



Manual Técnico

1º FASE PROJETO – QUIOSQUE CHECK-UP

Afonso Cunha nº 180221017 && Marco Pereira nº 180221019
2018/2019

Programação Orientada por Objetos | 17/05/2019

Licenciatura Engenharia Informática | 2º Semestre

Docentes: Noémia Ferro e José Cordeiro

Conteúdo

Introdução	2
Raciocínio do Projeto	2
Diagrama de Classes	7
Conclusão	8

Introdução

Nesta primeira fase do projeto de Programação Orientada por Objetos, pretende-se implementar um quiosque de Check-Up.

Neste projeto pretendemos desenvolver o programa principal e um modulo que é utilizado para acionar e controlar todos os diapositivos/ equipamentos e que irá apresentar os resultados no ecrã ligando às informações de testes (ClinicalRecord) dos clientes.

Assim, nesta fase tentámos utilizar e aplicar os conteúdos obtidos nas aulas, conhecimentos tais como interfaces, hierarquias, polimorfismo, entre outros.

Raciocínio do Projeto

Após a discussão inicial sobre a implementação e abordagem mais correta em relação ao problema apresentado, seguindo também as informações do enunciado como guia, podemos concluir que estas classes seriam as mais relevantes/necessárias:

Nota: Apesar de não referirmos aqui, todos os métodos gets, sets, toStrings, foram aplicados se achássemos necessário e que fariam sentido para o desenvolvimento do projeto, tal como, os construtores respetivos com ou sem parâmetros.

- **Person**
Guarda a informação nome, data de nascimento, género e cartão de cidadão
- **Clinical Data**
Guarda a informações sobre os testes que uma pessoa fez, ligando os nomes dos testes e os testes através de um HashMap
- **Buttons**
Esta classe é um enumerado que é depois utilizado na classe VisualMonitor para simular a escolha do utilizador

- **MeasureUnit**

Esta classe é apenas utilizada para facilitar a recolha de dados dos diferentes testes, sendo a sua utilização sugerida no enunciado do projeto

- **RandomHelp**

Esta classe foi criada para facilitar a aleatorização de dados de modo a simular os testes recolhidos das pessoas. Utilizando métodos estáticos esta classe acaba por ser utilizada por todos os testes que necessitem de simulação de resultados

- **Tests**

Esta classe é a superclasse de todos os testes que existem no nosso programa. Tem como atributo uma pessoa que de modo a facilitar a recolha dos testes para a ClinicalRecord da pessoa

- **Audiometer**

Esta classe está encarregue de simular o teste de audiometria tonal que vai verificar deste 500 até 8000 hertz e de -10 a 30 decibéis se o utilizador consegue ou não ouvir

- **BalanceStadiometer**

Esta classe está encarregue de simular o peso, a altura da pessoa e aí calcular o IMC. Também implementa a interface MotionSensor para simular uma certa probabilidade de o utilizador se mexer durante o teste e assim influenciando o resultado do teste, sendo o utilizador obrigado a repeti-lo

- **BloodPressureGauger**

A classe BloodPressureGauger está encarregue de simular os valores das pressões sistólica e diastólica. Também implementa a interface MotionSensor para simular uma certa probabilidade de o utilizador se mexer durante o teste e assim influenciando o resultado do teste, sendo o utilizador obrigado a repeti-lo

- **HorizontalScale**

Esta classe tem a função de simular os valores de do perímetro abdominal.

- **MotionSensor**

A interface MotionSensor é uma interface implementada nas classes BalanceStadiometer, BloodPressureGauger e VisualMonitor. Assim, esta interface adiciona a probabilidade de existirem erros de medição nos testes referidos

- **Oximeter**

Esta classe não só está encarregue de simular o batimento cardíaco de uma pessoa assim como o nível de oxigénio no sangue (percentagem)

- **Thermometer**

Está encarregue de simular o valor da temperatura corporal de uma pessoa

- **Tonometer**

Está encarregue de simular o valor da pressão intraocular de uma pessoa

- **VisualMonitor**

Esta classe tem 3 funções: simular o daltonismo, a acuidade visual e astigmatismo. Também implementa a interface MotionSensor para simular uma certa probabilidade de o utilizador se mexer durante o teste e assim influenciando o resultado do teste, sendo o utilizador obrigado a repeti-lo

