

Doel van de oefeningen

Deze oefeningen zijn oefeningen om je programmeervaardigheden te helpen ontwikkelen. Je maakt deze oefeningen zelf op je eigen ritme. Voor alle vragen omtrent de oplossingen raadpleeg je best je docent tijdens de les, zeker als je twijfelt of je oplossing correct is (ook al werkt je programma, daarom is de werkwijze nog niet altijd correct).

Organisatie van je oefeningen

Je kan meerdere programma's schrijven in 1 visual studio Solution.

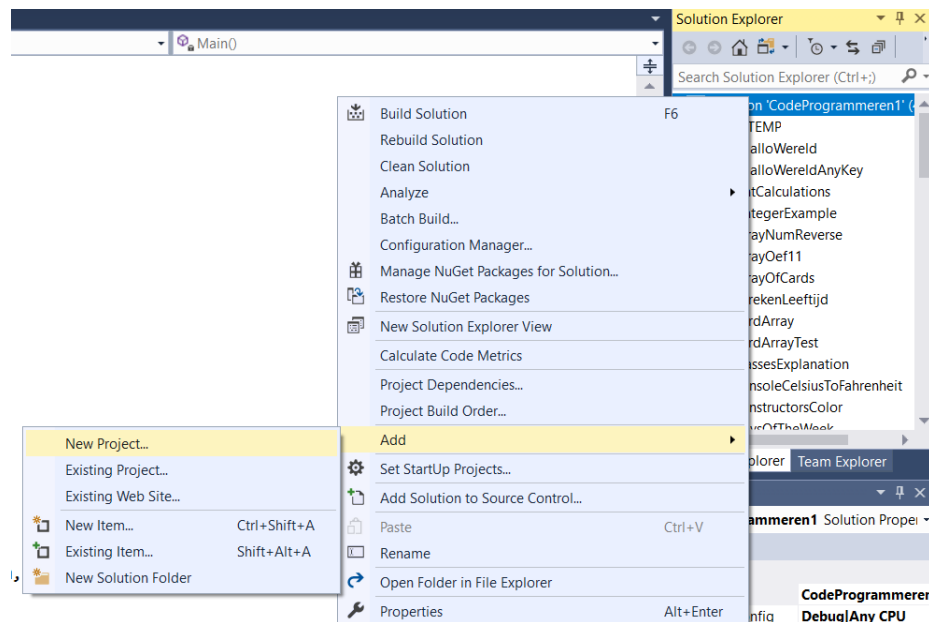
- Voor elke oefening voeg je dan aan je solution een nieuw project toe (Visual C# → windows application → console application)
- Kies een duidelijke naam bvb oefening1_1
- Om dit nieuwe project actief te maken (zorgen dat deze applicatie runt als je op debug drukt) klik je in je solution explorer met de rechter muisknop op je project en selecteer je "Set as StartUp project. (dit hoeft natuurlijk niet voor je eerste project)

Tips

- Bekijk eerst de theorie vooraleer je start, en zorg dat je alles goed begrijpt.
- Is er toch iets niet 100% duidelijk of twijfel je, vraag dan eerst uitleg aan je docent.
- Is de docent even niet beschikbaar, kan je altijd al hulp opzoeken in de msdn.
- Probeer steeds je oefeningen te maken zonder telkens naar voorbeelden te kijken, maar werk vanuit je kennis.
- Staan er fouten in je code zoek dan je foutmelding op, zoek hulp in de msdn, ga er niet van uit dat alle informatie op het internet betrouwbaar is.
- Debuggen is zeker zo belangrijk als programmeren, loopt er iets mis in je programma, is debuggen en door je code stappen de eerste stap.
- Maak de oefeningen zeker ZONDER kopiëren/plakken uit voorbeelden/msdn. Dit zorgt er enkel voor dat je elementen overbrengt, maar is geen goede leermethode. Probeer dit ten alle tijden te vermijden.
- Als je merkt dat je nog naar voorbeelden/theorie/msdn moet teruggrijpen, is het een teken dat je nog niet alles volledig onder de knie hebt, blijf oefenen (hermaak oefeningen, of variaties daarop). Ook bij het hermaken van oefeningen, kijk niet naar vorige oplossingen, maar werk vanuit je kennis.
- Wil je bijkomende oefeningen:
 - Bedenk zelf variaties op deze oefeningen, evt samen met anderen in de klas zodanig dat je ook samen de oplossing kan bekijken
 - Zoek oefeningen online op
 - Ook hierover mag je altijd vragen stellen

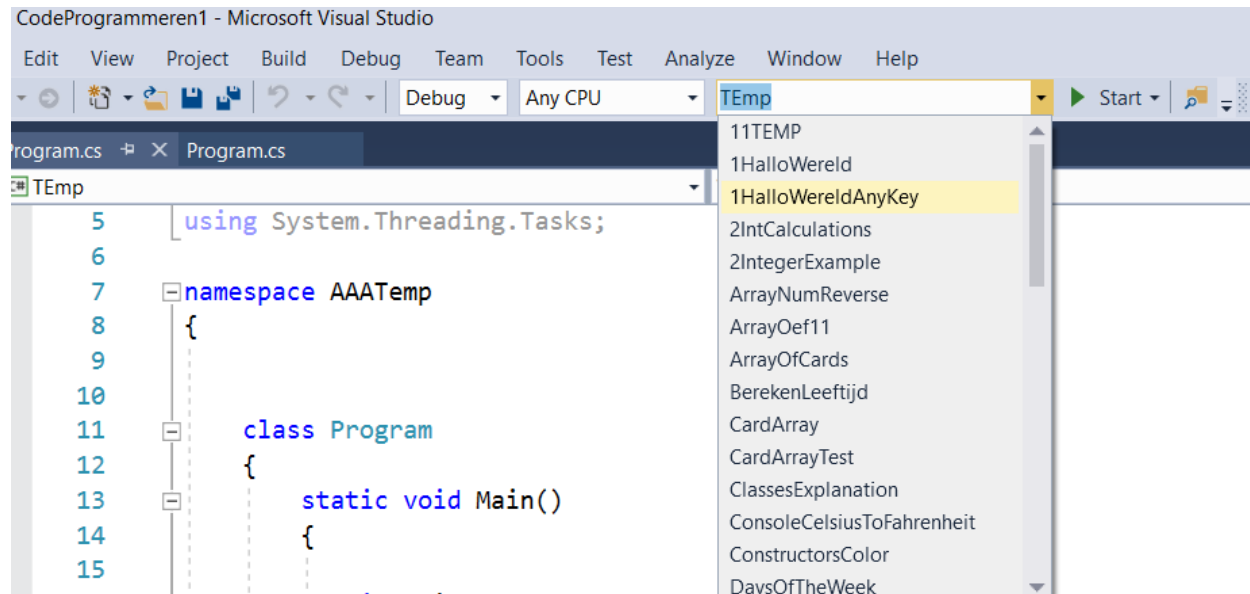
Herhaling Programming 1: introductie tot basisapplicaties

