< nome do componente = " InspectionProjectProfileManager " >

< versão do perfil = " 1.0 " >

< option name = " myName " value = " Project Default " />

< inspeção\_tool classe = " SpellCheckingInspection " enabled = " false " level = " TYPO " enabled\_by\_default = " false " >

< option name = " processCode " value = " true " />

< option name = " processLiterals " value = " true " />

< option name = " processComments " value = " true " />

</ ferramenta\_de\_inspeção >

</ perfil >

</ componente >

<? xml version = " 1.0 " encoding = " UTF-8 " ?>

< versão do projeto = " 4 " >

< component name = " EntryPointsManager " >

< entry\_points version = " 2.0 " />

</ componente >

< nome do componente = " ProjectKey " >

< option name = " state " value = " project: // e2804f05-5315-4fc6-a121-c522a6c26470 " />

</ componente >

< component name = " ProjectRootManager " version = " 2 " languageLevel = " JDK\_16 " default = " true " project-jdk-name = " 16 " project-jdk-type = " JavaSDK " >

< output url = " file: // $ PROJECT\_DIR $ / out " />

</ componente >

</ projeto >

<? xml version = " 1.0 " encoding = " UTF-8 " ?>

< versão do projeto = " 4 " >

< nome do componente = " ProjectModuleManager " >

< módulos >

< Módulo fileurl = " file: //$PROJECT\_DIR$/filaPilha.iml " filepath = " $ project\_dir $ / filaPilha.iml " />

</ modules >

</ componente >

</ projeto >

< template >

< input-field default = " com.company " > IJ\_BASE\_PACKAGE </ input-field >

</ template >

<? xml version = " 1.0 " encoding = " UTF-8 " ?>

< versão do projeto = " 4 " >

< nome do componente = " VcsDirectoryMappings " >

< mapeamento de diretório = " $ PROJECT\_DIR $ " vcs = " Git " />

</ componente >

</ projeto >

<? xml version = " 1.0 " encoding = " UTF-8 " ?>

< module type = " JAVA\_MODULE " version = " 4 " >

< component name = " NewModuleRootManager " inherit-compiler-output = " true " >

< exclude-output />

< url de conteúdo = " arquivo: // $ MODULE\_DIR $ " >

< sourceFolder url = " file: // $ MODULE\_DIR $ / src " isTestSource = " false " />

</ content >

< orderEntry type = " inheritedJdk " />

< orderEntry type = " sourceFolder " forTests = " false " />

</ componente >

</ módulo >

Package com.company ;

Import java.util.ArrayList ;

Import java.util.Arrays ;

Import java.util.Collections ;

Public class Fila {

Final privado ArrayList < Object > aFilaPaciente = new ArrayList <> ();

Public static void main ( String [] args ) {

Fila fila = novo Fila ();

Boolean sair = false ;

Faça {

Helper . exibe ( menu fila . ());

Switch ( Helper . capturaInteiro ()) {

Caso 1 : {

If (fila . getaFilaPaciente () . size () ! = 2 )

Fila . addPacienteFila (fila . adicionarPaciente ());

Senão

Helper . exibição ( “ Não podem ser considerados mais pacientes à fila “ );

Pausa ;

}

Caso 2 : {

Fila . atenderPaciente (fila . getaFilaPaciente ());

If (fila . getaFilaPaciente () . size () > 0 )

Fila . removePacienteFila ( 0 );

Pausa ;

}

Caso 3 : {

Sair = verdadeiro ;

}

Padrão : {

Helper . exibição ( “ Opção não reconhecida “ );

}

}

} while ( ! sair);

}

/ \*\*

\* Monta o menu principal

\* @return String

\* /

Public String menu () {

String textoMenu;

textoMenu = “ Menu Principal \ r \ n “ ;

textoMenu + = “ 1. Incluir paciente \ r \ n “ ;

textoMenu + = “ 2. Atender paciente \ r \ n “ ;

textoMenu + = “ 3. Sair “ ;

return textoMenu;

}

/ \*\*

\* Captura os dados para inserir um paciente na fila de espera

\* @return Retorna um array com os dados do usuário

\* /

Public ArrayList < Object > adicionarPaciente () {

ArrayList < Object > aDadoPaciente = new ArrayList <> ();

Helper . exibições ( “ Informe o nome do paciente: “ );

aDadoPaciente . add ( Helper . capturaString ());

Helper . exibe ( “ Informe a idade do paciente: “ );

aDadoPaciente . adicionar ( Helper . capturaInteiro ());

Helper . exibições ( “ Informe uma breve descrição do atendimento: “ );

aDadoPaciente . add ( Helper . capturaString ());

return aDadoPaciente;