- **Module**

Esta classe faz a gestão e está conectada a todos os testes que existem. O Module tem os métodos necessários para aceder e realizar os testes

- **Test**

Esta classe apenas surgiu com o propósito de fazer testes ao nosso programa confirmando e verificando assim o nosso progresso na realização do projeto

- **FileHandler**

Esta classe surgiu com o objetivo de auxiliar o manuseamento de ficheiros contendo as informações dos testes de cada pessoa

- **InputReader**

Esta classe foi estudada e trabalhada em IPOO e é utilizada para ler as opções do utilizador durante o correr do programa

- **TimeAux**

Esta classe não só se preocupa com a validação de datas como também consegue fazer uma espécie de “countdown”, ou seja, a classe é também um auxílio para o desenvolvimento do programa

- **Main**

Esta classe está encarregue de começar o programa como um “AppStart”, recorrendo ao menu e ao modulo

- **Menu**

A classe Menu, como o nome indica, está encarregue do menu, da sua apresentação e da sua fluidez, a rapidez, simplicidade e facilidade com que o utilizador consegue usufruir do menu criado

- **2ªFASE**

- **InitialMenu**

- Menu inicial que apresenta as possibilidades ao utilizador de criar um novo utilizador, carregar um utilizador já existente ou sair

- **AudiometerFx**

- Classe encarregue da transposição do teste Audiometer em JavaFx

- **BalanceFx**

- Classe encarregue da transposição do teste Balance em JavaFx

- **BloodPressureTest**

- Classe encarregue da transposição do teste BloodPressureGauger em JavaFx

- **HorizontalScaleFx**

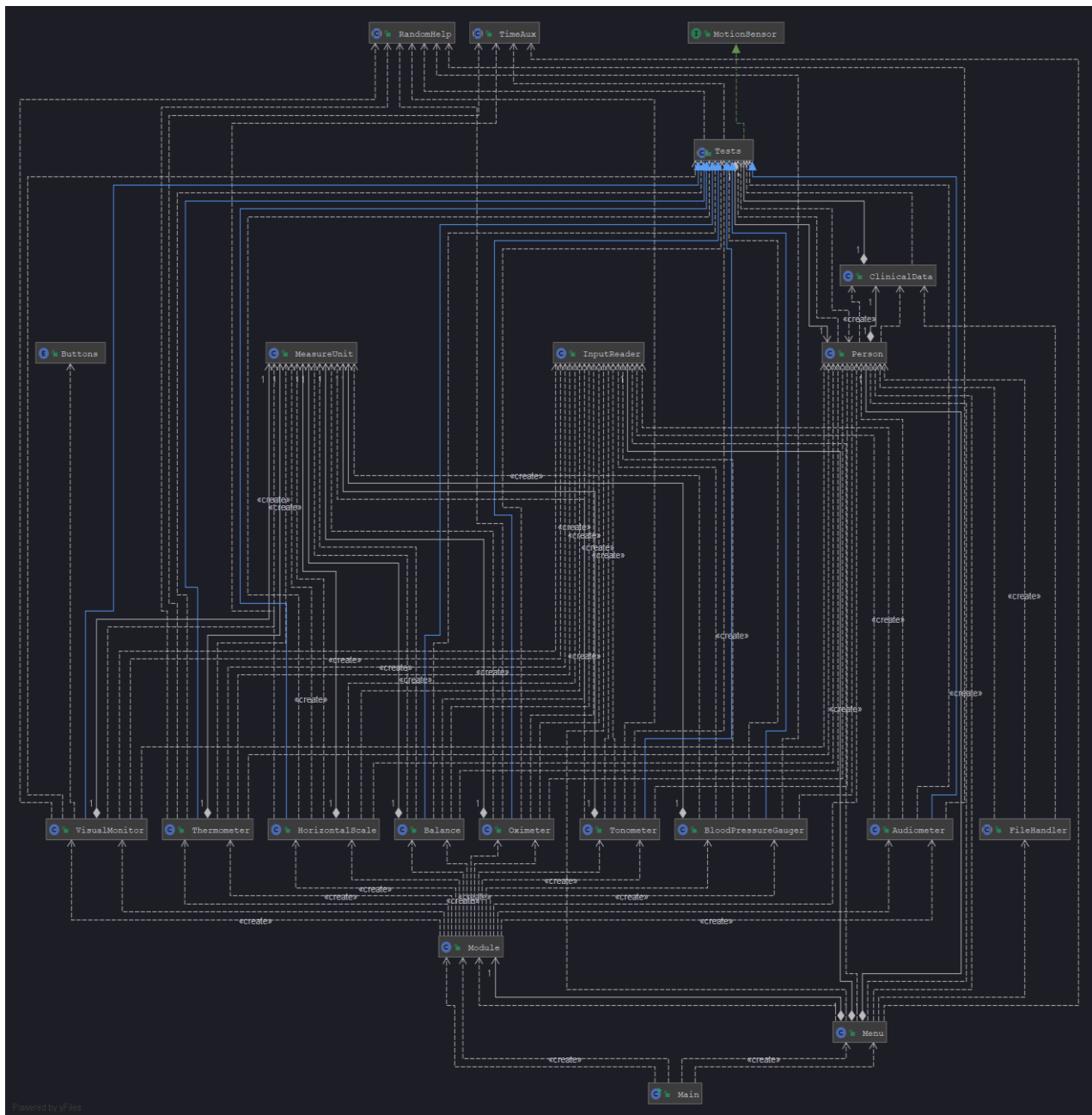
- Classe encarregue da transposição do teste HorizontalScale para JavaFx