1. Maak een nieuw project aan en schrijf een programma dat “Hello World” uitschrijft, maar in 2 aparte instructies (dus eerst “Hello” en daarna “World”). Zorg ook dat je de gebruiker vraagt om met een druk op de toets het programma af te sluiten en vang dit op. Let op de spatie tussen “Hello” en “World”. Kan je verschillende manieren bedenken om die spatie erin te krijgen? Probeer uit!
2. Werk met meerdere projecten in Visual studio, binnen 1 solution. In je solution explorer:
 - a. Klik met je rechtermuis op je solution
 - b. Ga naar Add
 - c. Selecteer naar New Project

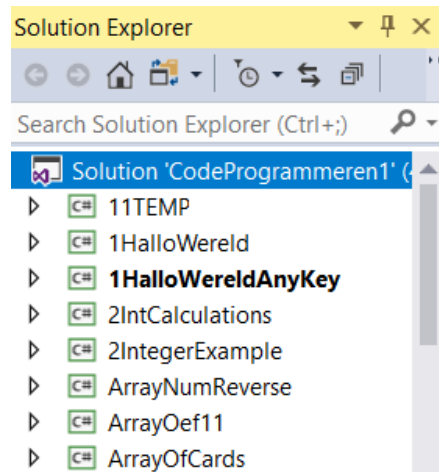


- d. Maak een nieuw project aan (met als naam 1HelloWorldAnyKey)

- e. Vergeet niet je project als startup project in te stellen, anders voer je een andere applicatie uit uit een vorig project. Dit selecteer je bovenaan



- f. Je ziet het ook in de solution explorer, dit project staat in het vet (alternatieve methode is dat je ,met rechtermuis op je project klikt en Set As Startup Project selecteerd)



Herschrijf nu je vorige oefening, maar zodanig dat “Hello” en “World” op aparte regels worden afgedrukt (zonder spatie dit keer). Zorg er nu voor dat “World” pas wordt afgedrukt nadat de gebruiker op ENTER heeft gedrukt. Sluit ook het programma pas af nadat “World” getoond werd aan de gebruiker en deze opnieuw op ENTER drukt.

3. Debug je vorige applicatie door stap per stap door de code te steppen, dit doe je door eerst een breakpoint te plaatsen op je eerste regel code die je zelf hebt toegevoegd.

Daarna kan je telkens een regel verder “steppen”. Doorloop zo heel je programma tot het afloopt. Gebruik `step into` om door de code te gaan.

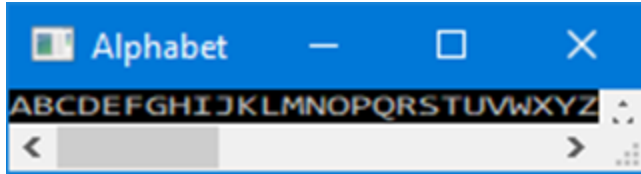
4. Herschrijf nu je vorige oefening, maar zodanig dat “Hello” en “World” op aparte regels worden afgedrukt maar met 1 Writeline instructie. Maak gebruik van een escape sequence (zoek escape sequences op in MSDN).
Sluit ook het programma pas af nadat “World” getoond werd aan de gebruiker en deze opnieuw op ENTER drukt.
5. Schrijf een programma dat volgende tekst uitschrijft:
 “Hello, my name is: XXX”
waarin je XXX vervangt door je eigen naam en voornaam, voor je naam staan geen spaties maar een tab (gebruik een escape sequence)
6. Herschrijf je vorige programma zodanig dat elk woord door een aparte instructie wordt uitgeschreven.
7. Zoek in de msdn op hoeveel versies er zijn van `Console.Write`
8. Zoek in de msdn op welke speciale karakters de `Console.WriteLine` extra toevoegt t.o.v. de `Console.Write`. Kan je deze zelf toevoegen aan een `Console.Write(“Hello World”)` zodat die toch werkt zoals een `Console.WriteLine(“Hello World”);`

Nieuwe commando's voor de volgende oefeningen, zoek de uitleg op in de MSDN:

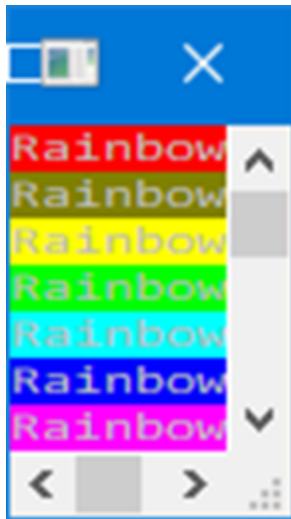
- **`Console.SetWindowSize(100, 50);`**
Verandert de grootte van het scherm
- **`Console.Title = “Test ABC”;`**
Verandert de titel bovenaan het scherm naar “*Test ABC*”
- **`Console.ForegroundColor = ConsoleColor.Red;`**
Verandert de kleur van de tekst naar rood
- **`Console.BackgroundColor = ConsoleColor.DarkYellow;`**
Verandert de achtergrond van de tekst naar donkergeel

Probeer volgende schermen na te maken met bovenstaande commando's:

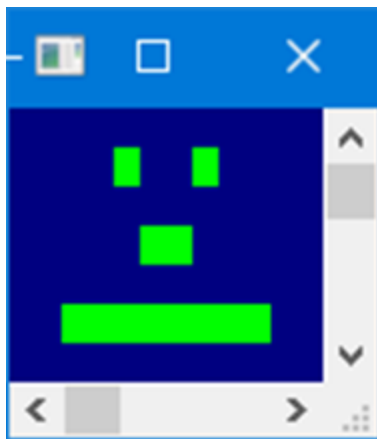
1.



2.



3.



4. Dit filmpje:

<https://i.imgur.com/kprQGag.gifv>

Herhaling Programming 1: variabelen

1. Schrijf een programma dat de voornaam van de gebruiker inleest en vervolgens de gebruiker groet met 'Hello, *first_name*'!

2. Ontwerp een toepassing waarmee je een ingevoerd bedrag in euros kan omrekenen naar dollar. Gebruik hierbij de volgende wisselkoers: 1000 EUR = 1193USD. Werk met gehele getallen. Het resultaat in de console wordt als volgt weergegeven: "X EUR is gelijk aan Y USD".
3. Schrijf een programma dat twee gehele getallen initialiseert op 3 en 8. Tel de twee getallen bij elkaar op en sla het resultaat op in een nieuwe variabele. Schrijf het resultaat naar het scherm.
4. Schrijf een programma dat twee getallen initialiseert op 3.3 en 8.8. Tel de twee getallen bij elkaar op en sla het resultaat op in een nieuwe variabele. Schrijf het resultaat naar het scherm.
5. Schrijf een programma dat de gebruiker vraagt om vijf getallen en schrijf daarna het gemiddelde van die vijf getallen naar het scherm (niet afronden).
6. Ontwerp een toepassing waarmee je een de temperatuur in graden Celsius kan omrekenen naar Fahrenheit. De formule die je hiervoor kan gebruiken is:
$$F = (9/5) \times C + 32$$

