

# Prova de C# Básico

Prof: Leonardo Trevisan Silio 😊

Sumário (sim tem um sumário)

## Sumário

Introdução.....	2
Questão 1: Básico – Estagiário.....	4
Questão 2: Básico – Aprendiz.....	5
Questão 3: Básico – Meio Oficial.....	7
Questão 4: Básico – Engenheiro.....	8
Questão 5: Orientação a Objetos – Estagiário.....	10
Questão 6: Orientação a Objetos – Aprendiz.....	11
Questão 7: Orientação a Objetos – Meio Oficial.....	12
Questão 8: Orientação a Objetos – Engenheiro.....	14
Questão 9: LINQ – Estagiário.....	16
Questão 10/12: LINQ – Aprendiz/Meio Oficial/Engenheiro.....	17

# Introdução

Sejam bem-vindos a prova de C# básico. Nela você encontrará os tópicos:

- 4 Questões de C# básico (variáveis, operações e sintaxe para classes)
- 4 Questões de Orientação a Objetos (abstração, encapsulamento, herança e polimorfismo)
- 4 Questões de LINQ

Além disso as questões são divididas em níveis de dificuldade:

- Estagiário
- Aprendiz
- Meio Oficial
- Engenheiro

Ou seja, cada tópico tem uma questão do nível fácil para o impossível. Você deve responder ao menos uma questão de cada tópico, mas pode tentar responder todas. Você vai receber a nota da questão mais difícil que responder.

- Questões do nível Estagiário: Valem no máximo 2 pontos, pois não lhe dão chances de avaliar todas as competências.
- Questões do nível Aprendiz: Valem no máximo 3 pontos, pois avaliam quase todas as competências, mas não profundamente.
- Questões do nível Meio Oficial: Valem no máximo 4 pontos, pois avaliam todas as competências integralmente.
- Questões do nível Engenheiro: Valem no máximo 5 pontos, pois avaliam todas as competências e com bônus de dificuldade.

Por exemplo, acertar completamente uma questão de Orientação a Objetos nos níveis Aprendiz e Meio Oficial lhe garante 4 pontos, pois a questão de Meio Oficial vale mais.

Outro exemplo: Acertar 20% das competências na questão de Engenheiro (20% de 5 = 1) e 75% das competências de Meio Oficial (75% de 4 = 3) em questões de LINQ lhe garantirá 3 pontos.

A sua nota final é a soma da nota de cada tópico (Básico, OO e LINQ) menos 1. Ou seja, a prova vale 14, e para tirar a nota máxima 10 você só precisa acertar uma questão de nível Aprendiz e duas Meio Oficial. Ou uma Estagiário e duas Engenheiro. De qualquer modo, existem várias combinações para de gabaritar a prova.

Em várias questões é possível que você encontre mais de um arquivo no projeto base entregue.

- Os arquivos *Program.cs* quase sempre devem ser editados por você.
- Os arquivos *Model.cs* são usados para classes e Orientação a Objetos em geral. Sempre devem ser alterados por você.
- Os arquivos *App.cs* são arquivos internos e nunca devem ser alterados por você.

Dica: Use Esc em algumas aplicações para sair.

Dica Extra: De uma olhada nas questões nível Aprendiz, escolha a mais difícil, faça as outras e Aprendiz e troque a Aprendiz mais difícil pela nível Estagiário do mesmo tópico. Após concluir isso busque tentar resolver ou a Aprendiz mais difícil ou a Meio Oficial mais fácil. É uma boa estratégia para resolução desta prova.

“Salve-se quem puder e boa prova”

## Questão 1: Básico – Estagiário

Os estagiários da Bosch fizeram uma competição de ginástica olímpica dentro de um forno industrial e deseja-se escrever um programa que, a partir das notas recebidas, decide a nota final. Para isso você deve implementar a função *mediaEspecial* que recebe um vetor de notas e retorna a média dos três valores intermediários, ou seja, jogando fora as melhores e piores notas.

Exemplo:

Se recebida as notas 1, 2, 3, 4 e 4, jogamos fora a menor nota 1 e a maior nota 4, ficando apenas com as notas intermediárias 2, 3 e 4, assim a média seria 3.

Não é necessário tratar o caso de quantidade de notas pares onde não sabemos qual nota jogar fora, no caso de 4, 5, 6 e 7. Neste caso não saberíamos se jogamos fora o 4 ou o 7.

