 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático de Época Recurso	Ano letivo 2018/2019
	Curso Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	
	Unidade Curricular Paradigmas da Programação	

## 1 Destinatários

Este trabalho destina-se a todos os estudantes inscritos na unidade curricular de Paradigmas da Programação em época normal no regime de avaliação final que pretendam obter aprovação à Unidade Curricular.

## 2 Objetivos

Com a realização do trabalho prático, pretende-se que os alunos ponham em prática todos os conhecimentos adquiridos na utilização do paradigma de programação orientado a objetos e a sua implementação na linguagem de programação Java, demonstrando as suas apetências em:

- Conhecer e compreender os conceitos fundamentais associados ao paradigma da programação orientada a objetos;
- Conceber e implementar, para problemas concretos, soluções que tenham por base o paradigma da programação orientada a objetos.
- Reconhecer e compreender a semântica e a sintaxe da linguagem Java.
- Reutilizar, alterar e desenvolver código recorrendo à linguagem Java tendo em vista um determinado problema com regras semânticas específicas.

Considere ainda que:

- Não é permitida a utilização de API's/conceitos Java que não tenham sido alvo de lecionação - ano letivo 2018-2019 - da unidade curricular Paradigmas de Programação. Os alunos que pretendam utilizar API's adicionais devem atempadamente pedir autorização a um dos docentes da unidade curricular.
- Não é permitida a utilização de coleções Java predefinidas (*Java Collections Framework*).
- Os recursos de suporte ao trabalho referenciados no enunciado, são de utilização obrigatória.

## 3 Enunciado

Para avaliar a performance e o domínio de compreensão da matéria lecionada numa unidade curricular pelos alunos é necessário testar o seu conhecimento. Neste sentido é proposto, neste trabalho prático o desenvolvimento de uma plataforma de testes de autoavaliação. Esta será uma ferramenta importante para diagnosticar a compreensão de vários conteúdos de aprendizagem associados a uma determinada unidade curricular, que seja objeto de estudo.

A aplicação a desenvolver terá como objetivo carregar testes a partir de um ficheiro criado previamente e expor aos alunos um conjunto de questões de resposta variada. Para cada resposta é anotado o tempo de início o tempo de fim e a resposta fornecida. No final do teste, após responder à última pergunta será apresentado o número de respostas certas e erradas, bem como o tempo médio e desvio padrão do tempo médio gasto em cada pergunta.

### 3.1 Descrição Técnica

O desenvolvimento da aplicação irá obrigar ao cumprimento de contratos já pré-estabelecidos para a definição de questões. São fornecidos os contratos:

- IQuestion, - contrato de uma questão genérica;
- IQuestionMetadata, contrato com os metadatos de uma questão;
- IQuestionYesNo – contrato de uma questão sim e não;
- IQuestionNumeric – contrato de uma questão numérica;

<p><b>P.PORTO</b></p> <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	Tipo de Prova Trabalho Prático de Época Recurso	Ano letivo 2018/2019
	Curso Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	
	Unidade Curricular Paradigmas da Programação	

- IQuestionMultipleChoice -contrato de uma questão de escolha múltipla.

Observe as relações entre os contratos na figura 1. Devem ser criadas classes em java que implementem da forma mais adequada as interfaces descritas.