Het resultaat in de console wordt als volgt weergegeven: "A Temperature of x degrees Celsius corresponds to y degrees Fahrenheit."
7. In de KdG Winkel zijn de prijzen van alle artikelen weergegeven exclusief BTW. Ontwerp een toepassing waarmee je met de opgegeven prijs exclusief BTW de prijs inclusief BTW kan berekenen. Het BTW-tarief is 21 %. Het resultaat in de console wordt als volgt weergegeven: "De prijs inclusief BTW bedraagt: x"
8. De plaatselijke voetbalclub organiseert een mossselfestijn. Naast mosselen met frietjes (20 EUR) bieden ze voor de kinderen de mogelijkheid om een koninginnehapje (10 EUR) te kiezen. Verder is er een ijsje als nagerecht voorzien (3 EUR). Om het gemakkelijk te maken kosten alle dranken 2 EUR. Ontwerp een consoletoepassing zodat de vrijwilliger aan de kassa alleen maar de juiste aantallen moet ingeven om de totaal te betalen prijs te berekenen. Het resultaat in de console wordt als volgt weergegeven: "Het totaal te betalen bedrag is x EUR".
9. De inkomprijs voor KdGPretLand bedraagt 30 EUR voor volwassenen en 15 EUR voor kinderen tot 12 jaar. Ontwerp een consoletoepassing waarmee de kassier het aantal kinderen en volwassen kan ingeven om zo het totaal te betalen bedrag te berekenen. Voorzie een overzichtelijke weergave in de console.
10. Schrijf een programma dat de gebruiker vraagt om het huidige jaartal en om het jaar waarin hij geboren is. Reken uit hoe oud de gebruiker is aan het eind van het huidige jaar (in hele jaren) en schrijf het antwoord naar het scherm.

11. Bereken hoeveel uren vrije tijd je in een week hebt. Vraag aan de gebruiker hoeveel uur les hij heeft, en bereken aan de hand daarvan hoeveel uur hij die week vrij heeft om andere dingen te doen.

Hou er rekening mee dat je voor elke 2 uur les, 1 uur thuis studeert/werkt.

Hou er ook rekening mee dat je ook nog elke dag moet slapen! Je slaaptijd telt niet als vrije tijd.

Druk de resterende vrije tijd af.

12. Schrijf een programma dat een aantal seconden vraagt (bv 4000) en dan het aantal uren, minuten en seconden berekent die hierin passen.

Voorbeelden:

- 70 seconden wordt 1 minuut en 10 seconden
- 130 seconden wordt 2 minuten en 10 seconden
- 3601 seconden wordt 1 uur, 0 minuten en 1 seconde
- 4000 seconden wordt 1 uur, 6 minuten en 40 seconden zijn.

13. Kies minstens 2 van je voorgaande oefeningen uit en step doorheen heel het programma. Bekijk de variabelen en hun waarden eens vanuit de debug vensters: Local, Autos en Watch.

14. Schrijf een programma dat 2^{125} berekent, of **$2 \times 2 \times 2 \times \dots \times 2$** 125 keer. Maak slim gebruik van variabelen om geen 125 keer 2×2 te moeten schrijven (let op dit getal past niet in een integer).

15. Schrijf een programma dat de wortel berekent van een door de gebruiker ingevoerd getal.

16. Schrijf een programma dat **de derdemachtswortel** van een ingegeven getal berekent.

17. Schrijf een programma dat de wortel berekent van een ingegeven getal, afgerond tot 2 cijfers na de komma.

18. Schrijf een programma dat je het aantal letters uit een string laat zien.










19. Schrijf een dronken papegaai applicatie: je voert tekst in. De papegaai herhaalt wat je hebt ingegeven. Daarna herhaalt die alles al roepend (alles in hoofdletters). Daarna herhaalt die alles zonder klinkers. Daarna zegt die tussen elk woord van je originele zin "I want a cracker !!!". Zo zal "Dag papegaai, hoe gaat het er mee?" resulteren in

"Dag papegaai, hoe gaat het er mee?"

"DAG PAPEGAAI, HOE GAAT HET ER MEE?"

"Dg ppg, h gt ht r m?"

“Dag I want a cracker !!! papegaai, I want a cracker !!! hoe I want a cracker !!! gaat I want a cracker !!! het I want a cracker !!! er I want a cracker !!! mee?”

20. Schrijf een programma dat vraagt om je naam en voornaam (in 1 ReadLine instructie) en die omgekeerd in beeld brengt. Zo zal “Joop Peeters” resulteren in “Peeters Joop”.
21. Deel een van je programma's met een klasgenoot en laat die je programma runnen. M.a.w. Zorg ervoor dat de juiste exe file bij een klasgenoot geraakt.
22. Herschrijf je dronken papegaai applicatie zodat die antwoordt met speciale tekens zoals         .
23. Maak zelf een ToUpper programma zonder de methode ToUpper van string te gebruiken. Ga ervan uit dat gebruikers enkel kleine letters ingeven. Je applicatie maakt er hoofdletters van. Gebruik de ASCII tabel. Gebruik ook een vaste lengte.
24. Laat een gebruiker 2 getallen ingeven. Toon true in beeld als het eerste getal kleiner is dan het tweede.
25. Laat een gebruiker 2 namen ingeven. Toon true in beeld als de eerste naam alfabetisch voor de tweede komt (let op met hoofdletters).
26. Schrijf een programma dat een schrijffteken (unicode) inleest en het getal waarmee het overeenkomt in de unicode of ascii tabel uitschrijft (gebruik een impliciete conversie).
27. Schrijf een programma dat m.b.v. roman cypher 3 leestekens verschuift in het alfabet. Je vraagt aan de gebruiker hoeveel de letters in het alfabet verschoven moeten worden, bvb bij getal 3 wordt een 'a' een 'd' en 'b' een 'e', let op, op het einde van het alfabet moeten de letters ook terug naar het begin gaan, bvb bij 3 wordt een 'z' een 'c' en 'y' een 'b'. Ga uit van een ingevoerde tekst van 4 letters. Later zien we lussen en arrays om een meer elegante oplossing te bekomen.
28. Breidt vorige oefening uit zodat je ook terug vertaald naar de originele tekst, dus versleutelde tekst en sleutel hem terug naar het origineel

Herhaling Programming 1: Selecties

1. Om deel te nemen aan de spelen van de verkiezingen moet je minimum 18 jaar oud zijn. Ontwerp een consoletoepassing waarmee je het geboortjaar van de gebruiker opvraagt. Als de ingevoerde leeftijd minimum 18 jaar bedraagt, krijgt de gebruiker

toegang tot de stemcomputer en verschijnt de tekst: "Je bent x jaar oud. Je mag stemmen". Indien niet voldaan is aan deze voorwaarde verschijnt de tekst: "Om deel te nemen aan de verkiezingen moet je minimum 18 jaar oud zijn.". Als het geboortjaar na het huidige jaar ligt, schrijf dan naar het scherm 'Je bent nog niet geboren.'.

