	Avaliação Escrita Distância			
	Elaborada por	Felipe Marques Martins		
	Data / Local	23/09/2019 / AytyTech		
	Projeto	Prova Escrita .NET		
	Nr	0001	**Versão**	1.0

Objetivo

1. Possibilitar que o(a) candidato(a) à vaga em questão consiga mostrar seus conhecimentos técnicos.

2. Possibilitar avaliação mais precisa do conhecimento do(a) candidato(a) à vaga em questão.

Atividades

1) Considere o trecho de código a seguir:

```
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
   public class Funcionario {}
   public class Professor : Funcionario {}

   public partial class Form1 : Form
   {
      public Form1()
      {
         InitializeComponent();

         comportamento(new Funcionario());
         comportamento(new Professor());
      }

      public void comportamento(Funcionario colaborador)
      {
            if (typeof(Funcionario) == colaborador.GetType())
            {
                  MessageBox.Show(" Funcionário typeof ");
            }

            if (colaborador is Funcionario)
            {
                  MessageBox.Show(" Funcionário is ");
            }

            if (typeof(Professor) == colaborador.GetType())
            {
                  MessageBox.Show(" Professor typeof ");
            }

            if (colaborador is Professor)
            {
                  MessageBox.Show(" Professor is ");
            }
      }
   }
}
```

Existem hoje essas duas formas de comparar objetos C# (*typeof* e *is*). Considerando isso, responda:
Quais mensagens serão apresentadas?
Existe alguma diferença entre essas duas formas de comparar os objetos?
Em caso afirmativo, explique a diferença.

Considere o código abaixo para responder as duas próximas questões:

```
public class HRIntegratorFactory
{
    public static object GetHRIntegrator()
    {
        object obj = null;

        if (AppManagerLocalSingleton.Instance.User.Role.IdRole == AytyFramework.Entity.Core.RoleConstant.IdRoleOperador
            &&
            AppManagerLocalSingleton.Instance.Module.IdModule == ModuleConstant.Attend)
        {

            int idIntegrator = 0;

            if (ConfigurationManager.AppSettings["IdHRIntegrator"] != null)
            {
                idIntegrator = ConfigurationManager.AppSettings["IdHRIntegrator"].ToInt();
            }
            else
            {
                idIntegrator = AppManagerLocalSingleton.Instance.Project.ProjectCRM.IdHRIntegrator;
            }

            switch (idIntegrator)
            {
                case HRIntegratorBaseConstant.IdNone: //"0": => None
                    obj = new HRIntegratorNone();
                    break;

                case HRIntegratorBaseConstant.IdAyty: //"1": => Ayty
                    obj = new HRIntegratorAyty();
                    break;

                case HRIntegratorBaseConstant.IdExternal: //"2": => External
                    obj = new HRIntegratorExternal();
                    break;
                default:
                    throw new Exception("GetHRIntegrator " + idIntegrator + " not found!");
            }

            return obj;
        }
        else
        {
            return new HRIntegratorNone();
        }
    }
}
```

2) Pensando em Orientação ao Objeto, analisando o código apresentado, podemos concluir que HRIntegratorNone, HRIntegratorAyty e HRIntegratorExternal possuem algo em comum? O quê?

3) Em poucas palavras, explique o funcionamento da função GetHRIntegrator da classe HRIntegratorFactory.

4) O código abaixo possui um erro que não permite sua compilação.

```
if (AppManagerLocalSingleton.Instance.DataTableProject.Columns.Contains("NU_MAX_VEHICLE_AGE")
&& !string.IsNullOrEmpty(((CustomerItemItauCard)this.customerItem).NuVehicleYear)
&& ((CustomerItemItauCard)this.customerItem).NuVehicleYear.Length == 4
&& System.DateTime.Now.ToString("yyyy").ToInt() - ((CustomerItemItauCard)this.customerItem).NuVehicleYear.ToInt() >
AppManagerLocalSingleton.Instance.DataTableProject.Rows["NU_MAX_VEHICLE_AGE"].ToInt())
{
        this.returnTransactionByValidatorCustomDTO.AddFailedResult("Ano veículo menor que o permitido!",
"txtNuVehicleYear");
}
```

Explique qual o problema e como podemos corrigir.

5) Considere uma linguagem de programação estruturada hipotética com as seguintes características:

a passagem de parâmetros se dá exclusivamente por valor;
o símbolo "=" representa o comando de atribuição que atribui um valor a uma variável;
"print" é uma função pré-definida que mostra na tela o valor de uma variável;
"p" é um procedimento definido pelo programador; e
não existe o conceito de variável global.

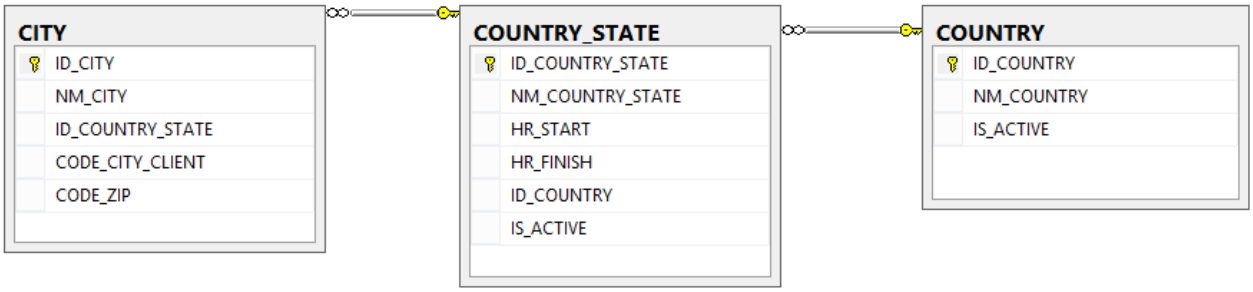
Considere agora a execução das três linhas a seguir:

```
v1 = 50;
p(v1 + 10);
print (v1);
```

O que pode ser afirmado em relação ao valor que será mostrado na tela?

a) Será necessariamente 10.
b) Será necessariamente 50.
c) Será necessariamente 60.
d) Dependerá do nome dado ao argumento na definição de "p".
e) Dependerá do algoritmo implementado no procedimento "p".

6) Hoje no ambiente do cliente, temos a seguinte estrutura das cidades cadastradas no sistema:



Escreva os scripts necessários para realizar os seguintes ajustes:

- Acrescentar a coluna IS_ACTIVE do tipo boolean na tabela CITY

- Remover as colunas HR_START, HR_FINISH da tabela COUNTRY_STATE

- Inserir o registro 'Não Informado' na tabela COUNTRY (considerar como Auto Increment)

- Atualizar o nome das cidades de forma que as cidades dos estados 'PA' e 'PE' fiquem com todas as letras maiúsculas

- Deletar todos os registros da tabela CITY que estiverem com ID_COUNTRY_STATE com valor nulo

-Criar uma view com nome VW_CITY_DETAILS para que retorne em uma mesma consulta o nome da Cidade, Estado e País (conforme figura ao lado)

Results | Messages

	NM_CITY	NM_COUNTRY_STATE	NM_COUNTRY
1	Abaetetuba	PA	BRASIL
2	ADAMANTINA	SP	BRASIL
3	ADOLFO	SP	BRASIL
4	Água Preta	PE	BRASIL
5	AGUDOS	SP	BRASIL
6	AIMORES	MG	BRASIL
7	ALMIRANTE TAMANDARE	PR	BRASIL
8	ALTINOPOLIS	SP	BRASIL
9	AMERICANA	SP	BRASIL