//Padari

/\*

class Produto {

nome: string

valor: number

quantidade: number

constructor(nome: string, valor: number, quantidade: number) {

this.nome = nome

this.valor = valor

this.quantidade = quantidade

}

}

class Mantimentos {

nome: string

valor: number

quantidade: number

constructor(nome: string, valor: number, quantidade: number) {

this.nome = nome

this.valor = valor

this.quantidade = quantidade

}

}

class Estoque {

estoqueM: Mantimentos[] = []

estoqueP: Produto[] = []

adicionarMantimentos(mantimento: Mantimentos) {

this.estoqueM.push(mantimento)

}

adicionarProdutos(produto: Produto) {

this.estoqueP.push(produto)

}

}

class Clietes {

nome: string

cidade: string

cpf: string

compra: string

constructor(nome: string, cidade: string, cpf: string, compra: string) {

this.nome = nome

this.cidade = cidade

this.cpf = cpf

this.compra = compra

}

}

class Vendas {

clientes: Clietes[] = []

adicionarVenda(cliete: Clietes){

this.clientes.push(cliete)

}

}

class Padaria {

vendas: Vendas[] = []

estoque: Estoque[] = []

clientes: Clietes[] = []

adicionarVendas(minhasVendas: Vendas){

this.vendas.push(minhasVendas)

}

adicionarEStoque(meuEstoque: Estoque){

this.estoque.push(meuEstoque)

}

adicionarCliente(cliete: Clietes){

this.clientes.push(cliete)

}

}

let produto = new Produto("pao de queijo", 2.00, 20)

let mantimento = new Mantimentos("faria", 15.00, 3)

let comprador = new Clietes("gui" , "sao leo" , "000.000.000-00", "pao de queijo")

let estoque = new Estoque()

let venda = new Vendas()

let padaria = new Padaria()

estoque.adicionarProdutos(produto)

estoque.adicionarMantimentos(mantimento)

venda.adicionarVenda(comprador)

padaria.adicionarEStoque(estoque)

padaria.adicionarVendas(venda)

padaria.adicionarCliente(comprador)

//console.log(venda)

//console.log(estoque.estoqueP)

//console.log(estoque.estoqueM)

//console.log(estoque)

console.log(padaria)

\*/

//Academia

class Treinador {

nome: string

idade: number

tipoDeTreino: string

constructor(nome: string, idade: number, tipoDetreino: string){

this.nome = nome

this.idade = idade

this.tipoDeTreino = tipoDetreino

}

}

class Alunos {

nome: string

idade: number

cpf: string

treinador: Treinador[] = []

constructor(nome: string, idade: number, cpf: string){

this.nome = nome

this.idade = idade

this.cpf = cpf

}

adicionarTreinador(meuTreinador: Treinador){

this.treinador.push(meuTreinador)

}

}

class Mensal {

alunos: Alunos[] = []

alunosMensasl(aluno: Alunos){

this.alunos.push(aluno)

}

}

class Semestral {

alunos: Alunos[] = []

alunosSemestral(aluno: Alunos){

this.alunos.push(aluno)

}

}

class Anual {

alunos: Alunos[] = []

alunosAnual(aluno: Alunos){

this.alunos.push(aluno)

}

}

class Equipamentos {

nomdeDoEquipamento: string

fornecedor: string

preco: number

constructor(nomeDoEquipamento: string, fornecedor: string, preco: number ){

this.nomdeDoEquipamento = nomeDoEquipamento

this.fornecedor = fornecedor

this.preco = preco

}

}

class Sala {

andarDaSala: number

equipamentos: Equipamentos[] = []

treinador: Treinador[] = []

constructor(andarDaSala: number){

this.andarDaSala = andarDaSala

}

adicionarTreinador(treinadorDaSala: Treinador){

this.treinador.push(treinadorDaSala)

}

adicionarEquipamentos(equipamentosDaSala: Equipamentos){

this.equipamentos.push(equipamentosDaSala)

}

}

class Academi {

alunos: Alunos[] = []

treinadores: Treinador[] = []

salas: Sala[] = []

adicionarAlunos(aluno: Alunos){

this.alunos.push(aluno)

}

adicionarTreinadores(treiandor: Treinador){

this.treinadores.push(treiandor)