}

/ \*\*

\* Atende o próximo paciente da fila

\* @param prAFilaPaciente Array que contem os dados da fila e dos pacientes

\* /

Public void atenderPaciente ( ArrayList < Object > prAFilaPaciente ) {

If (prAFilaPaciente . size () > 0 ) {

ArrayList <?> ADadoPaciente = ( ArrayList <?> ) PrAFilaPaciente . get ( 0 );

String textoPaciente;

textoPaciente = “ Próximo paciente \ r \ n “ ;

textoPaciente + = “ Nome: “ + aDadoPaciente . get ( 0 ) + “ \ r \ n “ ;

textoPaciente + = “ Idade: “ + aDadoPaciente . get ( 1 ) + “ \ r \ n “ ;

textoPaciente + = “ Descrição: “ + aDadoPaciente . get ( 2 ) + “ \ r \ n “ ;

Helper . exibe (textoPaciente);

} else {

Helper . exibição ( “ Não há pacientes na fila “ );

}

}

Public void addPacienteFila ( ArrayList < Object > prDadoPaciente ) {

Isso . aFilaPaciente . adicionar (prDadoPaciente);

}

Public void removePacienteFila ( int prIndex ) {

Isso . aFilaPaciente . remover (prIndex);

}

Public ArrayList < Object > getaFilaPaciente () {

Return aFilaPaciente;

}

}

package com.company ;

import java.util.ArrayList ;

import java.util.Arrays ;

import java.util.Collections ;

public class Fila {

final privado ArrayList < Object > aFilaPaciente = new ArrayList <> ();

public static void main ( String [] args ) {

Fila fila = novo Fila ();

boolean sair = false ;

faça {

Helper . exibe ( menu fila . ());

switch ( Helper . capturaInteiro ()) {

caso 1 : {

if (fila . getaFilaPaciente () . size () ! = 2 )

fila . addPacienteFila (fila . adicionarPaciente ());

senão

Helper . exibição ( " Não podem ser considerados mais pacientes à fila " );

pausa ;

}

caso 2 : {

fila . atenderPaciente (fila . getaFilaPaciente ());

if (fila . getaFilaPaciente () . size () > 0 )

fila . removePacienteFila ( 0 );

pausa ;

}

caso 3 : {

sair = verdadeiro ;

}

padrão : {

Helper . exibição ( " Opção não reconhecida " );

}

}

} while ( ! sair);

}

/ \*\*

\* Monta o menu principal

\* @return String

\* /

public String menu () {

String textoMenu;

textoMenu = " Menu Principal \ r \ n " ;

textoMenu + = " 1. Incluir paciente \ r \ n " ;

textoMenu + = " 2. Atender paciente \ r \ n " ;

textoMenu + = " 3. Sair " ;

return textoMenu;

}

/ \*\*

\* Captura os dados para inserir um paciente na fila de espera

\* @return Retorna um array com os dados do usuário

\* /

public ArrayList < Object > adicionarPaciente () {

ArrayList < Object > aDadoPaciente = new ArrayList <> ();

Helper . exibições ( " Informe o nome do paciente: " );

aDadoPaciente . add ( Helper . capturaString ());

Helper . exibe ( " Informe a idade do paciente: " );

aDadoPaciente . adicionar ( Helper . capturaInteiro ());

Helper . exibições ( " Informe uma breve descrição do atendimento: " );

aDadoPaciente . add ( Helper . capturaString ());

return aDadoPaciente;

}

/ \*\*

\* Atende o próximo paciente da fila

\* @param prAFilaPaciente Array que contem os dados da fila e dos pacientes

\* /

public void atenderPaciente ( ArrayList < Object > prAFilaPaciente ) {

if (prAFilaPaciente . size () > 0 ) {

ArrayList <?> ADadoPaciente = ( ArrayList <?> ) PrAFilaPaciente . get ( 0 );

String textoPaciente;

textoPaciente = " Próximo paciente \ r \ n " ;

textoPaciente + = " Nome: " + aDadoPaciente . get ( 0 ) + " \ r \ n " ;

textoPaciente + = " Idade: " + aDadoPaciente . get ( 1 ) + " \ r \ n " ;

textoPaciente + = " Descrição: " + aDadoPaciente . get ( 2 ) + " \ r \ n " ;

Helper . exibe (textoPaciente);

} else {

Helper . exibição ( " Não há pacientes na fila " );

}

}

public void addPacienteFila ( ArrayList < Object > prDadoPaciente ) {

isso . aFilaPaciente . adicionar (prDadoPaciente);

}

public void removePacienteFila ( int prIndex ) {

isso . aFilaPaciente . remover (prIndex);

}

public ArrayList < Object > getaFilaPaciente () {

return aFilaPaciente;

}

}