2. Pas de vorige oefening aan zodat dit werkt op de gehele datum, dus ook de dag en de maand.
3. Ontwerp een consoletoepassing waarmee de uitbater van een schoenwinkel het totaal te betalen bedrag kan berekenen. Als je 2 of meer paar schoenen koopt, dan krijg je 25 % korting. Een paar schoenen kost 100 EUR. Zorg ervoor dat als je een negatief getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd.
4. De solden lopen op hun einde en de uitbater van de Fnac geeft speciale kortingen. Ontwerp een consoletoepassing waarmee de uitbater van de Fnac het totaal te betalen bedrag kan berekenen. Als je één paar schoenen koopt, krijg je 20 % korting, bij de aankoop van 2 paar schoenen krijg je 40% korting en bij aankoop van meer dan twee paar schoenen zelfs 60 % korting. Een paar schoen kost 125 EUR. Zorg ervoor dat als je een negatief getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd. En controleer ook op geheel ongeldige invoer (geen getal).
5. Schrijf een programma dat de gebruiker vraagt of hij wil optellen, aftrekken, vermenigvuldigen, delen of modulo rekenen. Vraag daarna om twee gehele getallen en voer daar de gekozen bewerking op uit. Schrijf het antwoord naar het scherm. Maak deze oefening met if en met switch.
6. Ontwerp een consoletoepassing waarmee je aan de gebruiker het resultaat van 5 examens opvraagt. De opgevraagde resultaten zijn de behaalde punten op 100. De student is geslaagd als hij hoogstens 1 onvoldoende heeft en een gemiddelde van minstens 50 % behaalde. Schrijf ook het behaalde percentage uit. Zorg ervoor dat als je een negatief getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd.
7. Schrijf een programma dat een jaartal inleest en uitschrijft of het een schrikkeljaar is. Een jaar is een schrikkeljaar wanneer het deelbaar is door 4, maar niet door 100, het is echter toch een schrikkeljaar wanneer het deelbaar is door 400.
8. Ontwerp een consoletoepassing waarmee je op basis van de taalkeuze de tekst "Goeiedag" in de gekozen taal weergeeft. Doe dit voor Nederlands, Engels, Frans en Duits. Als de gebruiker opnieuw iets intypt zegt hij nog vaarwel in dezelfde taal.

9. Ontwerp een consoletoepassing waarmee je het lidgeld voor de Zoo berekent. Het lidgeld is afhankelijk van de leeftijd. Het lidgeld bedraagt 5 EUR voor iedereen jonger dan 26 jaar, anderen betalen 10 EUR. Zorg ervoor dat als je een negatief getal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd.
10. Maak een applicatie die 2 namen vraagt en controleert welke naam alfabetisch eerst komt. Zorg ervoor dat deze applicatie ook werkt als een van de 2 namen in hoofdletters is.
11. Ontwerp een console toepassing waarmee je berekent hoeveel iemand voor een bioscoopkaartje moet betalen. De volle prijs van een bioscoopticket bedraagt 13,70 EUR. Op basis van de leeftijd wordt de prijs bepaald:
 - Jonger dan 5 jaar: gratis;
 - Tussen de 5 en de 12: halve prijs;
 - Tussen de 13 en de 55: vol tarief;
 - 55+: gratis
12. Ontwerp een consoletoepassing waarmee je de dagen van de week omzet in cijfers (maandag is 1, dinsdag 2,...). Gebruik een switch. Gebruik de default om foute invoer op te vangen.
13. Schrijf een applicatie die vraagt of je de omtrek (C voor circumference) of oppervlakte (S voor surface) van een cirkel wenst te berekenen, vraag daarna naar de straal van de cirkel. Gebruik voor de berekening de waarde van PI uit de Math klasse. Zorg ervoor dat als je een negatieve straal ingeeft, de gebruiker een foutmelding krijgt en de berekening niet wordt uitgevoerd.
14. Schrijf opnieuw een programma dat m.b.v. roman cypher 3 leestekens verschuift in het alfabet. Je vraagt aan de gebruiker hoeveel de letters in het alfabet verschoven moeten worden, bvb bij getal 3 wordt een 'a' een 'd' en 'b' een 'e', let op, op het einde van het alfabet moeten de letters ook terug naar het begin gaan, bvb bij 3 wordt een 'z' een 'c' en 'y' een 'b'. Nu kan je ook een if gebruiken ipv %.
15. Breidt vorige oefening uit zodat je ook terug vertaald naar de originele tekst, dus versleutel de tekst en sleutel hem terug naar het origineel.
16. Breidt vorige oefening uit zodanig dat je zowel positieve als negatieve verschuivingen kan hebben (+3 of -5,...) zodat je met 1 programma in 2 richtingen kan vertalen.

Herhaling Programming 1: lussen (loops)

1. Schrijf een consoletoepassing die de eerste 100 even getallen in beeld brengt.
2. Schrijf een consoletoepassing dat de tafel van vermenigvuldiging afdrukt. Vraag aan de gebruiker van welk getal hij de tafel wil, en druk voor dat getal de tafel af t.e.m. x10. Gebruik gehele getallen.
3. Schrijf een consoletoepassing dat 490 keer (7 maal 70) in beeld brengt "Ik vergeef de ander zijn schuld". Gebruik een geneste while. Probeer ook eens met een for lus. En daarna met een geneste for lus.
4. We zijn het beu dat gebruikers foutieve input geven. Vraag de gebruiker om een getal in te geven. Blijf hem vragen om een getal, zolang hij iets anders ingeeft. Als hij een getal heeft ingegeven, bedankt je hem door alle getallen van 0 tot zijn getal onder elkaar te tonen.

5. Schrijf een consoletoepassing dat deze output in het console venster laat zien:

```
*  
**  
***  
****  
*****  
*****
```

Gebruik een geneste lus, probeer zowel met while als for. Schrijf elk sterretje apart uit.

6. Schrijf een consoletoepassing dat deze output in het console venster laat zien:

```
0****  
01***  
012**  
0123*  
01234
```

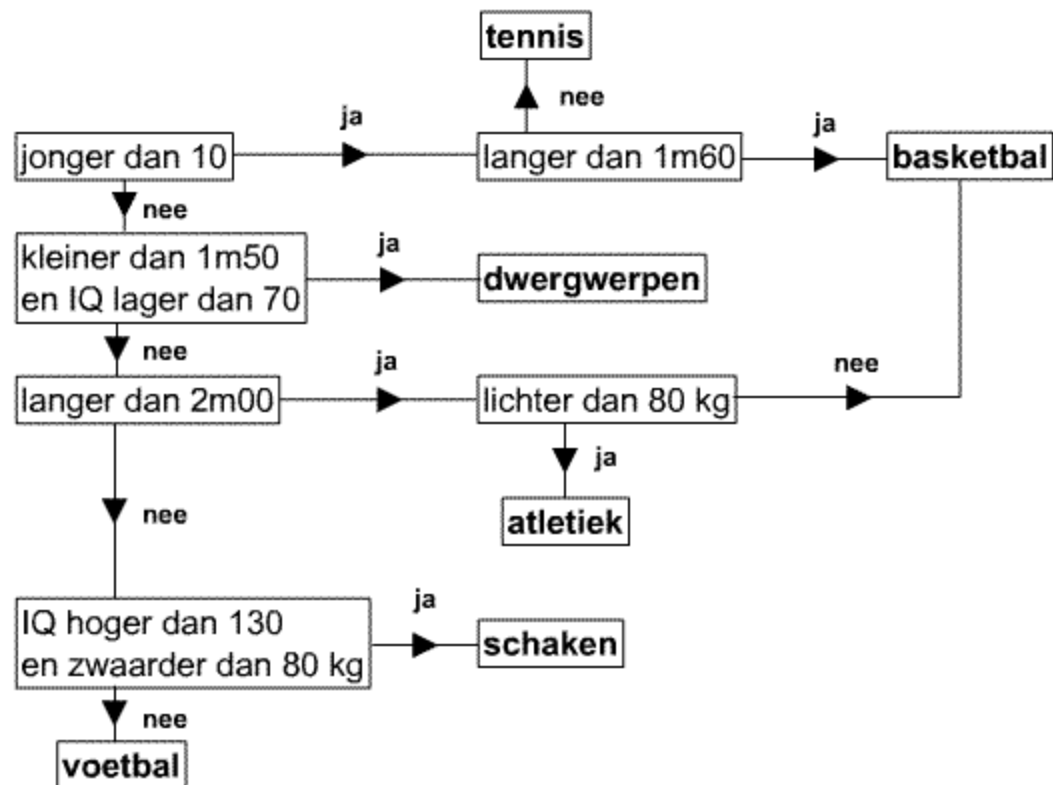
Of probeer dit:

```
****1  
***22  
**333  
*4444  
55555
```