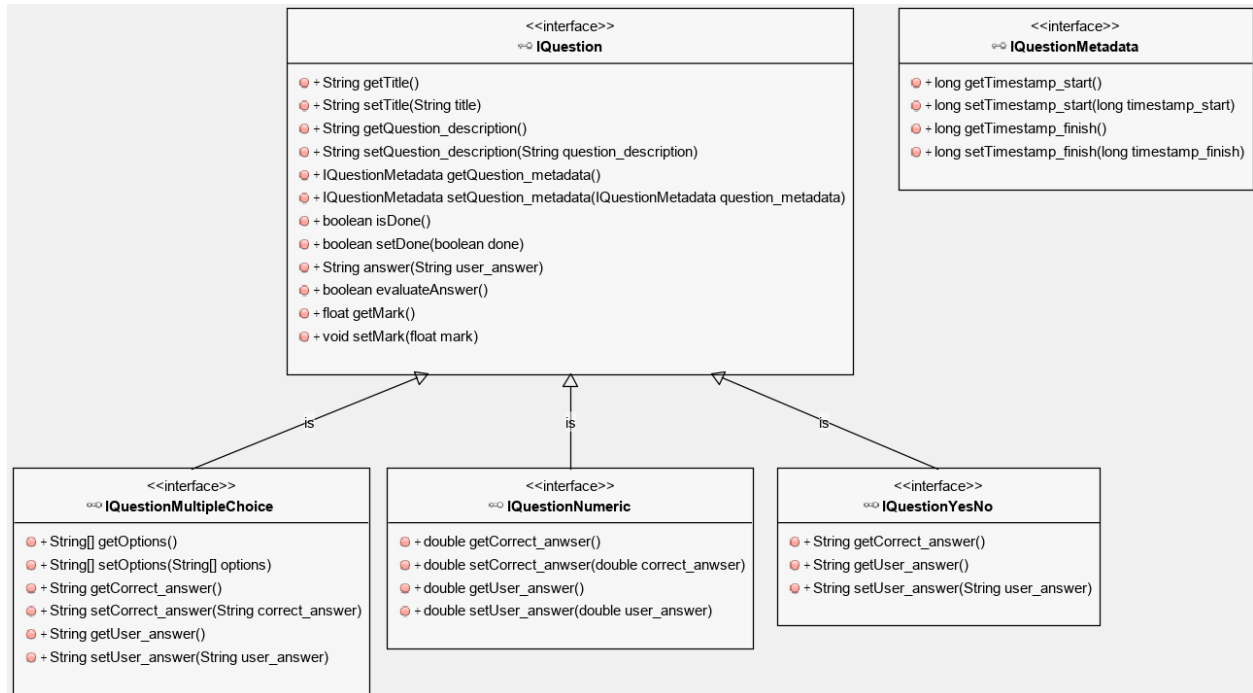


Figura 1 - Contratos relacionados com Questions

Para guardar as questões de um determinado teste numa coleção deve usar o contrato ITest. Esta interface, por sua vez usa um objeto que implementa a o contrato ITestStatistics para o cálculo das estatísticas sobre as respostas a um determinado teste e um objecto que implementa o contrato IScoreStrategy para avaliar os testes segundo uma estratégia de classificação a definir (ex: Não Satisfaz, Satisfaz, Satisfaz bastante e Excelente) a partir dos pontos obtidos no teste.

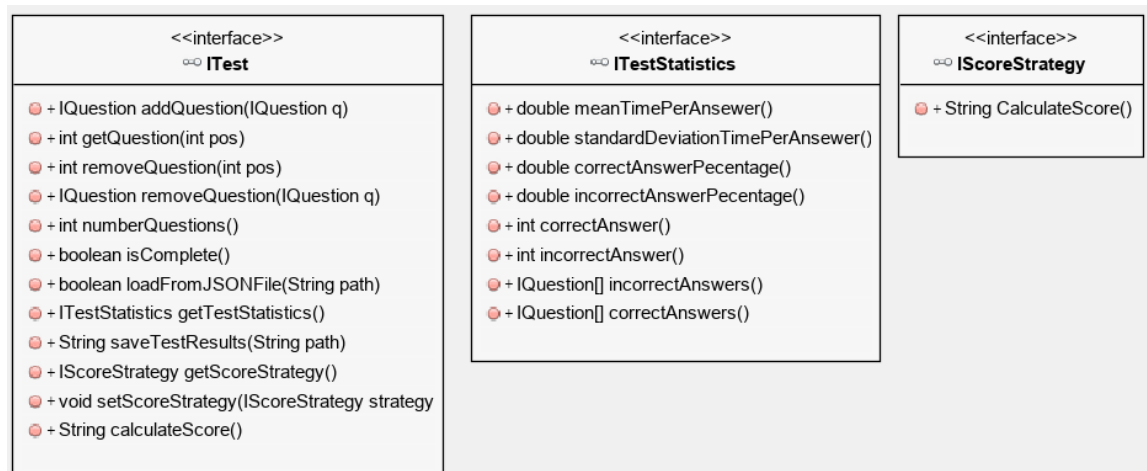


Figura 2 - Contratos para relacionados com testes

<p><b>P.PORTO</b></p> <p>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</p>	Tipo de Prova Trabalho Prático de Época Recurso	Ano letivo 2018/2019
	Curso Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	
	Unidade Curricular Paradigmas da Programação	

Deve implementar os contratos definidos usando as estruturas adequadas. O uso dos contratos fornecidos nos recursos é obrigatório.