O vetor de exemplo possui 9 valores, mas você não pode supor que a entrada sempre terá 9 valores.

O algoritmo não precisa ser rápido ou eficiente.

## Questão 2: Básico – Aprendiz

A equipe de Marketing da Bosch pediu a ETS que desenvolvesse uma aplicação com alguma ferramentas de edição de imagens e você ficou encarregado de parte da implementação de um sistema de *Blur*.

Para isso, na classe *BlurControl* você deve implementar a função *ApplyBlur* que recebe um vetor de valores da imagem. Você deve alterar o vetor para que ele afete a imagem final.

Um Meio Oficial lhe explicou o que precisa ser feito:

Cada valor do vetor recebido deve receber a média dos 20 valores anteriores e dos próximos 20 valores. Assim, contando consigo mesmo, serão 41 valores somados e divididos por 41 para obter o novo valor do vetor. Basicamente a operação:

$$data[i] = (data[i-20] + data[i-19] + \dots + data[i-1] + data[i] + data[i+1] + \dots + data[i+19] + data[i+20]) / 41$$

Para evitar estouros aritméticos, o Meio Oficial ainda disse que você não precisava se preocupar em calcular os primeiros 20 valores de *data*, nem os últimos 20. Ou seja, *data[0]*, por exemplo, fica inalterado.

Abaixo você pode ver a imagem inicial e o resultado esperado:



Seu Meio Oficial disse que com este algoritmo, você vai espalhar as cores nos cálculos das médias resultando num efeito de embaçamento.

Experimente: Experimente mudar todos os valores de *data* para 0, você vai ver que você realmente está mexendo na imagem e vai pintá-la para preto.

Dica: O vetor está em *byte* ou seja, se você fizer a soma sem cuidado você dará *overflow* na variável que só aceita valores até 255. Use uma variável temporária para o cálculo da média.

Dica Extra: É melhor você criar um vetor temporário para evitar um efeito adverso de usar um valor já calculado em um valor futuro. Isso pode melhorar o resultado.

### Questão 3: Básico – Meio Oficial

Um projeto interno da Bosch conta com um sensor que recebe uma informação que varia entre 400 e 600 N de força. Precisa-se de um valor relativamente preciso porém, existe muito ruído captado pelo sensor que é de menor qualidade para tornar o produto barato e escalável atendendo as demandas da empresa.

O Engenheiro do projeto garantiu a você que, usarmos a fórmula:

$$S = 1.57 M - 285$$

Onde **M** é a média dos últimos 40 valores captados teremos **S** que é a real saída do sensor, sem ruído.

