3º Trabalho[1]

Investimentos

Quarta-feira, 24 de março de 2021.

Desenvolva um programa chamado CalculoInvestimentos usando as classes apresentadas no diagrama de classes abaixo.

Segue as definições dessas classes.

1. Classe Imposto

```
#ifndef IMPOSTO_H
#define IMPOSTO_H

/*
   Define as alíquotas de Imposto de Renda (IR) a serem aplicadas sobre o rendimento bruto do investimento.
*/
class Imposto {
public:
      // Taxa de 22.5% de IR a ser calculado sobre o rendimento bruto de um investimento de até 180 dias.
      static const float TAXA_IR_22_5;

      // Taxa de 20% de IR a ser calculado sobre o rendimento bruto de um investimento de 181 a 360 dias.
      static const float TAXA_IR_20;

      // Taxa de 17.5% de IR a ser calculado sobre o rendimento bruto de um investimento de 361 a 720 dias.
      static const float TAXA_IR_17_5;

      // Taxa de 15% de IR a ser calculado sobre o rendimento bruto de um investimento acima de 720 dias.
      static const float TAXA_IR_15;

      Imposto();
```

[1] Atualizado em 25/03/2021.

```
        Imposto(float valor);

        Imposto& setImposto(float valor);
        float getImposto();

private:
        float valor;
};
#endif
```

2. Classe Estratégia

```
#ifndef ESTRATEGIA_H
#define ESTRATEGIA_H

#include <string>

using namespace std;

enum class EstrategiaInvestimento { INFLACAO, PREFIXADO, POS_FIXADO, RENDA_VARIAVEL,
                                    INTERNACIONAL, MULTIMERCADO, ALTERNATIVO };

/*
        Define as estratégias dos diferentes tipos de investimento.
*/
class Estrategia {
public:
        Estrategia();
        Estrategia(EstrategiaInvestimento estrategia);

        /* Obtém uma descrição corresponde ao tipo de estratégia de um investimento.

           O valor resultante, segundo uma das constantes definidas na enumeração EstrategiaInvestimento,
           corresponde a uma string literal a seguir:

           "Inflação", "Prefixado", "Pós-fixado", "Renda Variável", "Internacional", "Multimercado" ou
           "Alternativo".
        */
        static string obterEstrategia(EstrategiaInvestimento estrategia);

        Estrategia& setEstrategia(EstrategiaInvestimento estrategia);
        EstrategiaInvestimento getEstrategia();

private:
        EstrategiaInvestimento estrategia;
};
#endif
```

