregada
2. Atributos e blocos de inicialização de instância
   - Na ordem em que aparecem
   - É feito cada vez que a classe é instanciada
3. Construtores
   - É feito cada vez que a classe é instanciada

Atributos e blocos de inicialização estáticos

São inicializados uma única vez

Quando a classe é carregada pela JVM

A ordem é

Atributos estáticos são inicializados em ordem de declaração

Blocos estáticos são executados na ordem em que aparecem na classe

ROGERAOARAUJO
www.rogeraoaraujo.com.br

21

Atributos e blocos de inicialização de instância

São inicializados

Cada vez que um objeto é criado

A ordem é

Atributos de instância são inicializados em ordem de declaração

Blocos não estáticos são executados na ordem em que aparecem na classe

ROGERAOARAUJO
www.rogeraoaraujo.com.br

22

Construtor

É executado

- Para finalizar a inicialização
- Após as variáveis estáticas e de instância
- Do objeto
- Serem inicializadas

ROGERAOARAUJO www.rogeraoaraujo.com.br

23

Construtor

Inicialização a cada instância

- O construtor é chamado para cada nova instância da classe
- Após a inicialização das variáveis de instância

Personalização da inicialização

- Os construtores podem ser sobrecarregados para oferecer diferentes maneiras de inicializar um objeto
- Possibilitando a passagem de parâmetros para configurar o estado inicial do objeto

ROGERAOARAUJO www.rogeraoaraujo.com.br

24

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] As classes Java a seguir são públicas e ocupam arquivos separados.

```
public class Tst {

	int ini=0,fim=25;

	void print() {
		System.out.println(ini+fim);
	}

	{
		ini=fim%7;
		fim=ini*3;
	}

	Tst(int a, int b) {
		ini+=a;
		fim+=b;
	}

	{
		ini/=2;
		fim+=10;
	}
}

public class Main {

	public static void main(String[] args) {
		new Tst(4, -4).print();

	}
}
```


Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido no console quando o método main for executado?

- [A] 0
- [B] 10
- [C] 24
- [D] 25
- [E] 33

Comentários

```
public class Tst {

	int ini=0,fim=25;

	void print() {
		System.out.println(ini+fim);
	}

	{
		ini=fim%7;
		fim=ini*3;
	}

	Tst(int a, int b) {
		ini+=a;
		fim+=b;
	}

	{
		ini/=2;
		fim+=10;
	}
}

public class Main {

	public static void main(String[] args) {
		new Tst(4, -4).print();

	}

}
```

	ini	fim
Carregamento da classe	-	-
Criação do objeto	0	25
Primeiro bloco de inicialização de instância	4	12
Segundo bloco de inicialização de instância	2	22
Construtor Tst(4, -4)	6	18
Método Tst(4, -4).print()	**24**	

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido no console quando o método main for executado?

- [A] 0
- [B] 10
- **[C] 24**
- [D] 25
- [E] 33

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido no console quando o método main for executado?

- [A] 0
- [B] 10
- **[C] 24**
- [D] 25
- [E] 33

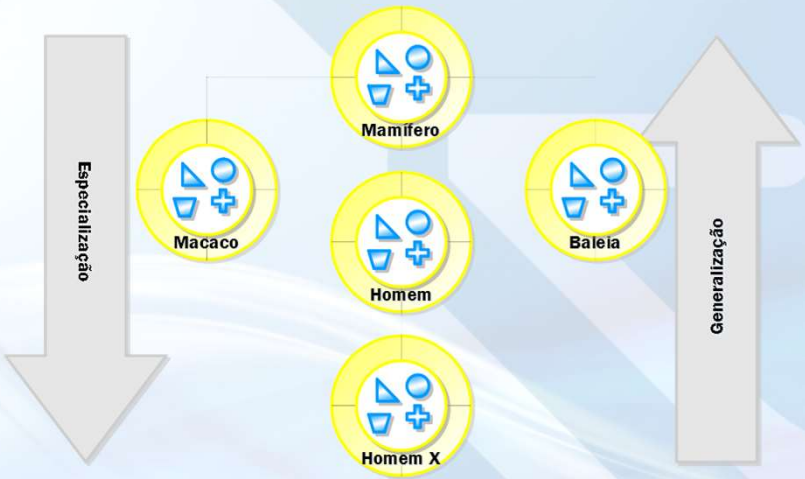
ROGERAOARAUJO
www.rogeraoaraujo.com.br

29

29

30

Esquema



Classe Mamifero

- public **abstract** class Mamifero {
  - public **abstract** String **locomoverSe**();
- }

Classes Macaco e Baleia

- public class Macaco extends Mamifero {
  - @Override
  - public String **locomoverSe**() {
    - return "Pulando de galho em galho.";
  - }
- }