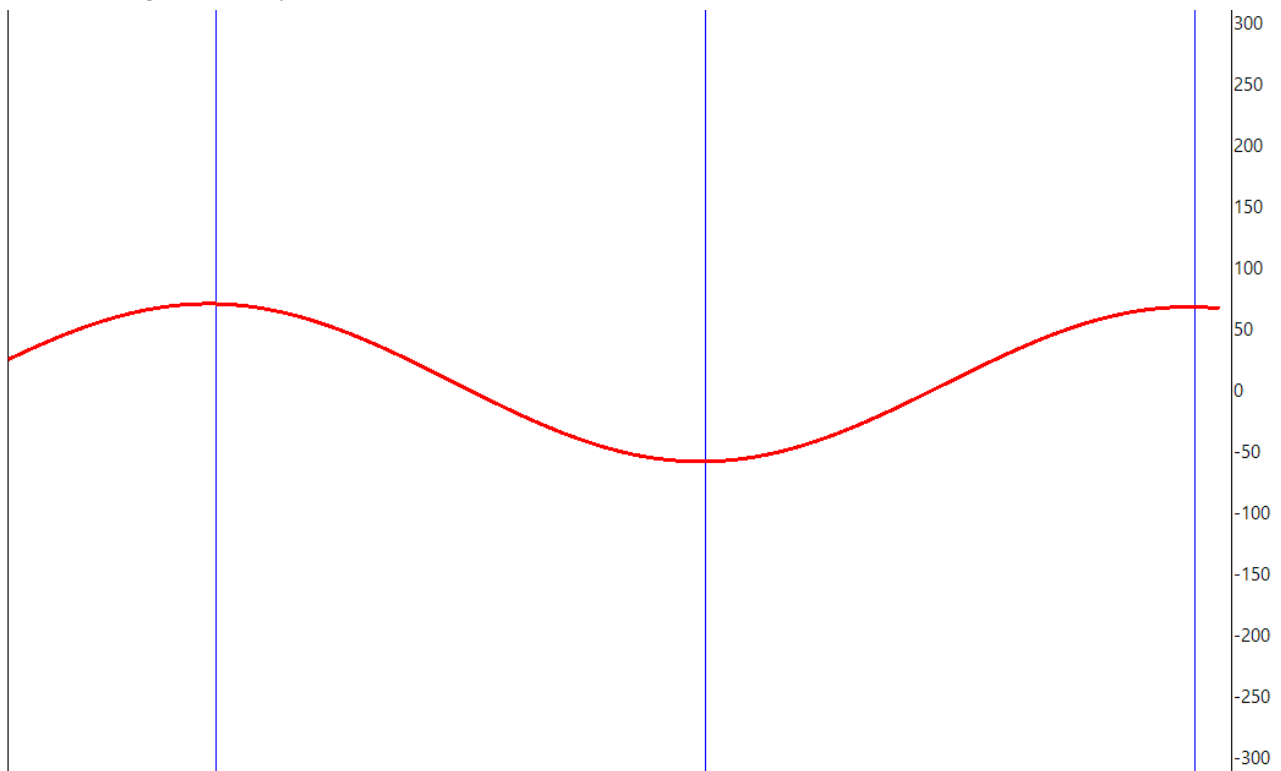
Implemente a função *Control* na classe *Controller* que será chamada várias vezes ao longo do tempo. O parâmetro da função recebe o último valor lido pelo sensor e deve retornar o real valor lido pelo sensor sem o ruído.

Use a barra de espaço para habilitar e desabilitar o controlador. Inicialmente o sensor retorna o valor recebido, ou seja, não faz nada mostrando na tela o valor com ruído.

## Questão 4: Básico – Engenheiro

A Bosch está buscando montar uma estação de rádio para trabalhar com uma comunicação que não precisa de internet ou conexão em geral. Para isso ela emite de uma estação uma onda que pode ser captada por aparelhos semelhantes a rádios.

Contudo, várias transmissões em várias frequências são captadas sobre a planta confundindo o sinal, assim ele parece ser apenas ruído. O sinal que a Bosch emite é uma onda de frequência 0.5 Hz, ou seja, ela leva 2 segundos para subir e descer:

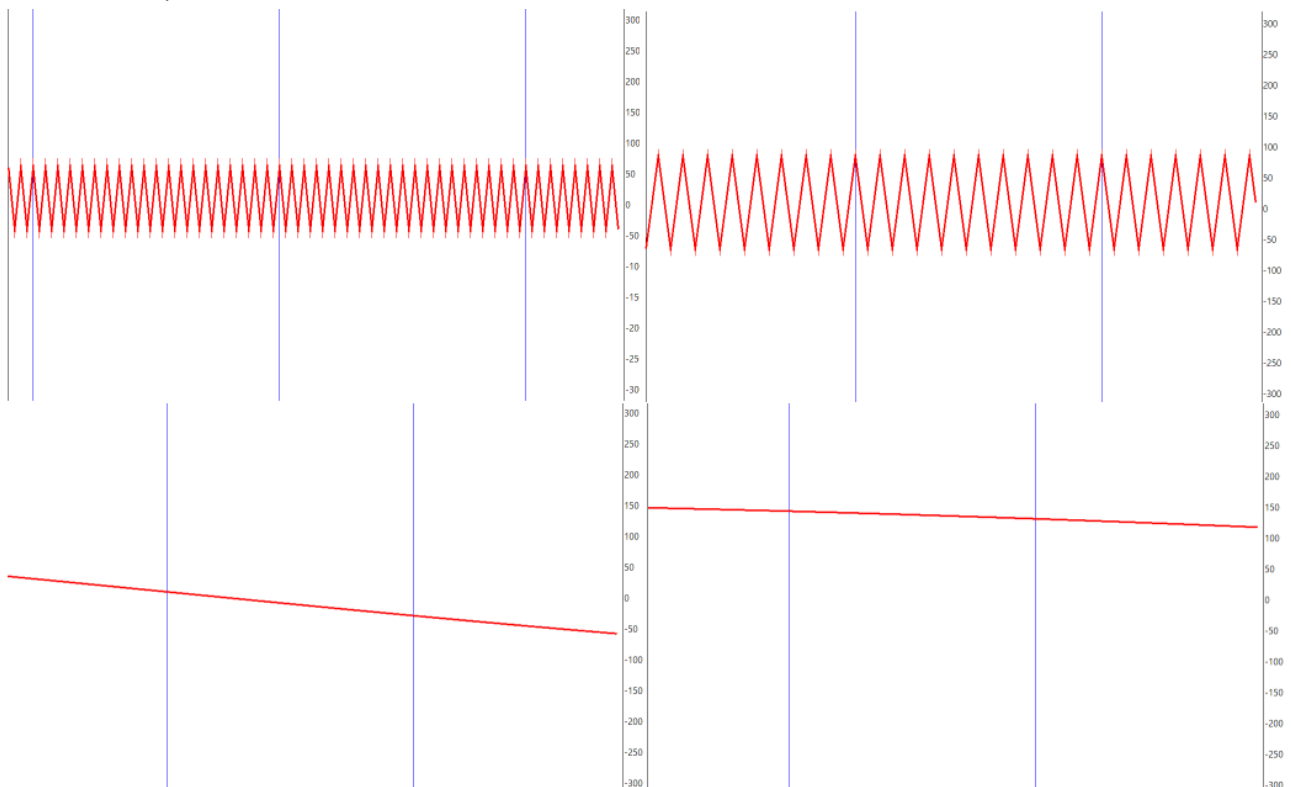


A cada 2 segundos (duas linhas azuis) o valor de vermelho desce até -50 e sobe até 50. Porém existem outros 4 sinais que estão atrapalhando a captação correta deste sinal 2 de alta frequência (que sobem e descem muitas vezes por segundo) e 2 de baixa frequência (que sobem e descem lentamente parecendo uma linha reta). Esses 4 sinais podem ser vistos na próxima imagem.

Você falou com um Engenheiro e ele te indicou a solução: Filtros passa baixa. Um filtro é basicamente o ato de tirarmos uma média de valores que entram em um sistema e colocar esse valor na saída. Quanto mais valores pegamos para tirarmos uma média, mais alto a frequência que filtraremos. Seu amigo Engenheiro disse, você pode obter todas remover as frequências mais altas simplesmente tirando a média dos últimos 20 valores recebidos.



Além disso, você pode remover as frequências mais altas junto da frequência enviada pela Bosch calculando a média dos últimos 40 valores recebidos.



Implemente a função *Control* na classe *Controller* que será chamada várias vezes ao longo do tempo. O parâmetro da função recebe o último valor lido pelo rádio com todos as ondas recebidas e deve retornar o sinal filtrado apenas com a onda da Bosch.

Use a barra de espaço para habilitar e desabilitar o controlador.

As linhas azuis verticais indicam a passagem de 1 segundo.

Curiosidade: Os rádios reais funcionam exatamente com esses princípios.

## Questão 5: Orientação a Objetos – Estagiário

Um ex-estagiário que foi promovido a cliente após entrar em um forno industrial por motivos olímpicos estava produzindo um sistema que permitir salvar e ler clientes e produtos, contudo, devido a sua surpreendente falta de produtividade ele saiu da empresa deixando a aplicação inacabada.

Por sorte você está aqui.

Implemente toda a estrutura da forma que você achar adequado apenas onde você encontrar comentários *TODO*(*To do* ou a fazer), é necessário concluir aplicação Orientada a Objetos já começada.

## Questão 6: Orientação a Objetos – Aprendiz

Sua equipe está desenvolvendo um jogo plataforma e lhe pediram para implementar os inimigos do jogo. Neste jogo o jogador controla um bloco azul que deve desviar de inimigos laranjas que caem do céu.

Você deve implementar inimigos no arquivo *Model.cs* que herdam da classe *Enemy*. Você deve iniciar os parâmetros iniciais dos inimigos na função *Build*. Para isso você pode utilizar uma função *build*, esta função cria um retângulo na tela que faz parte do inimigo (já que alguns inimigos podem ter mais que um único retângulo que faz parte de si). Você pode criar um corpo deslocado do centro (*Line, Column*) para fazer inimigos diferentes.