3. Classe Investimento

```
#ifndef INVESTIMENTO_H
#define INVESTIMENTO_H

#include <string>
#include "Estrategia.h"
```

```
#include "Imposto.h"

using namespace std;

class Investimento {
public:
	Investimento();
	Investimento(string nome);

	Investimento& setNome(string nome);
	string getNome() const;

	Investimento& setRating(string rating);
	string getRating() const;

	Investimento& setProtecaoFgc(bool protecaoFgc);
	bool getProtecaoFgc();

	Investimento& setEstrategia(Estrategia estrategia);
	Estrategia getEstrategia();

	Investimento& setValorInvestido(float valorInvestido);
	float getValorInvestido();

	Investimento& setDataInvestimento(string dataInvestimento);
	string getDataInvestimento();

	Investimento& setDataResgate(string dataResgate);
	string getDataResgate();

	// Define o prazo em dias úteis.
	Investimento& setPrazo(short prazo);

	// Retorna o prazo em dias úteis.
	virtual short getPrazo();

	// Define a taxa percentual ao ano (a.a.).
	Investimento& setTaxa(float taxa);

	// Retorna a taxa percentual ao ano (a.a.).
	float getTaxa();

	Investimento& setImposto(Imposto imposto);
	Imposto getImposto();

	/* Retorna uma string que representa os dados do investimento. Usa o formato abaixo:

			<nome> | <rating> | <estratégia> | FGC: <sim | não>

	   Veja nos exemplos abaixo como os valores devem substituir as strings no formato acima
	   representadas por <>.
```

```
        O caractere pipe (|) entre os parênteses angulares < > significa ou, por exemplo: <sim | não> resulta
        em sim ou não.

        Exemplos:
                                CDB Original Jan/2020 | BBB+ | Prefixado | FGC: sim
                                Ações da Vale | AA+ | Renda Variável | FGC: não
    */
    virtual string toSring();

    // Calcula o valor do Imposto de Renda (IR) sobre o rendimento obtido no investimento.
    virtual float calcularIR() = 0;

private:
    // Nome do investimento, por exemplo, CDB, Tesouro Direto, LCI, LCA , Ações, etc.
    string nome;

    /* O rating, ou classificação de risco, é a nota dada a uma empresa, um país, um título ou uma
       operação financeira para medir o risco de crédito. Serve para indicar a capacidade de um país ou
       empresa de pagar suas dívidas e as chances de não conseguir, atrasando o pagamento ou dando
       calote. Os investidores usam os “ratings” para tomarem decisões na hora de aplicar seu dinheiro.

       Valores válidos de rating, do menor risco, alto grau de investimento (AAA), para o maior risco, baixo
       grau de investimento (D).

       Grau de investimento com qualidade alta e baixo risco: AAA, AA+, AA, AA-, A+, A, A-
       Grau de investimento com qualidade baixa: BBB+, BBB, BBB-
       Categoria de especulação, baixa classificação: BB+, BB, BB-, B+, B, B-
       Risco de inadimplência (calote) e baixo interesse: CCC+, CCC, CCC-, CC, C, D
    */
    string rating;

    /* Indica se o investimento tem proteção do Fundo Garantidor de Créditos (FGC).

            true: tem proteção
            false: sem proteção

      Os investimento que tem proteção do FGC são: Poupança, CDB, LCI, LCA e Letras de Câmbio (LC).
    */
    bool protecaoFgc;

    // Indica o tipo de estratégia usada no Investimento.
    Estrategia estrategia;

    // Valor investido no investimento.
    float valorInvestido;

    // Data em que o investimento foi realizado, ou seja, em que o dinheiro foi investido.
    string dataInvestimento;

    // Data em que o dinheiro pode ser resgatado do investimento.
    string dataResgate;
```

```
        /* Período em dias úteis do investimento. Representa a diferença entre as datas de resgate e
           investimento.
        */
        short prazo;

        /* Valor percentual usado para rentabilizar ou remunerar o valor investido. A taxa (juros) deve ser
           especificada ao ano (a.a.).

           Na renda fixa a taxa do investimento é conhecida na data em que o dinheiro é investido, e na renda
           variável a taxa só é conhecida na data de resgate do investimento.
        */
        float taxa;

        /* Taxa de imposto de renda a ser calculado sobre o rendimento bruto de um investimento de acordo
           com o seu prazo em dias.
        */
        Imposto imposto;
};
#endif // !INVESTIMENTO_H
```

4. Classe Renda Fixa

```
#ifndef RENDA_FIXA_H
#define RENDA_FIXA_H

#include "Investimento.h"
#include "Imposto.h"

class RendaFixa : public Investimento {
public:
        RendaFixa();
        RendaFixa(string nome);

        /* Retorna uma string que representa os dados do investimento. Usa o formato abaixo:

           <nome> | <rating> | <estratégia> | FGC: <sim | não> | Prazo: <X> anos [e <Y> meses] | [ <Y> meses]

           Veja nos exemplos abaixo como os valores devem substituir as strings no formato acima
           representadas por <>. O caractere pipe (|) entre os parênteses angulares < > significa ou, por
           exemplo: <sim | não> resulta em sim ou não.

           Na renda fixa o prazo deve ser exibido em meses se o período de investimento for inferior a um ano
           (até 12 meses), caso contrário, em anos.

           Os colchetes indicam uso opcional e que será usado no caso em que o período de investimento não
           representar um ano completo, por exemplo, 5 anos e 6 meses.

           Exemplos:
                        CDB Original Jan/2020 | BBB+ | Inflação | FGC: sim | Prazo: 3 anos e 4 meses
                        CDB PAN Nov/2025 | BBB+ | Pós-fixado | FGC: sim | Prazo: 4 anos
                        CRI WS | AA+ | Prefixado | FGC: não | Prazo: 11 meses
        */
        virtual string toSring();
```

```
        /* Retorna o prazo em meses se o período de investimento for inferior a um ano (12 meses), caso
           contrário, em anos.
        */
        virtual short getPrazo();

        /* Calcula o valor do Imposto de Renda (IR) sobre o rendimento obtido no investimento.
           Investimentos de renda fixa possuem quatro alíquotas de IR, definidas na classe Imposto segundo o
           prazo do investimento.
        */
        virtual float calcularIR();
};
#endif // !RENDA_FIXA_H
```

