

CURSO: SISTEMAS DE INFORMAÇÃO	
PROFESSOR: JOTA PÊ	
TURMA: SINF03NBG4A	DISCIPLINA: PROGRAMAÇÃO DE COMPUTADORES III
AVALIAÇÃO: [] AV1 [x] AV2 [] AV3	
ALUNO 1: DANIEL PEREIRA DA COSTA	RA: 0050019755
ALUNO 2: MATHEUS DIAS BAPTISTA	RA: 0050022794
ALUNO 3: XXXXXXXX	RA:
ALUNO 4: XXXXXXXX	RA:
NOTA:	VISTO DO PROFESSOR:
DATA: até 17 de junho de 2025	

Trabalho de 2ª Avaliação de Programação de Computadores III

COMPETÊNCIA A SER DESENVOLVIDA: construir aplicações em Java manipulando os elementos da tela a partir de eventos executados pelo usuário.

PERÍODO DE ENTREGA: até 17 de junho de 2025.

FORMA DE ENTREGA: realizar o download deste arquivo na sala de aula do Google *Classroom*, colocar o nome e a matrícula dos componentes do grupo, colocar o tema e as regras do jogo no quadro correspondente, salvar esse arquivo em PDF e adicionar esse arquivo a aplicação Java em uma pasta compactada (formato ZIP, pelo amor de Deus não enviar em formato RAR e nenhum outro formato) a atividade “Trabalho de 2ª Avaliação” na sala de aula do Google *Classroom* e marcar a atividade como entregue.

OBSERVAÇÃO: não serão aceitos trabalhos após a data de entrega, grupos com mais de 4 alunos e grupos com aplicações iguais.

Construir uma aplicação Java para simular um mini game de RPG. Neste game deve ter:

- 1 tema: “**OnePieceAp**”
- 4 personagens (jogo para até 4 jogadores): **Luffy, Skanks, Teach, e Buggy.**
- Um número ímpar de mundo
- Em cada mundo um número ímpar de fases/etapas
- 10 perguntas (sobre qualquer tema) para o jogador responder (as perguntas devem estar armazenadas em um banco de dados MySQL)

Forma de jogar:

- Cada jogador escolhe um personagem.
- Cada jogador.
- Caso o trabalho seja feito em grupo, basta um componente do grupo entregar o trabalho, mas lembre-se de colocar o nome de todos os colegas do grupo.

Componentes:

- Grupos tem 2 alunos.
- É permitido fazer individual, em dupla ou em trio.
- Caso o trabalho seja feito em grupo, basta um componente de o grupo entregar o trabalho, mas lembre-se de colocar o nome de todos os colegas do grupo.

Etapas do Trabalho

1. Escolher um tema (Ex.: Sonic, Sistema Solar, Mapa Mundi, etc...).
2. Dentro desse tema, criar um número ímpar de mundos.
3. Dentro de cada mundo, criar um número ímpar de fases/etapas/lugares.

Forma de Entrega:

- Baixar esse documento Word (com as instruções do trabalho)
- Preencher o nome e o RA de todos os componentes do grupo no cabeçalho deste documento
- Colocar nos quadros correspondentes o tema e as regras do game
- Salvar esse documento em formato PDF
- Criar um arquivo compactado (em formato ZIP, não criar formato RAR e nenhum outro formato compactado, pelo amor de Deus)
- No arquivo compactado colocar a aplicação em Java e o documento PDF com os nomes e as regras do game
- Enviar o arquivo compactado na postagem do trabalho no *Classroom* e marcar como entregue

Tema do Game

OnePieceUPica

Regras do Game

O jogo é um quiz interativo baseado em *One Piece*, projetado para até 4 jogadores, com perguntas organizadas em grupos (mundos) armazenadas em um banco de dados MySQL (onepiece_quiz). A interface gráfica (Mapa) exibe um mapa interativo para posicionar personagens e selecionar perguntas, com mecânicas competitivas para determinar o vencedor em cada mundo.