As propriedades *Line* e *Column* indicam a linha e coluna, ou seja, a posição do inimigo. Você pode indicar a progressão dele na função *Move* que é executada a cada *frame* (40 vezes por segundo) do jogo.

Por sorte, seu colega de trabalho já implementou 2 inimigos para que você use como exemplo em seu desenvolvimento.

Você também pode usar a função *random(max)* para obter um número entre 0 e *max*.

Você deve implementar os seguintes inimigos:

1. Um inimigo que não se mexe e é criado na posição 500, 500 com tamanhos 30 por 30.
2. Um palito que cai, semelhante a um dos exemplos, porém, com uma altura maior que 300.
3. Uma plataforma que cobre a tela toda e cai aos poucos, porém possui um buraco por onde o jogador pode passar.
4. Um inimigo que se move de forma aleatória a cada movimento.
5. Um inimigo que teleporta a cada 2 segundos.
6. Um inimigo que cai extremamente rápido.
7. Um inimigo que se move horizontalmente.

Para que cada uma destas implementações funcione você deve adicionar no *Program.cs* para que a execução aconteça.

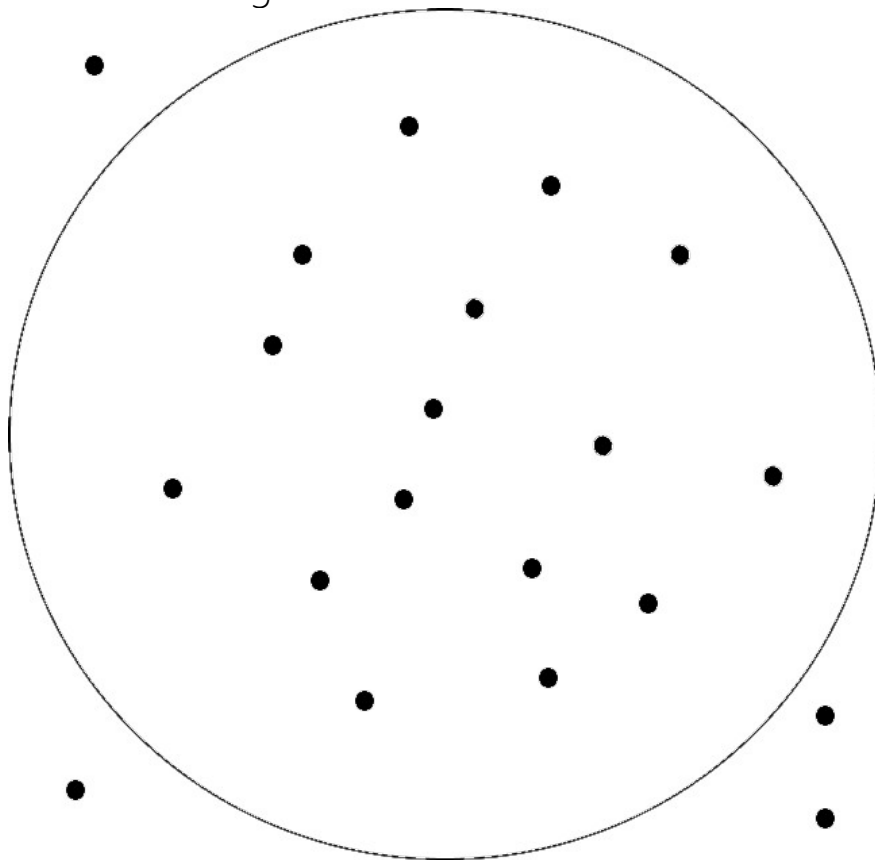
Dica: Se está confuso com a abstração, experimente criar objetos com valores aleatórios e espere ele aparecer para ver como ficou.

Dica Extra: Se você remover do *Program.cs* os outros inimigos, o inimigo que você fizer vai aparecer mais rápido, já que as aparições são aleatórias.

## Questão 7: Orientação a Objetos – Meio Oficial

Você deseja calcular o valor da constante  $\pi$  de uma maneira divertida. Você conversa com um colega no setor de engenharia e ele te recomenda tentar o Método de Monte Carlo que consiste em usar métodos probabilísticos para aproximar valores.