5. Classe Renda Variável

```
#ifndef RENDA_VARIAVEL_H
#define RENDA_VARIAVEL_H

#include "Investimento.h"

class RendaVariavel : public Investimento {
public:
        RendaVariavel();
        RendaVariavel(string nome);

        /* Retorna uma string que representa os dados do investimento. Usa o formato abaixo:

           <nome> | <rating> | <estratégia> | FGC: <sim | não> | Prazo: <X> anos [e <Y> meses] | [ <Y> dias]

           Veja nos exemplos abaixo como os valores devem substituir as strings no formato acima
           representadas por <>. O caractere pipe (|) entre os parênteses angulares < > significa ou, por
           exemplo: <sim | não> resulta em sim ou não.

           Na renda variável o prazo deve ser exibido em dias se o período de investimento for inferior a um
           ano (até 252 dias úteis), caso contrário, em anos.

           Os colchetes indicam uso opcional e que será usado no caso em que o período de investimento não
           representar um ano completo, por exemplo, 5 anos e 6 meses.

           Exemplos:
                        Real FIA | BBB+  | Renda Variável | FGC: não | Prazo: 3 anos e 4 meses
                        CS Saúde Digital | AA+ | Alternativos | FGC: não | Prazo: 5 anos
                        Trend Bolsa Americana | A+  | Internacional | FGC: não | Prazo: 180 dias
        */
        virtual string toSring();

        /* Retorna o prazo em dias se o período de investimento for inferior a um ano (252 dias úteis), caso
           contrário, em anos.
        */
        virtual short getPrazo();

        /* Calcula o valor do Imposto de Renda (IR) sobre o rendimento obtido no investimento.
          Investimentos de renda variável possui uma única alíquota de IR de 15%, definida na classe Imposto
          como a constante TAXA_IR_15.
```

```
        */
        virtual float calcularIR();
};
#endif // !RENDA_VARIAVEL_H
```

As funcionalidades abaixo devem ser criadas em uma classe chamada CalculoInvestimentos.

1. Obter investimentos

Lê um arquivo texto fornecido pelo usuário com o nome Investimentos.txt e que possui os seguintes dados: tipo, estratégia e nome do investimento, *rating*, proteção do FGC, valor investido, taxa do investimento ao ano (a.a.), data em que o dinheiro foi investido e a data em que ele pode ser resgatado do investimento. O arquivo possui um formato em que cada linha de texto possui os dados de um investimento separados por ponto-e-vírgula (;), como se pode ver no exemplo abaixo.

> **Tipo;Estratégia;Nome;Rating;FGC;Valor Investido;Taxa;Data do Investimento;Data de Resgate**
> RF;Pós-fixado;CDB Original Jan/2026;AAA;sim;2000;10%;14/01/2021;14/01/2026
> RV; Multimercado;Western Asset FIM;A;não;4000;58%;15/01/2021;15/12/2021

Os dados de cada investimento devem ser extraídos do arquivo e armazenados em um objeto da classe RendaFixa ou RendaVariavel. O tipo do objeto a ser criado deve ser definido de acordo com o tipo de investimento obtido em cada linha do texto (RF = Renda Fixa ou RV = Renda Variável).

Esses objetos devem ser armazenados em um vector de referências para objetos da classe Investimento.

2. Relatório de Investimentos

Gera um relatório para cada tipo de investimento obtido do vector. Os dados a serem exibidos devem ser agrupados pelo tipo de investimento, exibindo primeiro os investimentos de renda fixa e depois os de renda variável. O relatório deve ser gravado em um arquivo texto chamado Investimentos.rel no diretório atual do projeto. Os valores a serem exibidos no relatório para cada investimento são:

1. nome;
2. *rating*;
3. estratégia de investimento;
4. proteção FGC;
5. prazo de acordo com o tipo de investimento (ver comentários nas classes acima);
6. valor investido;
7. taxa do investimento ao ano (a.a.);
8. taxa do investimento ao mês (a.m.);
9. data do investimento;
10. data de resgate;
11. valor bruto acumulado do investimento (sem desconto de IR);
12. valor líquido acumulado do investimento (com desconto de IR);
13. alíquota de IR (Imposto de Renda);
14. valor do IR em reais a ser pago sobre o rendimento;
15. rendimento bruto (sem desconto de IR);
16. rendimento líquido (com desconto de IR).