- **OximeterFx**

- Classe encarregue da transposição do teste Oximeter para JavaFx

- **ThermometerFx**
 - Classe encarregue da transposição do teste Thermometer para JavaFx
- **Tonometer**
 - Classe encarregue da transposição do teste Tonometer para JavaFx
- **VisualMonitor**
 - Classe encarregue da transposição do teste VisualMonitor para JavaFx
- **CreatePerson**
 - Classe encarregue de manusear a janela de criação de uma pessoa
- **ExitMenu**
 - Classe encarregue do último menu de saída
- **JavaFxMain**
 - Classe encarregue de começar o programa
- **LoadUser**
 - Classe encarregue do menu exibido para carregar uma pessoa já existente
- **MachineMenu**
 - Classe encarregue do menu de todas as máquinas e testes possíveis realizar no nosso projeto

Diagrama de Classes



Conclusão(1ªFase)

Com este projeto tivemos a oportunidade de aperfeiçoar os conteúdos apreendidos nas aulas de IPOO bem como em POO mais especificamente classes abstratas, interfaces, entre outros.

Tivemos o cuidado de implementar o projeto da forma mais organizada e simples que nos foi possível, tendo em atenção os vários pormenores pedidos e as boas práticas de programação.

Foram encontrados alguns percalços previstos e imprevistos, que foram resolvidos, nomeadamente na interpretação do enunciado que contém várias interpretações possíveis.

Durante a realização deste trabalho deparamo-nos com algumas dificuldades tais como sendo uma delas a gestão de tempo. Também existiu a preocupação com a implementação mais correta do código criado, que originou um certo conflito entre ideias acabando por ver o seu desenlace já perto da data de entrega.

Por fim, achamos que o resultado compensou o esforço do grupo e ficámos contentes com o projeto, olhando e pensando já na 2ª fase deste mesmo.

Conclusão(2ªFase)

A realização desta 2ªfase permitiu-nos consolidar a matéria de JavaFx sendo o foco principal tornar o projeto que foi desenvolvido na fase anterior, mais funcional e interativo.

Tal como na fase anterior, existiram alguns imprevistos e percalços, mas que através do esforço e trabalho da equipa, conseguimos resolve-los e ultrapassá-los.

O nosso foco principal e que acabámos por adotar como abordagem foi dar mais peso à parte funcional do programa que nos fez sentido tendo em conta a matéria abordada nesta fase do Projeto.

A maior dificuldade que o grupo sentiu durante a realização deste trabalho foi a sobreposição de atividades (Testes, Projetos, Apresentações, etc) mas que acabou por ser aliviada com o adiamento da data de entrega deste projeto.

Assim, deixamos com encerrado o nosso projeto que retrata um quiosque de check-up com recurso aos nossos conhecimentos adquiridos durante as unidades curriculares de POO e IPOO.