Tip: gebruik ook weer een geneste for.

7. Schrijf een console applicatie die de gebruiker vraagt om een geheel getal groter dan 2 en de faculteit van dat getal teruggeeft. De faculteit van een getal is het getal vermenigvuldigd met elk kleiner geheel getal t.e.m. 1.
vb: $4! = 4 \times 3 \times 2 \times 1$ en $6! = 6 \times 5 \times 4 \times 3 \times 2 \times 1$
8. Dit is een uitbreiding op oefening 3.6: Schrijf een programma dat de gebruiker om 2 jaartallen vraagt > 0 . En toon de gebruiker alle schrikkeljaren die tussen die 2 jaartallen inzitten. Zorg ook dat je foutieve invoer opvangt.
9. Schrijf een programma dat de gebruiker telkens om een getal vraagt. Na elke invoer schrijft het programma de som van de tot dan toe ingevoerde getallen naar het scherm. Vervolgens krijgt de gebruiker de keuze uit de volgende drie opties: a. voer nog een getal in, b. begin opnieuw (zet de som op 0) of c. beëindig het programma.
10. Schrijf een programma dat de gebruiker telkens om een getal vraagt. Na elke invoer schrijft het programma de som van de tot dan toe ingevoerde getallen naar het scherm. Het programma stopt als de gebruiker geen geldig geheel getal invoert. Gebruik hiervoor een do while.
11. Schrijf een kassaprogramma. De gebruiker moet eerst kiezen wat hij wil aanrekenen: (1) een maaltijd (€10), (2) een drankje (€1.5), (3) een dessert (€4) of (4) niets meer. Druk de som af van de tot dan gekozen aangerekende items. Blijf vragen naar een nieuw item tot de gebruiker "niets meer" kiest. Wanneer de som groter is dan €30, geef dan een korting van 10% op de totale aangerekende prijs.
12. Schrijf een programma dat berekend of een ingegeven getal een priemgetal is of niet. Een priemgetal is enkel deelbaar door 1 en zichzelf.
13. Schrijf een programma dat de eerste 30 fibonacci-getallen afdruckt. De fibonacci-reeks is een reeks die begint met 2 x een 1, en elk volgend getal is de som van de twee vorige. Dus de reeks gaat als volgt: 1 1 2 3 5 8 13 etc. Zie https://en.wikipedia.org/wiki/Fibonacci_number voor meer info.
14. Schrijf een consoletoepassing dat de tafels van vermenigvuldiging afdruckt. Vraag aan de gebruiker tot welk getal hij wil vermenigvuldigen en maak voor elk getal tot en met dat getal de tafel t.e.m. x10.
15. Schrijf een programma dat een string vraagt en elk karakter apart uitschrijft, samen met zijn unicode waarde. Doe met een lus die gebruik maakt van de lengte van een string.

16. Schrijf een applicatie die een string vraagt en elk teken omzet met een roman cypher verschuiving van 5 karakters naar rechts. Doe met een lus die gebruik maakt van de lengte van een string.
17. Herschrijf oefening 4 en 5, maar vraag aan de gebruiker voor hoeveel rijen hij dit wil doen en pas de uitvoer ook aan zodat dit gaat voor alle ingevoerde getallen van 1 t.e.m. 9.
18. Zet het onderstaande schema om in C#. Zorg dat je alles aan de gebruiker vraagt, maar enkel indien nodig. Reken in cm en leeftijd in jaren het programma geeft terug welke sport daarbij past.



Herhaling Programming 1: Arrays

1. Schrijf een programma dat de gebruiker tien getallen in laat voeren en dat vervolgens die getallen in omgekeerde volgorde terug schrijft naar het scherm. Maak gebruik van een array om de invoer op te slaan. Pas daarna je programma aan om dit voor 20 getallen te doen. Als je dit goed gedaan hebt, hoef je maar op 1 plaats iets aan te passen.
2. Ontwerp een console applicatie waarmee je aan de gebruiker 5 getallen opvraagt. Bewaar de getallen in een array. Vervolgens vraag je aan de gebruiker het hoeveelste

getal hij wil zien. Het resultaat geef je weer in de console.

3. Schrijf een programma dat een array genereert van 20 random getallen van 1 tot en met 5. Vraag aan de gebruiker een getal tussen 1 en 5 (inbegrepen) en tel hoeveel keer dit getal voorkomt in de array. Print het aantal af.
4. Ontwerp een console waarmee je aan de gebruiker 10 getallen opvraagt. Je toont als resultaat in de console enkel de even getallen die de gebruiker heeft ingegeven.
5. Schrijf een programma dat nakijkt of een ingegeven string een palindroom is. Een palindroom is een woord dat hetzelfde blijft als je het omdraait (bv negen of redder). Je kunt een string op dezelfde manier gebruiken als een array om elk individueel karakter uit een string op te vragen (dus je kunt `text[2]` doen om het derde karakter van de string op te vragen).
6. Schrijf een programma dat aan de gebruiker vraagt hoeveel getallen hij wil ingeven. Vraag hem dan om dat aantal getallen en sla deze op in een array. Druk het gemiddelde, het minimum en het maximum van deze getallen af. Doe dit eerst zonder hulp van de Array klasse methodes en daarna ook een keer door gebruik te maken van `Array.Sort`
7. Ontwerp een console waarmee je aan de gebruiker 4 namen van studenten opvraagt. Breng die dan gesorteerd in beeld.
8. Pas oefening 6 aan zodanig dat de **getallen** van klein naar groot worden weergegeven.
9. Pas oefening 6 aan zodanig dat de getallen van groot naar klein worden weergegeven.
10. Je declareert 3 arrays met telkens 5 waarden. De eerst 2 arrays vul je op met random getallen tussen 0 en 10. De derde array vul je achteraf op met de som van de overeenkomstige velden uit array 1 en 2. Je toont in de console de 3 arrays in de vorm van een som.
11. Ontwerp een consoletoepassing waarmee je de namen van de leerlingen van je klas opvraagt. Bewaar de namen in een array. Je vraagt de gebruiker eerst hoeveel waarden hij wil ingeven en je voorziet dan voldoende ruimte in de array. Toon de namen in de console.
12. Pas de oplossing van oefening 11 aan zodat de namen alfabetisch weergegeven worden in de console.
13. Maak een programma dat eerst aan de gebruiker vraagt hoeveel getallen hij wenst in te voeren. Na de invoer van deze getallen brengt het programma de grootst ingevoerde waarde (samen met zijn positie in de tabel) op de console. Let op: indien alle ingevoerde

getallen negatief zouden zijn, mag niet op het scherm komen dat 0 de grootste ingevoerde waarde is.