O rendimento em reais do investimento corresponde aos juros recebidos de acordo com a taxa de

remuneração do investimento. Para calcular o valor bruto acumulado de um investimento use a fórmula abaixo.

$$\mathbf{VF = VP \times (1 + t)^{p}}$$

Onde:

VF = valor futuro (valor bruto acumulado)
VP = valor presente (valor investido)
t = é a taxa de remuneração do investimento
p = é o prazo (período) do investimento

☞ **Atenção:** 1. A taxa e o prazo devem usar a mesma unidade, ou seja, ambos em dias, meses ou anos.

2. O cálculo do valor bruto acumulado considera apenas os dias úteis, portanto assuma que um ano possui 252 dias úteis e um mês 21 dias úteis.

3. Para se calcular o número de dias úteis entre duas datas desconsidere qualquer feriado nos dias úteis da semana (segunda a sexta).

Para obter uma taxa equivalente, por exemplo, 12% ao ano correspondem a que taxa mensal? Use a fórmula abaixo para obter a taxa mensal.

$$taxaequivalente = (1 + taxaconhecida)^{\frac{prazoquequero}{prazoquetenho}} - 1$$

Usando a fórmula acima, $((1 + 12\%)^{(1 / 12)} - 1)$, temos a taxa equivalente de 0,94% ao mês, que corresponde aos 12% ao ano. Veja que o prazoquequero corresponde a um mês (1) e o prazoquetenho a um ano (12 meses).

Use o leiaute abaixo para gerar o relatório a ser gravado no arquivo.

Observe que a *string* que representa o nome, o *rating* , a estratégia e indica se há ou não proteção FGC para o investimento, é exatamente o valor obtido pelo método toString das classes RendaFixa e RendaVariavel.

Relatório de Investimentos

1. Renda Fixa

- CDB PAN Jan/2026 | AAA | Pós-fixado | FGC: sim | Prazo: 3 anos

Valor investido:	R$ 4 .500,00
Taxa ao ano:	11,00% a.a.
Taxa ao mês:	0,87% a.m.
Data do investimento:	14/04/2018
Data de resgate:	14/04/2021
Valor bruto:	R$ 6.154,34
Valor líquido:	R$ 5.906,19
Alíquota de IR:	15,00%
Valor do IR:	R$ 248,15
Rendimento bruto:	R$ 1.654,34
Rendimento líquido:	R$ 1.406,19

2. Renda Variável

- Western Asset FIM | A | Multimercado | FGC: não | Prazo: 231 dias

Valor investido:	R$ 4.000,00
Taxa ao ano:	58,00% a.a.
Taxa ao mês:	3,89% a.m.
Data do investimento:	15/01/2017
Data de resgate:	15/12/2017
Valor bruto:	6.083,62
Valor líquido:	5.771,08
Alíquota de IR:	15,00%
Valor do IR:	R$ 312,54
Rendimento bruto:	R$ 2.083,62
Rendimento líquido:	R$ 1.771,08

3. Relatório de Investimentos por estratégia

Exibe os dados abaixo organizando os investimentos pelo tipo de estratégia utilizado.

1. estratégia e nome do investimento;
2. valor investido;
3. valor bruto acumulado do investimento (sem desconto de IR);
4. valor líquido acumulado do investimento (com desconto de IR);
5. rendimento bruto (sem desconto de IR);
6. rentabilidade bruta percentual do investimento segundo a fórmula abaixo.

A rentabilidade expressa a valorização (ou desvalorização) de um determinado investimento em termos percentuais. Desta forma, um investimento de R$ 10 que, após um mês vale R$ 11, registrou uma rentabilidade de 10%. A fórmula de cálculo da rentabilidade é a seguinte:

Rentabilidade = ((Valor Bruto Acumulado / Valor Investido) - 1) * 100

Esse relatório deve exibir os dados usando o leiaute abaixo. Observe que todos os investimentos que possuem a mesma estratégia devem ser agrupados e exibidos na sua categoria de estratégia.

1. Estratégia: Pós-fixado

- CDB PAN Jan/2026

Valor investido	Valor bruto	Valor líquido	Rendimento bruto	Rentabilidade
R$ 4 .500,00	R$ 6.154,34	R$ 5.906,19	R$ 1.654,34	36,76%

2. Estratégia: Multimercado

- Western Asset FIM

Valor investido	Valor bruto	Valor líquido	Rendimento bruto	Rentabilidade
R$ 4 .000,00	R$ 6.083,62	R$ 5.771,08	R$ 2.083,62	52,09%

☞ **Atenção:** Observe que a classe Investimento é uma classe abstrata, portanto o programa deve sempre usar uma referência dessa classe para processar polimorficamente os métodos das subclasses de Investimento.