A ideia é simples: Para calcular a área de Figuras geométricas basta sortear pontos e jogar em um retângulo 100 x 100 qualquer onde a figura está dentro. Ao jogarmos 10 mil pontos podemos usar conceitos estatísticos para apontar que a área da figura é aproximadamente igual a quantidade pontos caíram dentro da figura.



Para botar isso em prática vocês começam a implementar a classe ponto e a classe Figura que possui um método *EstaDentro* que testa se um ponto está dentro da figura. E a função *sortearPonto*. Complete as implementações e sorteie 10 mil pontos entre as posições (0, 0) e (100, 100) no plano cartesiano. Depois use o *EstaDentro* para descobrir quantos pontos estão dentro de cada figura. Assim faça os seguintes testes:

- Implemente um Retângulo 50 x 50 que sobre apenas 1 quarto da tela (ele deve receber 2500 pontos aproximadamente).

- Implemente um Triângulo que cobre toda a base da tela e se estende ao topo (ele deve receber 5000 pontos aproximadamente).
- Implemente um Circulo com centro em (50, 50) e 50 de raio (ele deve receber  $\pi r^2 = \pi 2500$ , ou seja, basta pegar os pontos, dividir por 2500 e teremos uma aproximação de  $\pi$ , que sabemos que deve ser algo como 3.14).

Abaixo você pode ver as figuras:



Dicas: Para saber se um ponto está dentro da primeira figura, basta ver se a posição x e y dela são menores que 50.

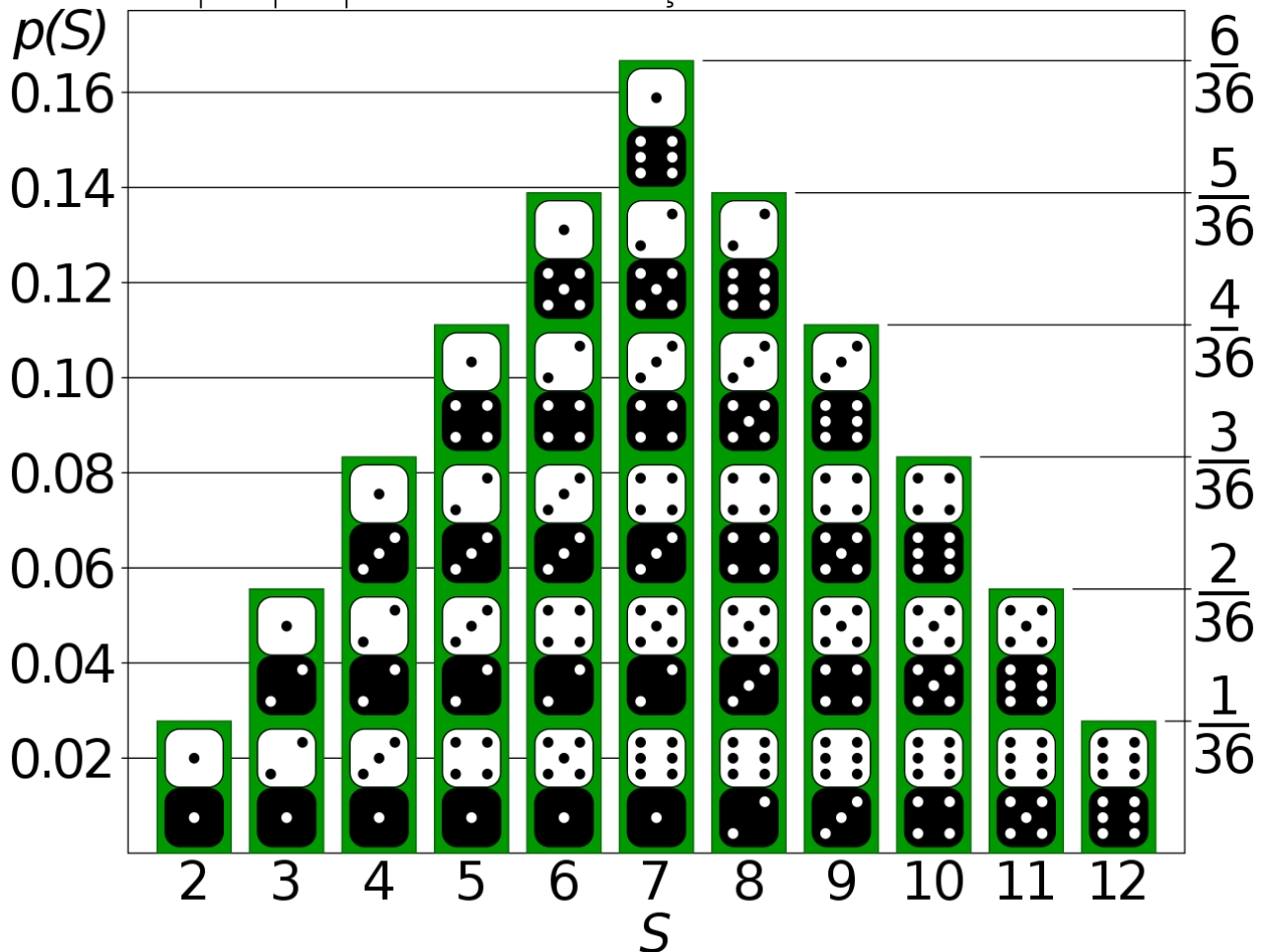
Para saber se um ponto está dentro da segunda figura, basta que  $x + y$  seja menor que 100.