Regras e Mecânicas:

1. **Jogadores e Personagens:**
 - O jogo suporta até 4 jogadores, cada um associado a um personagem (Luffy, Shanks, Teach, Buggy).
 - Cada jogador seleciona um personagem no início da partida.
2. **Estrutura dos Mundos:**
 - O jogo é dividido em 5 mundos, cada um com 5 posições disponíveis para os jogadores colocarem seus personagens.

- As posições representam locais no mapa onde os jogadores avançam ao responder perguntas corretamente.

3. Perguntas e Botões:

- Existem 10 botões de grupos de perguntas, cada um associado a 3 perguntas (total de 30 perguntas).
- As perguntas são selecionadas aleatoriamente, sem repetição dentro de um grupo até que todas sejam exibidas.
- Cada pergunta possui 3 alternativas de resposta (uma correta e duas incorretas).

4. Mecânica de Respostas:

- Um jogador seleciona um botão de grupo, e uma pergunta aleatória é exibida em uma janela modal (JDialog).
- **Resposta Correta:** O jogador posiciona seu personagem em uma das 5 posições disponíveis no mundo atual e escolhe outra pergunta do mesmo grupo.
- **Resposta Incorreta:** A vez passa para o próximo jogador, sem posicionamento no mapa.
- O sistema exibe o resultado ("Correto!" ou "Errado!" com a resposta correta) em uma janela pop-up.


5. Condição de Vitória:


- Em cada mundo, vence o jogador que posicionar seu personagem em 3 ou mais das 5 posições disponíveis, acertando o maior número de perguntas.
- O vencedor do mundo ganha o direito de escolher o próximo mundo a ser jogado.
- O jogo prossegue até que todos os 5 mundos sejam completados, ou conforme definido pelos jogadores.


Notas Adicionais: As perguntas são carregadas de um banco de dados MySQL via QuizController, garantindo aleatoriedade e controle de repetição.


- A interface Mapa exibe o mapa, botões de grupos, e botões para posicionar personagens, usando ícones dos personagens.
- O jogo pode incluir um sumário automático (ex.: número de acertos por jogador, mundo atual) exibido na interface.


