nome: stiring pacientes : Colecao <paciente> Consultorio(Consultorio(string): abod addPaciente(string): bool addConsultaToPaciente(int, float, string, string): bool printPaciente(int, int, float, string, string): bool printPaciente(int): Paciente(int): void getNome(): string& getCatalFaturado: float findPaciente(int): Paciente* Servico Id: int cuso: float data: string Servico Id: int cuso: float data: string Servico(int, float, string) getId(): int getCusto(): float getData(): string& Servico(int, float, string) getId(): int getCusto(): string& Servico(int, float, string) getId(): int getCusto(): string& Exame Tipologia: enum Tipologia): bool printConsulta(): void addExame(Tloat, string, string) getCusto(): string& Exame Tipologia: enum Tipologia: enum Tipologia): printConsulta(): void addExame(Tloat, string, enum Tipologia) printConsulta(): void getData(): string& Exame Tipologia: enum Tipologia: enum Tipologia): printConsulta(): void printExame(): void</paciente>	Consultorio		Paciente
Consultorio() Consultorio(string) : bool addConsultaToPacente(int, float, string, string) : bool addConsultaToPacente(int); void getNome() : string& getVeno() : strin	nome : string		id : int
Consultorio(string) addPaciente(string): bool addConsultaToPacients(int, float, string, enum Tipologia): bool printPacients(int, float, string, enum Tipologia): bool printPacients(int, float, string, enum Tipologia): bool printPacients(int): void getNome(): string& getTotalFaturador): float getNome(): string& getTotalFaturador): float dindPaciente(int): Pacients*	pacientes : Colecao <paciente></paciente>		nome : string
Consultation(string) addPaciente(string): bool addExameToConsulta(int, int, float, string, string): bool primPacientes(): void primConsultasFromPaciente(int): void getNome(): string8 getTotalFaturado: float findPaciente(int): Paciente* Servico id: int custo: float data: string Servico(int, float, string) getIO(i.int getConsulta(int, int, float, string) getIO(i.int getConsulta(int, int, float, string, string)): bool addExameToConsulta(int, int, float, string, enum Tipologia): bool indConsulta(int): Consulta(int): Consulta(int) Consulta(int): Consulta(int): Consulta(int) int custo: float data: string Servico(int, float, string) getIO(i.int getCousto(): float getCousto	Consultorio()		consultas : Colecao <consulta></consulta>
addPaciente(string): bool addConsulta*(Int, Inf, float, string, string): bool addExame*ToConsulta*(Int, Inf, float, string, enum Tipologia): bool printPaciente(int): void getNome(): string& getTotalFaturado: float flindPaciente(int): Paciente* Servico			
addConsultaToPaciente(int, float, string, string): bool addExameToConsulta(int, int, float, string, enum Tipologia): bool primPaciente(st): void primConsultasFromPaciente(int): void getIotalFaturado: float getCostalFaturado: float, string, string): bool addExameToConsulta(int, int, float, string, enum Tipologia): bool getGotalFaturado: float findPaciente(int): Paciente* Servico			Paciento(int. etring)
addExameToConsulta(int, int, float, string, enum Tipologia) : bool printPacientes() : void printPacientes() : void printConsultasFromPaciente(int) : void getNome() : string& getTotalFaturado() : float addConsulta(float, string, string) : bool addExameToConsulta(int, float, string, enum Tipologia) : bool operator<(Paciente(int) : Paciente* Servico Id : int custo : float data : string Servico(int, float, string) getCusto() : float getCusto() : string& I Exame Tipologia : enum Tipologia) Exame(int, float, string, enum Tipologia) printExame() : void Exame Tipologia : enum Tipologia) Exame(int, float, string, enum Tipologia) printExame() : void Exame(int, float, string, enum Tipologia) printExame() : void			
printPacientes(): void printConsultasFromPaciente(int): void gentNome(): sting& getTotalFaturado: float findPaciente(int): Paciente* Servico id: int custo: float data: string Servico(int, float, string) getTotalFaturado: float findPaciente(int): Paciente* Servico id: int custo: float data: string Servico(int, float, string) getCusto(): float getData(): string& operator<(Consulta(a): tool addExameToConsulta(int, float, string, enum Tipologia): bool printConsulta(int, float, string, string) getCusto(): float getData(): string& O, * Lamber Consulta(int, float, string, string) getCusto(): string& operator<(Consulta(a): string, enum Tipologia): bool printConsulta(): void Exame Tipologia: enum eTipologia: enum Tipologia) printExame(): void Exame(int, float, string, enum Tipologia) printExame(): void			
printConsultasFromPaciente(int) : void getNome() : string& getTotalFaturado : float findPaciente(int) : Paciente*		0, * 1	
getNome(): string& getTotalFaturado: float findPaciente(int): Paciente* addExameToConsulta(int, float, string, enum Tipologia): bool operator<(Paciente&): bool findConsulta(int): Consulta* printConsultas(): void addExameToConsulta(int, int, float, string, enum Tipologia): bool O, * Consulta diagnostico: string exames: Colecao <exame> Consulta(int, float, string, string) getCustoTotal(): float getData(): int getCusto(): float getData(): string& 1 Exame Tipologia: enum eTipologia Exame(int, float, string, enum Tipologia) printExame(): void</exame>			