Para saber se um ponto está dentro da terceira figura, basta que a soma das distâncias das coordenadas ao quadrado do centro (procure por Pitágoras) seja menor que o raio, no caso 50.

Printe os três resultados (contagem de pontos que estão dentro das três figuras) na tela e por fim, aponte o valor aproximado de  $\pi$ .

## Questão 8: Orientação a Objetos – Engenheiro

Uma variável aleatória é uma variável que muda seu valor de forma aleatória tentando atender uma distribuição. A imagem abaixo mostra a distribuição de probabilidade de se obter algum valor na soma de dois dados. Você pode observa que existe uma probabilidade muito maior de se obter 7 do que qualquer outra combinação.



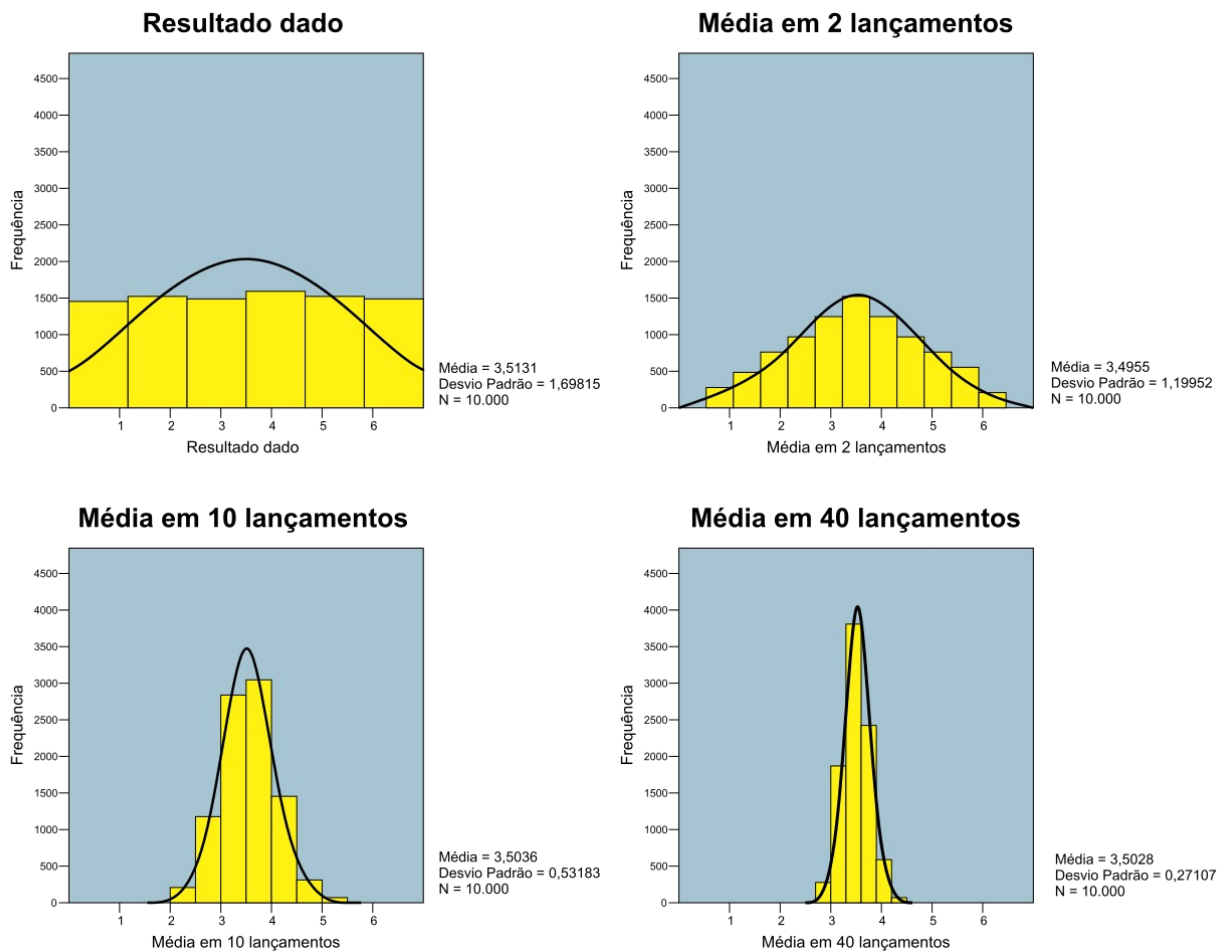
Usando a abstração inicial de outro colaborador Bosch você observa que existe a classe de variável aleatória já implementada, ou, parcialmente. Ainda resta implementar operações de soma de variáveis aleatórias, subtração, divisão e multiplicação.