Times














Pergunta 1

Pergunta 2

Pergunta 3

Pergunta 4

Pergunta 5

Pergunta 6


Pergunta 7


Pergunta 8


Pergunta 9


Pergunta 10


Times

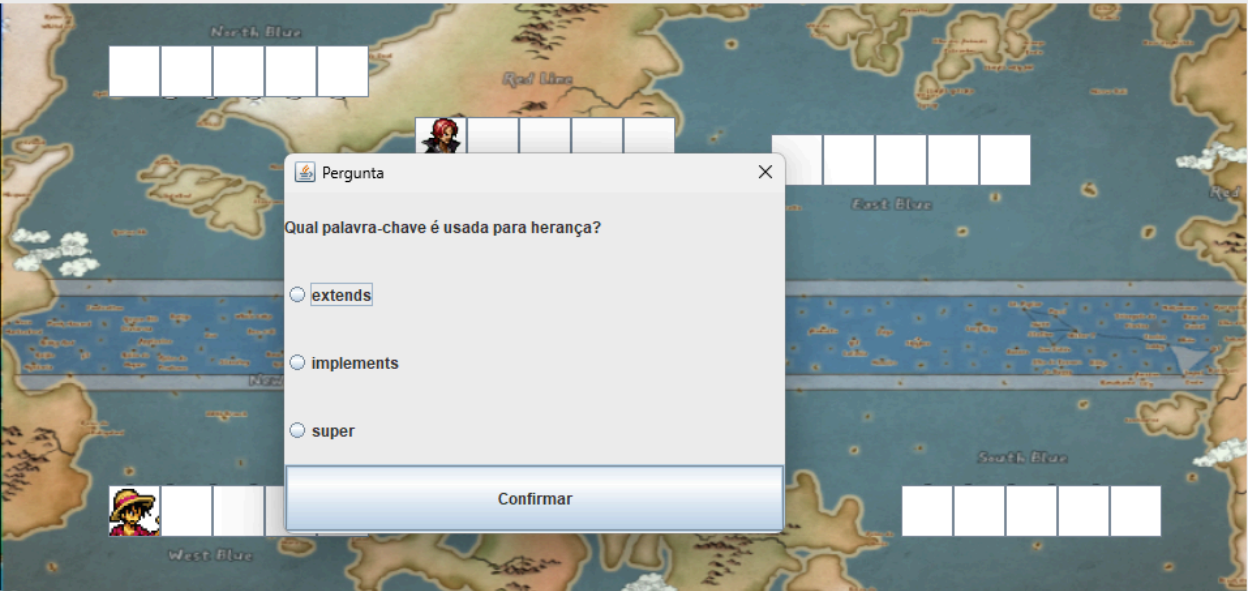












Pergunta 1

Pergunta 2

Pergunta 3

Pergunta 4

Pergunta 5

Pergunta 6

Pergunta 7

Pergunta 8

Pergunta 9

Pergunta 10

Pergunta

Qual palavra-chave é usada para herança?







☐ extends

☐ implements





















☐ super

Confirmar

Times



North Blue



phpMyAdmin

Recente

Favoritos

Novo

information_schema

mysql

oneplace_quiz

Nova

perguntas

performance_schema

ptshop

phpmyadmin

test

Visualizar

Estrutura

SQL

Procurar

Inserir

Exportar

Importar

Privilegios

Operações

Ações

Mostrar tudo

Número de linhas: 50

Filtrar linhas:

Procurar nesta tabela

Ordenar pela chave: Nenhum

Opções extras

			id	grupo	enunciado	opcao1	opcao2	opcao3	respostaCorreta
<input type="checkbox"/>				1	Qual é o nome da máquina virtual que executa progr...	JVM	JRE	JDK	JVM
<input type="checkbox"/>				2	Qual é o tipo de dado primitivo em Java usado para...	long	int	short	int
<input type="checkbox"/>				3	Qual palavra-chave é usada para definir uma classe...	class	interface	struct	class
<input type="checkbox"/>				4	Qual é a palavra-chave usada para criar uma instân...	new	create	instance	new
<input type="checkbox"/>				5	Qual método é chamado automaticamente quando um ob...	start()	init()	<init>()	<init>()
<input type="checkbox"/>				6	Qual é a saída do código: System.out.println(5 + 3)...	11Java	8Java21	8Java3	8Java21
<input type="checkbox"/>				7	Qual é o tipo de dado usado para valores booleanos...	boolean	bool	Boolean	boolean
<input type="checkbox"/>				8	Qual é o modificador de acesso que permite acesso...	public	protected	default	default
<input type="checkbox"/>				9	Qual interface em Java é usada para tomar uma cla...	Comparable	Comparator	Compare	Comparable
<input type="checkbox"/>				10	Qual é o método usado para converter uma string em...	Integer.parseInt()	String.parseInt()	Integer.valueOf()	Integer.parseInt()
<input type="checkbox"/>				11	Qual é a palavra-chave usada para herdar uma class...	implements	extends	inherits	extends
<input type="checkbox"/>				12	Qual exceção é lançada quando se tenta acessar um...	NullPointerException	ArrayIndexOutOfBoundsException	IllegalArgumentException	ArrayIndexOutOfBoundsException
<input type="checkbox"/>				13	Qual é a classe base para manipulação de strings e...	StringBuilder	String	StringBuffer	String
<input type="checkbox"/>				14	Qual é o método principal para iniciar a execução...	run()	main()	start()	main()
<input type="checkbox"/>				15	Qual coleção em Java não permite elementos duplica...	List	Set	Queue	Set
<input type="checkbox"/>				16	Qual é a palavra-chave para sincronizar threads em...	synchronized	thread	lock	synchronized
<input type="checkbox"/>				17	Qual é o operador usado para verificar igualdade d...	==	equals()	compareTo()	equals()
<input type="checkbox"/>				18	Qual é o tipo de retorno padrão de um método que n...	void	null	int	void
<input type="checkbox"/>				19	Qual é o operador para concatenação de strings em...	+	*		*
<input type="checkbox"/>				20	Qual palavra-chave é usada para lançar uma exceção...	throw	throws	try	throw
<input type="checkbox"/>				21	Qual é a superclasse de todas as classes em Java?	Class	Object	Super	Object
<input type="checkbox"/>				22	Qual é o modificador que torna um método acessível...	private	protected	public	private
<input type="checkbox"/>				23	Qual é o mecanismo em Java que permite múltiplas f...	Polimorfismo	Encapsulamento	Herança	Polimorfismo
<input type="checkbox"/>				24	Qual é a palavra-chave para declarar uma variável...	static	final	const	final
<input type="checkbox"/>				25	Qual é a interface para coleções ordenadas em Java...	List	Set	Map	List
<input type="checkbox"/>				26	Qual é a interface usada para iterar sobre coleções...	Iterator	Iterable	Enumeration	Iterator
<input type="checkbox"/>				27	Qual é o resultado de String s = null; System.out...	null	""	Larga NullPointerException	null
<input type="checkbox"/>				28	Qual é o comando para pausar a execução de um thr...	Thread.sleep()	Thread.wait()	Thread.pause()	Thread.sleep()
<input type="checkbox"/>				29	Qual é o pacote padrão para classes de entrada/saí...	java.util	java.io	java.net	java.io
<input type="checkbox"/>				30	Qual é a palavra-chave usada para implementar uma...	extends	implements	uses	implements