Para animar o teste, foi criada uma camada de visualização desenvolvida em JAVA, usando a biblioteca SWING que executa os testes baseando-se nas interfaces acima definidas. A representação gráfica de um teste pode ser iniciada a partir do seguinte excerto de código. Neste código é passado um objeto da classe Test, que deve ser criada, que implementa a interface ITest.

```
public static void main(String[] args) throws TestException {

    System.out.println("Inicio de Teste!");

    //Carregar o test
    ITest demoTest = new Test();
    demoTest.loadFromJSONFile("data/teste_A.json");

    //Executar o teste na camada gráfica
    TestWindow t = new TestWindow();
    t.startTest(demoTest);

    //Obter os resultados do teste
    System.out.println("Teste Efectuado!");
    System.out.println(demoTest.toString());

}
```

Figura 3 - Execução um teste na camada gráfica

A classe TestWindow encontra-se implementada no ficheiro resources.jar disponibilizado na pasta de recursos. Devem importar a biblioteca para o projeto e fazer uso dela.

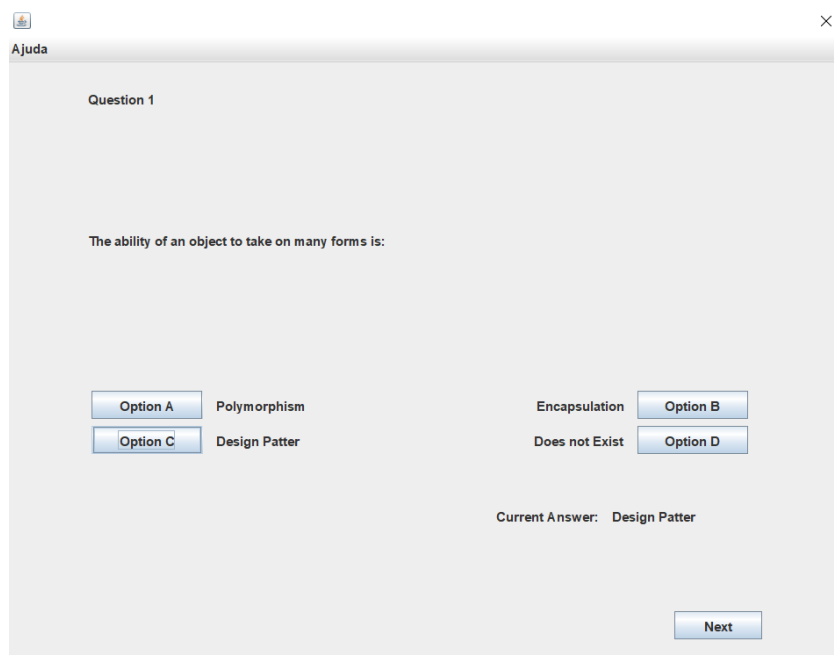


Figura 4 - Camada gráfica disponibilizada


<b>P.PORTO</b> <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático de Época Recurso	Ano letivo 2018/2019
	Curso Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	
	Unidade Curricular Paradigmas da Programação	

O formato do ficheiro de teste encontra-se em *json* e pode utilizar bibliotecas para fazer o parsing do ficheiro. É sugerido o uso da biblioteca GSON da google.

A estrutura do ficheiro de texto segue a forma de um array, onde são definidas questões em cada posição do array do ficheiro *json*. Um exemplo encontra-se disponível na pasta *data* no ficheiro com o nome *teste\_A.json*.

*Tabela 1 - Conteúdo do ficheiro test\_A.json*

```
[
  {
    "type": "MultipleChoice",
    "question": {
      "title": "Question 1",
      "score": 4,
      "mark": 5,
      "question_description": "The ability of an object to take on many forms
is:",
      "options": [
        "Polymorphism",
        "Encapsulation",
        "Design Patter",
        "Does not Exist"
      ],
      "correct_answer": "Polymorphism"
    }
  },
  {
    "type": "MultipleChoice",
    "question": {
      "title": "Question 2",
      "score": 4,
      "mark": 5,
      "question_description": "The bundling of data with the methods that
operate on that data is:",
      "options": [
        "Polymorphism",
        "Encapsulation",
        "Design Patter",
        "Does not Exist"
      ],
      "correct_answer": "Encapsulation"
    }
  },
  {
    "type": "YesNo",
    "question": {
```

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático de Época Recurso	Ano letivo 2018/2019
	Curso Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	
	Unidade Curricular Paradigmas da Programação	

```

        "title": "Question 3",
        "score": 4,
        "mark": 5,
        "question_description": "Object Oriented Programming is exclusive to the
JAVA programming language",
        "correct_answer": "no"
    }
},
{
    "type": "Numeric",
    "question": {
        "title": "Question 4",
        "score": 4,
        "mark": 5,
        "question_description": "How many programming languages are taught in
Paradigmas de Programação?",
        "correct_answer": "1"
    }
}
]