getTotalFaturado : float findPaciente(int) : Paciente* operator<(Paciente&) : bool findConsulta(int) : Consulta* printConsultas() : void addExameToConsulta(int, int, float, string, enum Tipologia) : bool o, * Consulta diagnostico : string exames : Colecao <exame> Consulta(int, float, string, string) getCustoTotal() : float getDiagnostico() : string& operator<(Consulta(int, float, string, enum Tipologia) : bool addExame(float, string, enum Tipologia) : bool printConsulta() : void Exame Tipologia : enum eTipologia : enum Tipologia) printExame() : void</exame>			
findPaciente(int): Paciente* findConsulta(int): Consulta* printConsultas(): void addExameToConsulta(int, int, float, string, enum Tipologia): box 0,* Consulta			
printConsulta(): void addExameToConsulta(int, int, float, string, enum Tipologia): bood 0, * Consulta			
addExameToConsulta(int, int, float, string, enum Tipologia): book Consulta	findPaciente(int): Paciente*		
Consulta			
Consulta			
Consulta diagnostico: string exames: Colecao <exame> Consulta(int, float, string, string) getCustoTotal(): float getDiagnostico(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void Exame Tipologia: enum eTipologia: enum Tipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void</exame>			0, *
Consulta diagnostico: string exames: Colecao <exame> Consulta(int, float, string, string) getCustoTotal(): float getDiagnostico(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void Exame Tipologia: enum eTipologia: enum Tipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void</exame>			
Consulta diagnostico: string exames: Colecao <exame> Consulta(int, float, string, string) getCustoTotal(): float getDiagnostico(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void Exame Tipologia: enum eTipologia: enum Tipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void</exame>			
Consulta(int, float, string, string)			1
Servico			Consulta
Servico id: int custo: float custo: float data: string Servico(int, float, string) getCustoTotal(): float getDiagnostico(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void Exame Tipologia: enum eTipologia: enum Tipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void			diagnostico : string
id: int custo: float data: string Servico(int, float, string) getCusto(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void Servico(int, float, string) getCusto(): float getDiagnostico(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void Exame Tipologia: enum eTipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void			exames : Colecao <exame></exame>
id: int custo: float data: string Servico(int, float, string) getCusto(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void Servico(int, float, string) addExame(float, string, enum Tipologia): bool printConsulta(): string& 1 Exame Tipologia: enum eTipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void			
id: int custo: float data: string Servico(int, float, string) getId(): int getCusto(): float getDiagnostico(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void 0, * Exame Tipologia: enum eTipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void	Servico		Consulta(int, float, string, string)
custo: float data: string Servico(int, float, string) getId(): int getCusto(): string& operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void	id : int		
data: string operator<(Consulta&): bool addExame(float, string, enum Tipologia): bool printConsulta(): void getId(): int getCusto(): float getData(): string& 1 Exame Tipologia: enum eTipologia Exame(int, float, string, enum Tipologia) printExame(): void	custo : float		
Servico(int, float, string) getId(): int getCusto(): float getData(): string& 1 Exame Tipologia: enum eTipologia: enum Tipologia Exame(int, float, string, enum Tipologia) printExame(): void	data : string		
Servico(int, float, string) getId(): int getCusto(): float getData(): string& D, *			
getId(): int getCusto(): float getData(): string& 1	Complete (int. float, atrice)		
getCusto() : float getData() : string& 1 Exame Tipologia : enum eTipologia : enum Tipologia Exame(int, float, string, enum Tipologia) printExame() : void			
getData(): string& Exame Tipologia: enum eTipologia: enum Tipologia			0, *
Exame Tipologia : enum eTipologia : enum Tipologia Exame(int, float, string, enum Tipologia) printExame() : void			
Exame Tipologia : enum eTipologia : enum Tipologia Exame(int, float, string, enum Tipologia) printExame() : void	getData(): string&		
Tipologia : enum eTipologia : enum Tipologia Exame(int, float, string, enum Tipologia) printExame() : void			
eTipologia : enum Tipologia Exame(int, float, string, enum Tipologia) printExame() : void			
Exame(int, float, string, enum Tipologia) printExame(): void			
printExame() : void			eTipologia : enum Tipologia
printExame() : void			Eveno(int float string survey Timelands)
operator<(Exame&) : void			
· · · · · · · · · · · · · · · · · · ·			operator<(Exame&) : void