Mostrar todos

Com marcados:

Mostrar tudo

Número de linhas: 50

Filtrar linhas:

Procurar nesta tabela

Ordenar pela chave: Nenhum

Operações resultantes das consultas

Imprimir

Descrição do Projeto em JAVA - Sistema de Perguntas Interativo

Objetivo

Criar um sistema interativo para apresentação em sala de aula, onde 10 botões são exibidos, cada um associado a um conjunto de 3 perguntas. Cada vez que um botão é clicado, uma nova pergunta é sorteada (ou exibida sequencialmente) com 3 opções de resposta, totalizando 30 perguntas no banco de dados, cada uma com 3 opções de resposta.

Funcionalidade

- **Estrutura dos Botões:**
 - O sistema possui 10 botões, cada um representando um grupo de perguntas.
 - Cada botão está vinculado a 3 perguntas exclusivas, totalizando 30 perguntas no banco de dados.
- **Comportamento do Botão:**
 - Ao clicar em um botão pela primeira vez, uma pergunta aleatória (ou sequencial) do conjunto de 3 perguntas associadas a esse botão é exibida, junto com suas 3 opções de resposta.
 - Ao clicar novamente no mesmo botão, uma nova pergunta do mesmo conjunto é apresentada, com suas respectivas 3 opções de resposta.
 - O processo se repete até que todas as 3 perguntas do botão sejam exibidas (ou em ordem aleatória, dependendo da implementação).
- **Banco de Dados:**
 - Contém 30 perguntas no total (3 perguntas por botão × 10 botões).
 - Cada pergunta possui 3 opções de resposta.

Exemplo de Fluxo

1. O usuário clica no **Botão 1**.
2. Aparece a **Pergunta 1** do conjunto do Botão 1, com 3 opções de resposta.
3. O usuário clica novamente no **Botão 1**.
4. Aparece a **Pergunta 2** do conjunto do Botão 1, com 3 opções de resposta.
5. O usuário clica mais uma vez no **Botão 1**.
6. Aparece a **Pergunta 3** do conjunto do Botão 1, com 3 opções de resposta.
7. O mesmo comportamento se aplica aos outros 9 botões.