```

### 3.2 Requisitos Mínimos


O trabalho submetido para avaliação deve no mínimo ser capaz de executar e calcular estatísticas sobre o tempo de resposta a cada questão. O trabalho a apresentar deve conter no mínimo as seguintes funcionalidades corretamente implementadas:

- Implementação de todos os contratos definidos no enunciado;
- Carregamento de um ficheiro de texto representando um teste;
- Deve utilizar a camada visual para executar cada teste. No caso de ter problemas deve no mínimo implementar um menu a ser apresentado na consola de JAVA da aplicação e executar um teste com as classes criadas;
- Apresentar e implementar um contrato para armazenar os testes já respondidos pelo utilizador;
- Guardar o resultado do teste, com os metadados e estatística associada num ficheiro de texto;
- Através do mecanismo de herança e lançamento de exceções, crie uma subclasse adicional que implemente o contrato ITest que apenas permita executar testes com respostas de escolha múltipla. Deve lançar exceções caso sejam adicionas questões de outro tipo à coleção.

### 3.3 Requisitos de Bonificação

Como requisitos de bonificação é sugerido que complete o trabalho prático com as seguintes funcionalidades:

- Ordene os testes respondidos pelo utilizador pelo número de respostas certas;
- Crie e implemente um novo contrato para estatísticas adicionais sobre um teste (ex: percentagem de respostas numéricas certas, percentagem de respostas numéricas erradas, etc)

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Trabalho Prático de Época Recurso	Ano letivo 2018/2019
	Curso Licenciatura em Engenharia Informática e Licenciatura em Segurança Informática em Redes de Computadores	
	Unidade Curricular Paradigmas da Programação	

## 4 Elaboração do trabalho

Este trabalho é realizado em grupo que deverá ser composto no máximo por 2 alunos da unidade curricular. Os alunos devem comunicar atempadamente o seu grupo de trabalho na plataforma moodle, até ao dia **03 de Julho de 2019**.

### 4.1 Datas e considerações

O trabalho deve ser entregue até às **23h55** horas do dia **03 de Julho de 2019**, devendo a entrega ser feita através da página da unidade curricular de Paradigmas de Programação em <http://moodle.estgf.ipp.pt>. A defesa do trabalho será realizada no dia **04 de Julho de 2019**. A defesa será realizada por turnos e a data exata para cada aluno (tendo em consideração o horário da respetiva turma) será comunicada na plataforma moodle após a entrega do trabalho. No caso de um grupo ser constituído por alunos de turmas diferentes, a marcação do horário será aleatória.

Considera-se por defesa satisfatória, quando o aluno demonstra que realizou o trabalho submetido e que domina todos os conceitos de programação orientada a objetos aplicados na resolução do trabalho. Tentativas de fraude, resultarão na avaliação do trabalho como: Fraude Académica.

#### Observações:

A entrega dos trabalhos deve obedecer aos requisitos identificados no ponto Formato da entrega. Os alunos que não realizem a entrega do trabalho até à data/hora definida serão sujeitos a penalização.

### 4.2 Formato da entrega

Os trabalhos entregues deverão evitar (se possível) utilizar caminhos absolutos ou endereços específicos, de modo a que possam ser facilmente utilizados em qualquer máquina. Para além disso, e no sentido de facilitar a receção dos vários trabalhos recebidos, estes deverão observar as seguintes regras:

- Todos os elementos do grupo deverão submeter o trabalho no link respetivo (Entrega do Trabalho);
- O trabalho desenvolvido deverá ser entregue através do moodle, através da submissão de um ficheiro com o nome PP\_AF\_<nr\_do\_aluno>\_<nr\_do\_aluno>.zip, contendo:
  - Os ficheiros criados incluindo o(s) projeto(s) do IDE Netbeans e uma pasta com a distribuição (jar) da solução proposta.
  - Recorra a comentários JavaDoc, e não só, de modo a documentar, o mais exaustivamente possível, o código desenvolvido.
- Cada ficheiro de código entregue por cada grupo terá de possuir no início do mesmo um comentário com pelo menos a seguinte informação (com as adaptações óbvias para cada aluno/grupo):

```

/*
 * Nome: <Nome completo do aluno>
 * Número: <Número mecanográfico do aluno>
 * Turma: <Turma do aluno>
 *
 * Nome: <Nome completo do colega de grupo>
 * Número: <Número mecanográfico do colega de grupo>
 * Turma: <Turma do colega de grupo>
 */

```