14. Maak een programma dat een array van 10 elementen opvult met de waarden 1 tot en met 10 :

index	0	1	2	3	4	5	6	7	8	9
value	1	2	3	4	5	6	7	8	9	10

Breng deze getallen 1 voor 1 op de console.

VERSCHUIF nu de inhoud van de elementen naar onderstaand schema : (let op: verschuiven en dus niet opnieuw genereren)

index	0	1	2	3	4	5	6	7	8	9
value	2	3	4	5	6	7	8	9	10	1

Breng nu ter controle ook deze elementen 1 voor 1 op de console.

15. Schrijf een programma dat op een grid van 10x10 op 5 willekeurige plekken “bommen” plaatst. Gebruik hiervoor een 2-dimensionale array van booleans. Vraag vervolgens aan de gebruiker om een locatie op de grid (dus vraag om 2 getallen: x en y) en kijk of er een bom op die locatie ligt. Als er een bom ligt, is het spel afgelopen en druk je het aantal gokken af die hij heeft moeten doen om een bom te vinden.

16. Breid oefening 15 uit zodat de speler een waarschuwing krijgt als hij “vlakbij” een bom zit maar er nog niet op. Kijk hiervoor in alle vakjes rond de gekozen locatie (dus als je 2,2 gekozen hebt, kijk dan of er een bom zit in 1,1 of 1,2 of 1,3 of 2,1 of 2,3 of 3,1 of 3,2 of 3,3). Tip: let erop dat je niet over de grenzen van de array gaat (-1,0 is geen geldige locatie!).

17. Maak een console applicatie die een boek kaarten in een array myCards (type string) heeft. Harten negen kort je af H9 enz. Trek nu een willekeurige kaart uit het boek telkens de speler op enter duwt. Als de speler “fold” ingeeft, dan stop de applicatie.

Breek deze oefening op in deelproblemen en werk elk deelprobleem afzonderlijk af.

1. Maak een array van 52 kaarten
2. Trek willekeurige kaart als je op enter duwt, tot in het oneindige
3. Stop als je “fold” typt
4. Onthoud in tweede array welke kaart al getrokken is (maar doe hier voorlopig niets mee)
5. Controleer als je een kaart trekt of die al getrokken is en dus in de tweede array -steekt. Trek een nieuwe als dit het geval is (tot in het oneindige)
6. Als er 52 kaarten zijn getrokken, begint alles opnieuw.

Tips:

Gebruik duidelijke namen (arrDeckOfCards, arrDeckOfCardsPulled, mustPullAgain, cardAlreadyPulled, ...). IN HET ENGELS!!!!

Gebruik veel Console.WriteLine om duidelijk te maken wat er aan het gebeuren is (de hoeveelste kaart wordt er getrokken bijvoorbeeld)

18. Breid oefening 17 uit zodat je nooit een kaart kan trekken die al getrokken is. Dus na 52 keer een willekeurige kaart trekken, moet alles opnieuw starten. Tip: hou bij in een tweede array of een kaart al getrokken is of niet. Deze oefening kan je op veel verschillende manieren oplossen.
19. Breid oefening 18 uit zodat de boek kaarten geschud wordt. Als de speler "shuffle" ingeeft, dan wordt er geschud. Controleer of je dit goed hebt gedaan door te debuggen en zo de inhoud van de array in beeld te brengen.
20. Schrijf een programma dat een ingegeven lijst van getallen sorteert. Druk de gesorteerde lijst af. Gebruik hiervoor geen bestaande functie (zoals Array.Sort) maar implementeer alles zelf.
21. Ontwerp een console applicatie waarmee je aan de gebruiker 5 strings opvraagt. Bewaar ze in een array. Vervolgens vraag je aan de gebruiker de hoeveelste string hij wil zien. Het resultaat geef je weer in de console. Zorg dat foutieve invoer of een ongeldige index je applicatie niet doet crashen, maar proper afhandelt met een foutmelding.
22. Ontwerp een console applicatie waarmee je aan de gebruiker 5 strings opvraagt. Bewaar ze in een array. Vervolgens vraag je aan de een string en zoekt in de array op welke plaats die evt voorkomt in je array. Als de string niet voorkomt schrijf je uit "string not found". Het resultaat geef je weer in de console.
23. Schrijf een programma dat een array genereert van 20 random getallen van 1 tot en met 5. Vraag aan de gebruiker een getal tussen 1 en 5 (inbegrepen) en tel hoeveel keer dit getal voorkomt in de array. Print het aantal af.
24. Ontwerp een console waarmee je aan de gebruiker 10 getallen opvraagt. Je toont als resultaat in de console enkel de even getallen die de gebruiker heeft ingegeven.
25. Schrijf een programma dat een spelwereld genereert voor een platformspel en toont aan de speler. Sla de wereld op in een 2D array.
 - a. Vraag eerst aan de gebruiker hoe groot de wereld moet zijn (hoogte en breedte)
 - b. Elk element in de array stelt een vakje voor in de wereld. Dit kan zijn:
 - i. Berg
 - ii. Water
 - iii. Gras
 - iv. Bos
 - v. Gebruik hiervoor een zelfgemaakt enum datatype.

- c. Elk veld wordt random ingevuld, maar de rand (boven, onder, links en rechts) is altijd berg, zodat de speler niet uit het veld kan lopen.
 - d. De wereld wordt niet met de namen van de velden getoond, maar 1 letter per veldje (berg = M, water = ~, Gras is i, bos is F van Bos)
 - e. Als de wereld getoond is vraag je of de gebruiker tevreden is met de wereld of, of hij een nieuwe wil genereren (de grootte blijft gelijk) en voer je zijn wensen uit.
26. Schrijf een blad-steen-schaar spelletje. Vraag eerst de naam van de speler. De computer vraagt: daarna in een lus 'R'(ock), 'P'(aper), 'S'(cissors), 'A'(play again) of 'E'(xit) aan de gebruiker. Hij genereert een zelf random antwoord (op basis van een getal), en steeds verschillend van het vorige antwoord dat hij gegeven heeft. Je controleert wie er wint en toont de huidige score (in de vorm Computer XX punten, SPELERNAAM XX punten). Bij Play Again zet je de scores terug op 0, bij Exit, schrijf je eerst uit wie de winnaar is.
27. Variatie op oefening 22 uit met een BEST OF 5 concept dus wie het eerste 3x wint is de winnaar. Je speelt dus maximaal 5 rondes. Hou voor elke ronde in een 3D array bij: wat de speler heeft gekozen, wat de computer heeft gekozen en wie de winnaar is ('S' voor speler, 'C' voor computer). Je hoeft niet te vragen of de speler wil stoppen, het spel stopt automatisch als er 1 winnaar is. Zorg wel dat foutieve invoer wordt opgevangen en de vraag wat de speler kiest wel opnieuw gesteld wordt. Sla de ingevoerde cijfers op in een 2D array (eerste rij zijn de keuzes van speler 1, tweede rij zijn de keuzes van speler 2).
28. Schrijf een programma dat aan de gebruiker vraagt hoeveel rijen en hoeveel kolommen er moeten zijn die een random char bevatten. Vul deze in met random chars (waarden tussen 65 en 91 (A-Z)). Nu ga je de rijen en de kolommen verwisselen, Maak dus een nieuwe 2D array maar met evenveel rijen als de eerste array kolommen heeft en evenveel kolommen als de eerste rijen heeft. Steek nu in de 2e array elk element op de getransponeerde positie (rijnr wordt kolomnr en omgekeerd, zie vb figuur)
- | | | | | | | | |
|---|---|---|--|--|---|---|---|
| A | B | C | | | A | D | G |
| D | E | F | | | B | E | H |
| G | H | J | | | C | F | J |
- ➡
29. Microsoft Mijnenveger <https://youtu.be/Z0EAysRluJk>
30. Game of life <https://youtu.be/ouipbDkwHWA>