}

adicionarSalas(sala: Sala){

this.salas.push(sala)

}

}

let treinador1 = new Treinador("marcos", 33 , "inferior")

let treinador2 = new Treinador("joao", 22 , "superior")

let aluno = new Alunos("gustavo", 17, "000.000.000-00")

let equipamento = new Equipamentos("esteira", "pedrin", 2200)

let equipamento2 = new Equipamentos("cruzificio" , "buggati" , 4000)

let sala = new Sala(2)

let academia = new Academi()

sala.adicionarTreinador(treinador1)

sala.adicionarEquipamentos(equipamento)

aluno.adicionarTreinador(treinador1)

academia.adicionarAlunos(aluno)

academia.adicionarTreinadores(treinador1)

academia.adicionarTreinadores(treinador2)

academia.adicionarSalas(sala)

console.log(academia)

//condominio

//condominio

class Morador {

nome: string

apartamento: string

telefone: number

email: string

dependentes: string

veiculo: string

animaisDeEstimacao: string

historicoDpagamento: Aluguel[] = []

contatoDeEmergencia: string

constructor(nome: string, apartamento: string, telefone: number, email: string, dependentes: string,veiculo: string, animaisDeEstimacao: string, contatoDeEmergencia: string){

this.nome = nome

this.apartamento = apartamento

this.telefone = telefone

this.email = email

this.dependentes = dependentes

this.veiculo = veiculo

this.animaisDeEstimacao = animaisDeEstimacao

this.contatoDeEmergencia = contatoDeEmergencia

}

adicionarPagamentoNoHistorico(pagamento: Aluguel){

this.historicoDpagamento.push(pagamento)

}

}

class Aluguel {

valorPago: number

data: string

constructor(valorPago: number, data: string){

this.valorPago = valorPago

this.data = data

}

}

class Pagamento {

valorPago: number

data: string

constructor(valorPago: number, data: string){

this.valorPago = valorPago

this.data = data

}

}

class Funcionario {

nome: string

funcao: string

setor: string

jornadaDeTrabalho: string

contatoDeEmergencia: string

historicoDePagamento: Pagamento[] = []

constructor(nome: string, funcao: string, setor: string, jornadaDeTrabalho: string, contatoDeEmergencia: string){

this.nome = nome

this.funcao = funcao

this.setor = setor

this.jornadaDeTrabalho = jornadaDeTrabalho

this.contatoDeEmergencia = contatoDeEmergencia

}

adicionarPagamentoNoHistorico(pagamento: Pagamento){

this.historicoDePagamento.push(pagamento)

}

}

class AreasComun {

nomeDaArea: string

responsavel: Funcionario

constructor(nomeDaArea: string, responsavel: Funcionario){

this.nomeDaArea = nomeDaArea

this.responsavel = responsavel

}

}

class ControleDeAreas {

agendamento: Morador[] = []

area: AreasComun

constructor(area: AreasComun){

this.area = area

}

adicionarUmaAgendamento( agendar: Morador){

this.agendamento.push(agendar)

}

}

class Manutencao {

tipoDeServico: string

data: string

responsavel: Funcionario

custo: number

descricao: string

constructor(tipoDeServico: string, data: string, responsavel: Funcionario, custo: number, descricao: string){

this.tipoDeServico = tipoDeServico

this.data = data

this.responsavel = responsavel

this.custo = custo

this.descricao = descricao

}

}

class HistoricoDeManutencao {

historico: Manutencao[] = []

adicionarManutencao(manutencao: Manutencao){

this.historico.push(manutencao)

}

}

class EmpressasContratadas {

empresa: string

servico: string

contato: string

constructor(empresa: string, servico: string, contato: string){

;this.empresa = empresa

this.servico = servico

this.contato = contato

}

}

class Servicos {

contratos: EmpressasContratadas[] = []

historicoDePagamento: Pagamento[] = []

adicionarPagamentoNoHistorico(pagamento: Pagamento){

this.historicoDePagamento.push(pagamento)

}

adicionarContrato(contrato: EmpressasContratadas){

this.contratos.push(contrato)

}

}

class Feedback {

feedback: string

usuario: Morador | null

constructor(feedback: string , usuario: Morador | null){

this.feedback = feedback

this.usuario = usuario

}

}