- public class Baleia extends Mamifero {
  - @Override
  - public String **locomoverSe**() {
    - return "Nadando.";
  - }
- }

ROGERAOARAUJO
www.rogeraoaraujo.com.br
33

Classes Homem e HomemX

- public class Homem extends Mamifero {
  - @Override
  - public String **locomoverSe**() {
    - return "Andando.";
  - }
- }

- public class HomemX extends Homem {
  - @Override
  - public String **locomoverSe**() {
    - return "Voando.";
  - }
- }

ROGERAOARAUJO
www.rogeraoaraujo.com.br
34

Exemplo geral

```
public class Main {
    public static void main(String[] args) {
        Mamifero mamifero;
        System.out.println("Macaco **************************");
        mamifero = new Macaco(); // Macaco como mamífero
        System.out.println("Locomoção: " + mamifero.locomoverSe());
        System.out.println("Baleia **************************");
        mamifero = new Baleia(); // Baleia como mamífero
        System.out.println("Locomoção: " + mamifero.locomoverSe());
        ...
```


Exemplo geral (continuação)

```
        ...
        System.out.println("Homem ************************");
        mamifero = new Homem(); // Homem como mamífero
        System.out.println("Locomoção: " + mamifero.locomoverSe());
        System.out.println("Homem X ************************");
        mamifero = new HomemX(); // Homem X como mamífero
        System.out.println("Locomoção: " + mamifero.locomoverSe());
    }
}
```

Resultado da execução

- Macaco **********
- Locomoção: Pulando de galho em galho.
- Baleia **********
- Locomoção: Nadando.
- Homem **********
- Locomoção: Andando.
- Homem X **********
- Locomoção: Voando.

Polimorfismo

É a propriedade de duas ou mais classes derivadas de uma mesma superclasse responderem a mesma mensagem

Ocorre quando uma subclasse redefine um método existente na superclasse

Cada uma de uma forma diferente

Métodos sobrescritos (overriding)

Métodos estáticos

Possuem o modificador static

- São invocados a partir do nome da classe
- Podem ser referidos a partir das instâncias da classe
- Seu uso comum
- Sem a necessidade de criar uma instância dela
- Mas não é recomendado porque questão de clareza
- Acessar atributos estáticos

ROGERAOARAUJO www.rogeraoaraujo.com.br

39

Características

São chamados usando a partir do nome da classe

- Podem acessar diretamente
  - Atributos estáticos
  - Métodos estáticos
- Não podem acessar diretamente
  - Atributos de instância
  - Métodos de instância

ROGERAOARAUJO www.rogeraoaraujo.com.br

40

Exemplo

- Código:
  - class Calculadora {
    - public static int somar(int a, int b) {
      - return a + b;
    - }
    - public static int subtrair(int a, int b) {
      - return a - b;
    - }
  - }

Exemplo (continuação)

- Código (continuação):
  - public class Main {
    - public static void main(String[] args) {
      - System.out.println("Resultado: " + Calculadora.somar(5, 10));
      - System.out.println("Resultado: " + Calculadora.subtrair(5, 10));
    - }
  - }
- Resultado da execução:
  - Resultado: 15
  - Resultado: -5

Métodos ocultados

São métodos estáticos da subclasse

Que possuem a mesma assinatura

Dos métodos estáticos da superclasse

Exemplo

- Código:
  - public class Animal {
    - public **static** void nomeClasse() {
      - System.out.println("Animal");
    - }
  - }

- Código (continuação):
  - public class Cachorro **extends** Animal {
    - public **static** void nomeClasse() {
      - System.out.println("Cachorro");
    - }
  - }

Exemplo (continuação)

- Código:
  - public class Main {
    - public static void main(String[] args) {
      - Animal.nomeClasse();
      - Cachorro.nomeClasse();
    - }
  - }

- Resultado da execução:
  - Animal
  - Cachorro

Métodos sobrescritos x ocultados

Versão do método sobrescrito	Versão do método oculto	
Quando invocado	Depende se é invocado	
É aquela da subclasse	Da superclasse	Da subclasse

Exemplo

- Código:
  - public class Animal {
    - public **static** void nomeClasse() {
      - System.out.println("Animal");
    - }
    - public void emitirSom() {
      - System.out.println("Som do animal");
    - }
  - }

- Código (continuação):
  - public class Cachorro **extends** Animal {
    - public **static** void nomeClasse() {
      - System.out.println("Cachorro");
    - }
    - @Override
    - public void emitirSom() {
      - System.out.println("Latindo...");
    - }
  - }

Exemplo (continuação)

- Código:
  - public class Main {
    - public static void main(String[] args) {
      - Animal.nomeClasse();
      - Cachorro.nomeClasse();
      - Animal animal = new Animal();
      - animal.emitirSom();
      - animal = new Cachorro();
      - animal.emitirSom();
      - animal.nomeClasse();
      - ...

