

```

import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;

class Compromisso {
    private String data;
    private String horario;
    private String descricao;

    public Compromisso(String data, String horario, String descricao) {
        this.data = data;
        this.horario = horario;
        this.descricao = descricao;
    }

    public String getData() {
        return data;
    }

    public String getHorario() {
        return horario;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setData(String data) {
        this.data = data;
    }

    public void setHorario(String horario) {
        this.horario = horario;
    }

    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }

    @Override
    public String toString() {
        return "Data: " + data + ", Horário: " + horario + ", Descrição: " + descricao;
    }
}

class Agenda {
    private List<Compromisso> compromissos;

    public Agenda() {
        this.compromissos = new ArrayList<>();
    }

    public void adicionarCompromisso(Compromisso compromisso) {

```

```

        compromissos.add(compromisso);
    }

    public void alterarCompromisso(int index, Compromisso compromisso) {
        if (index >= 0 && index < compromissos.size()) {
            compromissos.set(index, compromisso);
        } else {
            System.out.println("Índice inválido.");
        }
    }

    public void removerCompromisso(int index) {
        if (index >= 0 && index < compromissos.size()) {
            compromissos.remove(index);
        } else {
            System.out.println("Índice inválido.");
        }
    }

    public List<Compromisso> getCompromissos() {
        return compromissos;
    }
}

abstract class AcaoMenu {
    protected Agenda agenda;
    protected Scanner scanner;

    public AcaoMenu(Agenda agenda, Scanner scanner) {
        this.agenda = agenda;
        this.scanner = scanner;
    }

    public abstract void executar();
}

class InserirCompromisso extends AcaoMenu {
    public InserirCompromisso(Agenda agenda, Scanner scanner) {
        super(agenda, scanner);
    }

    @Override
    public void executar() {
        System.out.println("Digite a data do compromisso (DD/MM/AAAA):");
        String data = scanner.nextLine();
        System.out.println("Digite o horário do compromisso (HH:MM):");
        String horario = scanner.nextLine();
        System.out.println("Digite a descrição do compromisso:");
        String descricao = scanner.nextLine();

        Compromisso compromisso = new Compromisso(data, horario,
descricao);
        agenda.adicionarCompromisso(compromisso);
        System.out.println("Compromisso adicionado com sucesso!");
    }
}

```

```

    }
}

class AlterarCompromisso extends AcaoMenu {
    public AlterarCompromisso(Agenda agenda, Scanner scanner) {
        super(agenda, scanner);
    }

    @Override
    public void executar() {
        System.out.println("Digite o índice do compromisso a ser
alterado:");
        int index = scanner.nextInt();
        scanner.nextLine(); // Consumir a quebra de linha

        System.out.println("Digite a nova data do compromisso
(DD/MM/AAAA):");
        String data = scanner.nextLine();
        System.out.println("Digite o novo horário do compromisso
(HH:MM):");
        String horario = scanner.nextLine();
        System.out.println("Digite a nova descrição do compromisso:");
        String descricao = scanner.nextLine();

        Compromisso compromisso = new Compromisso(data, horario,
descricao);
        agenda.alterarCompromisso(index, compromisso);
        System.out.println("Compromisso alterado com sucesso!");
    }
}

class RemoverCompromisso extends AcaoMenu {
    public RemoverCompromisso(Agenda agenda, Scanner scanner) {
        super(agenda, scanner);
    }

    @Override
    public void executar() {
        System.out.println("Digite o índice do compromisso a ser
removido:");
        int index = scanner.nextInt();
        scanner.nextLine(); // Consumir a quebra de linha

        agenda.removerCompromisso(index);
        System.out.println("Compromisso removido com sucesso!");
    }
}

class ListarCompromissos extends AcaoMenu {
    public ListarCompromissos(Agenda agenda, Scanner scanner) {
        super(agenda, scanner);
    }

    @Override

```

```

        public void executar() {
            System.out.println("Compromissos agendados:");
            List<Compromisso> compromissos = agenda.getCompromissos();
            if (compromissos.isEmpty()) {
                System.out.println("Não há compromissos agendados.");
            } else {
                for (int i = 0; i < compromissos.size(); i++) {
                    System.out.println(i + ": " + compromissos.get(i));
                }
            }
        }
    }

class Sair extends AcaoMenu {
    public Sair(Agenda agenda, Scanner scanner) {
        super(agenda, scanner);
    }

    @Override
    public void executar() {
        System.out.println("Saindo...");
        System.exit(0);
    }
}

class Menu {
    private Scanner scanner;
    private Agenda agenda;

    public Menu(Agenda agenda) {
        this.scanner = new Scanner(System.in);
        this.agenda = agenda;
    }

    public void exibirMenu() {
        while (true) {
            System.out.println("\nEscolha uma opção:");
            System.out.println("1. Adicionar compromisso");
            System.out.println("2. Alterar compromisso");
            System.out.println("3. Remover compromisso");
            System.out.println("4. Mostrar compromissos");
            System.out.println("5. Sair");

            int escolha = scanner.nextInt();
            scanner.nextLine(); // Consumir a quebra de linha

            AcaoMenu acao;
            switch (escolha) {
                case 1:
                    acao = new InserirCompromisso(agenda, scanner);
                    break;
                case 2:
                    acao = new AlterarCompromisso(agenda, scanner);
                    break;
            }
        }
    }
}

```

```

        case 3:
            acao = new RemoverCompromisso(agenda, scanner);
            break;
        case 4:
            acao = new ListarCompromissos(agenda, scanner);
            break;
        case 5:
            acao = new Sair(agenda, scanner);
            break;
        default:
            System.out.println("Opção inválida. Por favor,
escolha novamente.");
            continue;
    }
    acao.executar();
}

}

public class Main {
    public static void main(String[] args) {
        Agenda agenda = new Agenda();
        Menu menu = new Menu(agenda);
        menu.exibirMenu();
    }
}

```