Já existem duas variáveis implementadas: O lançamento de um dado de 6 lados e a probabilidade de morrer ao longo da vida (10% na infância 20% na juventude, 30% na vida adulta e 40% para os idosos).

E mais importante que isso, a função *sample* ou amostrar. Essa função retorna uma nova variável aleatória que equivale a média de uma amostra da variável, ou seja, obter o valor da variável N vezes e tirar a média. Note que existe uma conversão direta de *RandomVariable* para *float*.

Após isso você deve tentar responder a pergunta, é verdade que uma variável aleatória de amostrar converge para uma Gaussiana?

A resposta é sim, este é o Teorema Central do Limite:



Apesar disto, você deve usar a função `printDistribution` para constatar isso.

Dica: Crie suas próprias classes para representar um objeto de soma, multiplicação, amostragem, etc.

## Questão 9: LINQ – Estagiário

Após 2 anos um Ex-Estagiário reconquistou seu emprego após sua demissão envolvendo um misterioso incidente com um forno industrial. Agora ele precisa implementar, junto a você, uma lista de filtro para os funcionários da Bosch para uma pesquisa rápida.

Para isso você deve considerar o seguinte funcionamento:

Ao se digitar na barra de pesquisa "x y z"(note a separação por espaço dos três textos) deve aparecer todos os usuários que possuem em seus nomes, cargos, setores ou Edv as *strings* x, y **ou** z. Ou seja se alguém possuir qualquer um destes caracteres no nome ele deve ser incluído.

Caso o pesquisador usar o carácter de subtração "-" antes da *string* isso significa que a *string* **não** pode aparecer na pesquisa.

Assim ao digitar-se na barra de pesquisa "x1 y2 -z3" serão filtrados apenas os usuários que possuírem em seus dados "x1" ou "y2" e obrigatoriamente não pode conter "z3".

A aplicação gráfica já está pronta, cabe a você implementar a função *Search* na classe *Pesquisador*.



## Questão 10/12: LINQ – Aprendiz/Meio Oficial/Engenheiro

As questões de nível Aprendiz, Meio Oficial e Engenheiro estão atreladas ao mesmo problema. O seguinte modelo foi criado:

- Existe uma universidade que possui vários departamentos.
- Cada departamento está relacionado a várias disciplinas por um identificador (*Join* pode ser útil).
- Cada departamento tem também seus professores.
- Existem turmas que são basicamente a união de uma disciplina com um professor que vai lecioná-la.
- Existem vários alunos também que possuem uma lista de matrículas, ou seja, os códigos das disciplinas que eles estão fazendo.

Você pode responder as questões de nível aprendiz implementando as funções de Pesquisa 1, 2 e 3 na classe pesquisador.

Também pode conquistar a nota de Meio Oficial implementando as questões 4 e 5.

A pesquisa 6, por sua vez, compreende a nota de nível Engenheiro.

Para associar diferentes coleções você deve utilizar-se dos identificadores.