- Código (continuação):
      - ...
      - Cachorro cachorro = new Cachorro();
      - cachorro.emitirSom();
      - cachorro.nomeClasse();
    - }
  - }

Exemplo (continuação)

- Resultado da execução:
  - Animal
  - Cachorro
  - Som do animal
  - Latindo...
  - Animal
  - Latindo...
  - Cachorro

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] Sejam as seguintes classes Java:

```
public class Va1 {

	public static String getStr() {
		return "abcdefghijklmnop";
	}

	public String ini(String s, int cpr) {
		return s.substring(0, cpr);
	}

	public String fin(String s, int cpr) {
		return ini(s, cpr)+s.substring(s.length()-cpr, s.length());
	}

}
```

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] Sejam as seguintes classes Java:

```
public class Va2 extends Va1 {

	public static String getStr() {
		return "0123456789ABCDEF";
	}

	public String ini(String s, int cpr) {
		return s.substring(s.length()-cpr, s.length());
	}

	public static void main(String[] args) {
		Va1 o=new Va2();

		System.out.println(o.fin(o.getStr(), 5));
	}

}
```


Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido no console quando o método main for executado?

- [A] 0123BCDE
- [B] BCDEFBCDEF
- [C] 01234BCDEF
- [D] abcdelmnop
- [E] lmnoplmnop

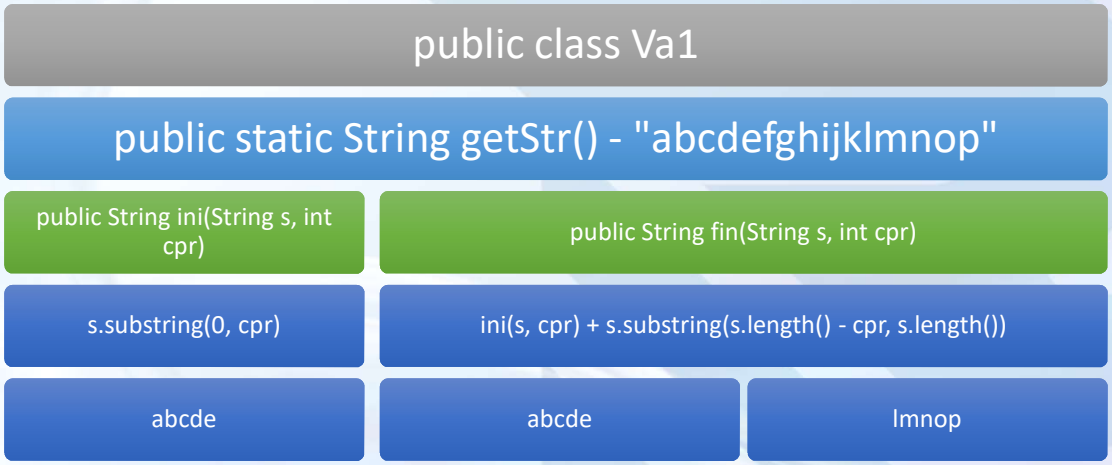
Comentários

- Código:
  - public class Va1 {
    - public static String getStr() { return "abcdefghijklmnop"; }
    - public String ini(String s, int cpr) { return s.substring(0, cpr); }
    - public String fin(String s, int cpr) {
      - return ini(s, cpr) + s.substring(s.length() - cpr, s.length());
    - }
    - public static void main(String[] args) {
      - Va1 o = new Va1();
      - System.out.println(o.ini(o.getStr(), 5));
      - System.out.println(o.fin(o.getStr(), 5));
    - }
  - }