- Critérios de avaliação

1. O trabalho será avaliado considerando:

   a. A validação dos dados fornecidos pelo usuário.
   b. A lógica empregada na solução do problema.
   c. O funcionamento do programa.
   d. O conhecimento da linguagem de programação C++.
   e. A implementação dos conceitos de orientação a objetos.
   f. O uso do princípio do menor privilégio[2].
   g. Código fonte sem erros e sem advertências do compilador.
   h. Código fonte legível, indentado, organizado e comentado.
   i. Identificadores significativos para aprimorar a inteligibilidade do código fonte.

2. O programa deve ser desenvolvido integralmente usando apenas os recursos da linguagem C++ e do *Microsoft Visual Studio Community* 2019, versão 16.8. Programas desenvolvidos em outras linguagens, mesmo que parcialmente, receberão nota zero.

3. Para que o programa seja avaliado o código deve executar com sucesso. Programas que apresentarem erros de compilação e/ou ligação receberão nota zero.

4. Trabalhos com plágio, ou seja, programas com código fonte copiados de outra pessoa (cópia integral ou parcial) receberão nota zero.

5. O desenvolvimento do trabalho é individual.

6. Incluir em cada classe apenas as definições de tipos de dados, variáveis, constantes e métodos que forem essenciais para a funcionalidade da mesma.

7. Escrever funções e métodos específicos, ou seja, com atribuição clara e objetiva.

   Exemplo: Pesquisa um nome em um vetor de *strings*. Retorna a posição do nome no vetor se ele for encontrado ou -1 caso contrário.

```
int pesquisarNome(const string vetor[], string nome);
```

   A descrição dessa função deixa claro que não é atribuição dela ler o nome via algum dispositivo de E/S e nem exibir o resultado da consulta, somente realizar a pesquisa do nome no vetor e devolver o resultado.

   O uso do *const* no protótipo de função acima é um exemplo do princípio do menor privilégio, porque se a função não precisa alterar os argumentos usados na sua chamada, ela não pode ter parâmetros formais com esse poder ou privilégio. O uso do

[2] O **princípio do menor privilégio** declara que deve ser concedido ao código somente a quantidade de privilégio e acesso de que ele precisa para realizar sua tarefa designada, não mais que isso.

const assegura que apesar da função receber um vetor, que sempre é passado por referência, ela não poderá modificá-lo.

8. Não escrever código redundante.

9. É proibido modificar os nomes de arquivos, identificadores, os protótipos de função, as declarações e/ou definições de métodos e classes fornecidos neste texto ou em anexo.

10. É permitido acrescentar novas declarações e/ou definições de classes, métodos, variáveis e constantes desde que estejam de acordo com os critérios acima.

11. Use a classe ArquivoTexto fornecida no projeto ArquivoTexto.7z para manipular os arquivos de texto.

- Instruções para entrega do trabalho

1. Crie uma solução com o nome Investimentos e um projeto com o seu nome e sobrenome, por exemplo: AyrtonSenna.

2. Limpe a solução para apagar todos os arquivos OBJ da pasta *Debug* do projeto. Confira se esses arquivos realmente foram excluídos da pasta *Debug*. De preferência exclua todo o conteúdo dessa pasta.

3. Antes de submeter os exercícios via SIGAA, compacte apenas o diretório do projeto para criar um arquivo 7z com o seu nome e sobrenome, por exemplo: AyrtonSenna.7z.

   Não inclua no arquivo 7z o diretório da solução Investimentos, somente o diretório do projeto que possui o seu nome e sobrenome.

   Assim o tamanho final do arquivo 7z não ultrapassará o limite de 10 MB do SIGAA, porque o conteúdo do diretório oculto .vs, criado dentro da pasta da solução, não será compactado, reduzindo drasticamente o tamanho final do arquivo.

   Para compactar o projeto use o *software* livre de código aberto 7-Zip, que está disponível em https://www.7-zip.org/download.html.

- Data de entrega

Quinta-feira, 1º de abril de 2021.

- Valor do trabalho

10,0 pontos.

Prof**. M**árlon **O**liveira **d**a **S**ilva
marlon.silva@ifsudestemg.edu.br