Herhaling Programming 1: Methodes

1. Schrijf een methode `AskNumber`, die een geheel getal aan de gebruiker vraagt en `true` of `false` teruggeeft als dit geslaagd is of niet. Gebruik een out-parameter om het getal terug te geven. Het resulterende getal is 0 als de gebruiker geen geldige invoer heeft ingegeven. Deze methode kan je ook in verdere oefeningen hergebruiken. Test door vanuit je main deze methode op te roepen en het resulterende getal te tonen aan de gebruiker.
2. Schrijf een methode `KeepAskingNumber` zodat je de vraag om een getal blijft herhalen totdat er een geldig getal wordt ingegeven. Nu geef je de waarde terug als resultaat en dus geen bool meer (want de methode stopt pas als er een geldig getal werd ingegeven). Deze methode heeft geen parameters. Test vanuit je main. Gebruik hiervoor een loop.
3. Herschrijf oefening 1 en 2 zodanig dat dit werkt voor floating point getallen.
4. Maak een nieuw project aan waarin je je code voor `KeepAskingNumber` uit oefening 2 kopieert. Maak een array aan van 10 integers, loop nu met een lus door je array en vul elk element in m.b.v. je `KeepAskingNumber` methode. Schrijf daarna met een `foreach` lus al je elementen van de array uit in je console.
5. Maak een programma dat 5 getallen vraagt aan de gebruiker, en deze optelt en op het einde de som afdruckt. Gebruik `KeepAskingNumber` zodat je foutief ingegeven getallen opvangt zonder dat het programma crasht.
6. Breid oefening 5 uit zodat je eerst aan de gebruiker vraagt hoeveel getallen hij wil ingeven (dus niet altijd 5).
7. Schrijf een functie `IsEven` die als parameter, een geheel getal heeft en als resultaat of dit getal even is (of anders gezegd: deelbaar is door 2) of niet. (Boolean). Vraag een getal aan de gebruiker en controleer of dit even is en schrijf dit ook uit vanuit je main methode.
8. Schrijf een methode, `AddOne` die 1 optelt bij zijn integer parameter en het resultaat teruggeeft. Roep de methode 4x op vanuit je main en toon het resultaat. Van de oproepen, zijn er 2 die rechtstreeks een waarde invoeren als parameter en 2 die een waarde by value doorgeeft vanuit een eerder aangemaakte variabele uit je main.
9. Schrijf een methode `CmToInch`, die bij een gegeven aantal centimeters het aantal inches berekent en retourneert. Vermenigvuldig het aantal centimeters met 0.394 om het aantal inches te verkrijgen. Test met meerdere waarden vanuit Main.
10. Schrijf een methode `CalculateCubeVolume` die bij gegeven lengte van een ribbe (=zijde van een kubus) het volume van de kubus retourneert.

11. Schrijf een methode `Smallest` die als parameters twee gehele getallen heeft en die als resultaat het kleinste van de twee getallen geeft.
12. Schrijf een methode `TimeDiffInSec` met zes parameters en een integer als resultaat. De invoer bestaat uit twee tijdstippen in uren, minuten en seconden. De methode moet het verschil tussen deze tijdstippen in seconden retourneren. Test met meerdere waarden vanuit `Main`.
13. Schrijf een methode `AppendStrings`, die 3 string parameters meekrijgt en deze achter elkaar plakt. Het resultaat is 1 string die wordt geretourneerd. Test vanuit je `main` door het resultaat van de aanroep rechtstreeks vanuit een `Console.WriteLine` uit te schrijven.
14. Schrijf een methode `SumArray` die een array van gehele getallen als parameter mee krijgt. Zorg ervoor dat het `return`-resultaat de som is van alle getallen in de array.
15. Maak een array aan en steek er 10 gehele getallen in. Gebruik de `AddOne` methode uit opgave 8 om elk getal uit de array met 1 te verhogen. Doe dit in een `for` lus. Schrijf de array daarna uit met behulp van een `foreach` lus.
16. Schrijf een methode `CmArrayToInchArray`, die bij een gegeven array van cms het aantal inches berekent, in een nieuwe array plaatst en die array retourneert. Vermenigvuldig het aantal centimeters met 0.394 om het aantal inches te verkrijgen. Of gebruik de methode `CmToInch` uit opgave 9. Test met een array die je maakt in je `Main`.
17. Schrijf een methode `StringToArray` die een string als parameter krijgt, en een array van strings teruggeeft. Elke string in deze array bevat exact 1 karakter uit de input-string. Gebruik hiervoor `Substring`.
18. Schrijf een functie `HowManyAreEven` die als parameter, een array van gehele getallen meekrijgt en als resultaat teruggeeft hoeveel van de getallen die erin zitten (mag random zijn) even zijn (of anders gezegd: deelbaar is door 2) of niet. Maak in je `main` een array met random 10 getallen en test je methode.
19. Schrijf een methode, `AddOneToArray` die 1 optelt bij alle elementen van een integer array parameter. Als resultaat `return` je ook de som van alle nieuwe getallen in de array. Roep de methode 4x op vanuit je `main` en toon het resultaat. Roep dit op vanuit je `main` met een array van 5 random getallen.
20. Schrijf een methode `CmToInch`, die bij een gegeven aantal centimeters het aantal inches berekent en in een tweede out parameter opslaat. Vermenigvuldig het aantal centimeters met 0.394 om het aantal inches te verkrijgen. Test met meerdere waarden vanuit `Main`.
21. Maak een array aan en steek er 20 random kommagetallen in. Geef de array mee als ref aan een `ArrayTimes2` methode en verdubbel hierin elk getal. Schrijf de array voor

en na het uitvoeren van de methode in de main uit m.b.v. een foreach lus. Random kommagetallen genereer je m.b.v Random.NextDouble()