ROGERAOARAUJO www.rogeraoaraujo.com.br

53

Comentários



ROGERAOARAUJO www.rogeraoaraujo.com.br

54

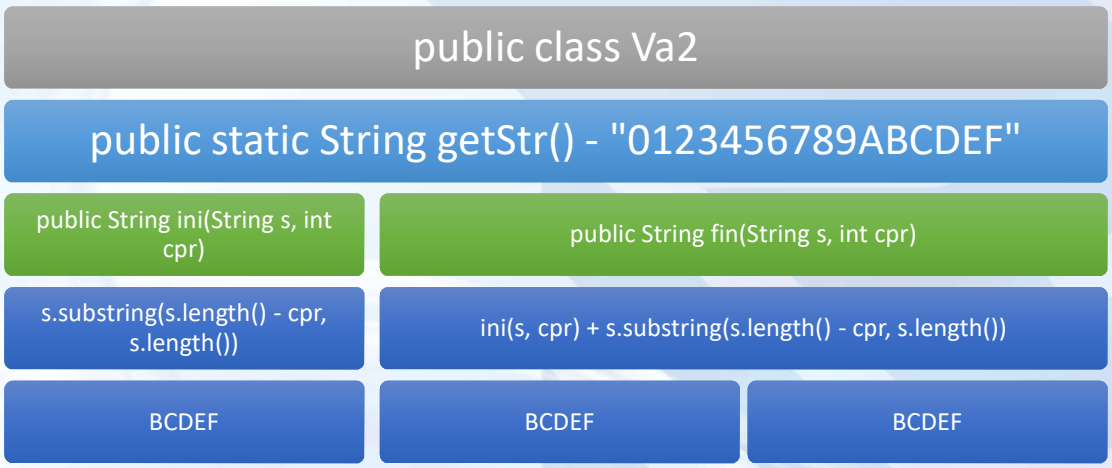
Comentários

- Resultado da execução:
  - abcde
  - abcdelmnop

Comentários

- Código:
  - public class Va2 extends Va1 {
    - public **static** String getStr() { return "**0123456789ABCDEF**"; }
    - public String **ini**(String s, int cpr) { return s.substring(s.length() - cpr, s.length()); }
    - public static void main(String[] args) {
      - **Va2** o = new **Va2**();
      - System.out.println(o.**ini**(o.getStr(), 5));
      - System.out.println(o.**fin**(o.getStr(), 5));
    - }
  - }

Comentários



ROGERAOARAUJO www.rogeraoaraujo.com.br

Comentários

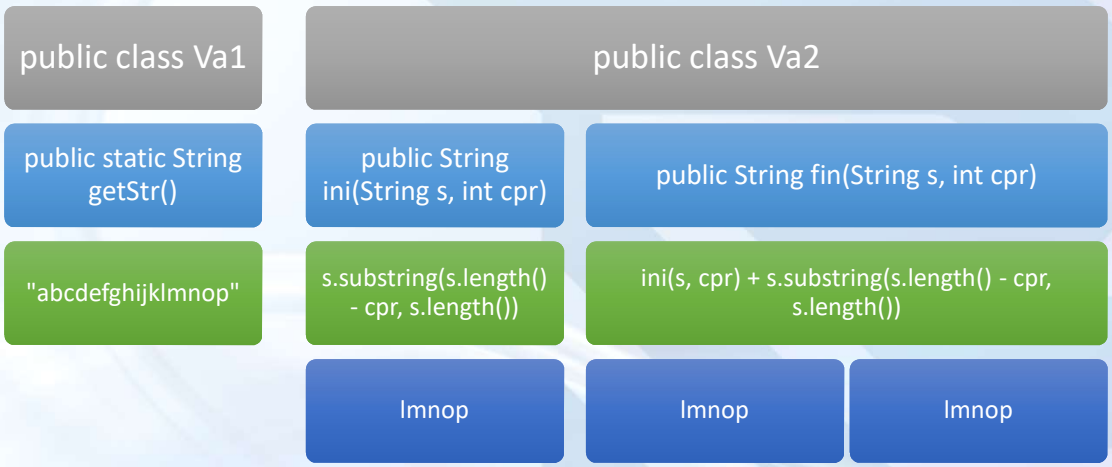
- Resultado da execução:
  - BCDEF
  - BCDEFBCDEF

ROGERAOARAUJO www.rogeraoaraujo.com.br

Comentários

- Código da questão:
  - public class Va2 extends Va1 {
    - public static String getStr() { return "0123456789ABCDEF"; }
    - public String ini(String s, int cpr) { return s.substring(s.length() - cpr, s.length()); }
    - public static void main(String[] args) {
      - Va1 o = new Va2();
      - // ini() foi sobrescrito.
      - System.out.println(o.ini(o.getStr(), 5));
      - System.out.println(o.fin(o.getStr(), 5));
    - }
  - }

Comentários



Comentários

- Resultado da execução:
  - lmnop
  - lmnoplmnop

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] Sejam as seguintes classes Java:

```
public class Va2 extends Va1 {

	public static String getStr() {
		return "0123456789ABCDEF";
	}

	public String ini(String s, int cpr) {
		return s.substring(s.length()-cpr, s.length());
	}

	public static void main(String[] args) {
		Va1 o=new Va2();

		System.out.println(o.fin(o.getStr(), 5));
	}

}
```

Questões de concursos

[CESGRANRIO 2021 Banco do Brasil – Agente de Tecnologia] O que será exibido no console quando o método main for executado?

- [A] 0123BCDE
- [B] BCDEFBCDEF
- [C] 01234BCDEF
- [D] abcdelmnop
- **[E] lmnoplmnop**