22. Schrijf een methode InchToCm, die een ref parameter heeft met het gegeven aantal inches. Gebruik diezelfde ref parameter om het resultaat van de omrekening naar centimeters op te slaan. Het aantal inches berekent en in een tweede out parameter opslaat. Deel het aantal inches door 0.394 om het aantal cm's te verkrijgen. Test met meerdere verschillende waarden vanuit Main.
23. Schrijf een functie AddOneAndCountEven die als parameter, een array van gehele getallen meekrijgt en in 2 out parameters opslaat:
 - a. Een array met alle elementen van de array -1.
 - b. hoeveel van de originele getallen die erin zitten even zijnMaak in je main een array met random 5 getallen en test je methode.
24. Schrijf een methode, AddOneToArray die 1 optelt bij alle elementen van een integer array parameter. Als resultaat return je ook de som van alle nieuwe getallen in de array. Roep de methode 4x op vanuit je main en toon het resultaat. Roep dit op vanuit je main met een array van 5 random getallen.
25. Vraag aan de gebruiker hoeveel getallen hij in een array wil steken. Steek in een array er evenveel random kommagetallen in. Geef de array mee als ref aan een ArrayTimes2 methode en verdubbel hierin elk getal. Schrijf de array voor en na het uitvoeren van de methode in de main uit m.b.v. een foreach lus. Random kommagetallen genereer je m.b.v Random.NextDouble().
26. Maak een console applicatie die een boek kaarten in een array myCards (type string) genereert heeft. Maak gebruik van een methode GenerateDeck, die een volledig kaartspel (52 kaarten genereert). Deze gebruikt een tweede methode GenerateCardString die 2 elementen omzet in de naam van de kaart (eerste parameter is een integer die de soort representeert, tweede parameter representeert de waarde 1 - koning = 13). Schrijf na de methode generateDeck het kaartspel uit.

Herhaling Programming 1: Enums, Lists en andere datatypen

1. Maak een console applicatie die een boek kaarten in een list myCards (type string) heeft. Harten negen kort je af H9, ruiten boer met R11, schoppen aas het S1, klaveren koning met K13 enz. Genereer deze lijst met for-loops. Trek nu een willekeurige kaart uit het boek telken de speler op enter duwt. Als de speler "fold" ingeeft, dan stop de applicatie.

2. Breid oefening 1 uit zodat je nooit een kaart kan trekken die al getrokken is. Dus na 52 keer een willekeurige kaart trekken, moet alles opnieuw starten. Deze oefening kan je op veel verschillende manieren oplossen.
3. Pas vorige oefening aan, gebruik een Enum om je type kaart te specificeren.
4. Breid oefening 2 uit zodat de boek kaarten geschud wordt. Controleer of je dit goed hebt gedaan door te debuggen en zo de inhoud van de list in beeld te brengen.
5. Programmeer op basis van vorige oefeningen een vereenvoudigd blackjack spel. Elke kaart krijgt gewoon zijn waarde, een prentje = 10 en een aas is altijd 1. Je wint het spel als je exact 21 kan halen. Je vraagt de speler telkens of hij nog een kaart wil. Na elk spel toon je of de speler gewonnen heeft of verloren en vraag je of hij nog een keer wil spelen.
6. Schrijf een programma dat de gebruiker tien getallen in laat voeren en dat vervolgens die getallen in omgekeerde volgorde terug schrijft naar het scherm. Maak gebruik van een list om de invoer op te slaan. Pas daarna je programma aan om dit voor 20 getallen te doen. Als je dit goed gedaan hebt, hoef je maar op 1 plaats iets aan te passen.
7. Ontwerp een console applicatie waarmee je aan de gebruiker 5 getallen opvraagt. Bewaar de getallen in een list. Vervolgens vraag je aan de gebruiker het hoeveelste getal hij wil zien. Het resultaat geef je weer in de console en verwijder je uit de lijst.
8. Ontwerp een console waarmee je aan de gebruiker getallen laat opvragen tot hij stop intypt. Je slaat de getallen op in een lijst. Je toont als resultaat in de console enkel de even getallen, in gesorteerde volgorde van groot naar klein.
9. Ontwerp een console waarmee je aan de gebruiker 4 namen van studenten opvraagt en opslaat in een lijst. Breng die dan gesorteerd in beeld.
10. Pas oefening 7 aan zodanig dat de **getallen** van klein naar groot worden weergegeven.
11. Schrijf een programma dat de gebruiker telkens om een getal vraagt. Na elke invoer schrijft het programma de som van de tot dan toe ingevoerde getallen naar het scherm. Vervolgens krijgt de gebruiker de keuze uit de volgende drie opties: a. voer nog een getal in, b. begin opnieuw (zet de som op 0) of c. beëindig het programma. Als het programma beëindigd wordt schrijf je nog 1x de hele lijst van getallen uit als 1 grote som en ook het resultaat van die som.
12. Maak een programma dat een list opvult met de waarden 1 tot en met 10 :
index 0 1 2 3 4 5 6 7 8 9

value 1 2 3 4 5 6 7 8 9 10

Breng deze getallen 1 voor 1 op de console.

VERSCHUIF nu de inhoud van de elementen naar onderstaand schema : (let op: verschuiven en dus niet opnieuw genereren)

index 0 1 2 3 4 5 6 7 8 9

value 2 3 4 5 6 7 8 9 0 1

Breng nu ter controle ook deze elementen 1 voor 1 op de console.

13. Schrijf een programma dat een lijst maakt van 10 willekeurige VormTypes. Een VormType is een Circle, een Triangle, een Rectangle of een Square. Daarna schrijf je met een lus de types uit met daarachter de formule om hun oppervlakte te berekenen.
14. Schrijf een programma dat een enum aanmaakt voor de vier windrichtingen. Bereken een random getal die de windrichting bepaald. De hoek het dichtste bij 1 van de hoofdrichtingen (noord = 0° of 360°, oost = 90°, zuid = 180°, west = 270°) wordt dan getoond. Herhaal tot de gebruiker stop zegt.
15. Schrijf een programma dat de gebruiker om 5 namen vraagt. Voeg deze toe aan een lijst van strings. Druk de hele lijst af met een loop als de 5 namen ingegeven zijn.
16. Schrijf een programma dat de gebruiker om een naam vraagt. Voeg deze naam toe aan de lijst. Blijf om nieuwe namen vragen tot de gebruiker STOP typt. Druk dan de volledige lijst af.
17. Breid het programma uit oefening 16 uit. Nadat de gebruiker STOP heeft gedrukt, vraag je opnieuw naar een naam. Maar deze keer VERWIJDER je de naam uit de lijst. Druk daarna de lijst met de overgebleven namen af.
18. Maak een Enum Language aan die drie elementen bevat: Dutch, French en English. Gebruik een random getal om één van de drie Enum-opties te kiezen, en groet de gebruiker in deze taal.
19. Maak een applicatie die een enum RainbowColors definieert. De kleuren zijn red, orange, yellow, green, blue, indigo, violet. Geef elke kleur ook het nummer dat overeenkomt met zijn positie in de regenboog (dus red = 1,...).
Vraag nu aan de gebruiker een kleur als naam, komt de kleur niet voor in de regenboog, dan schrijf je uit, This color is not a part of the rainbow. Anders Schrijf je uit, This color is de Xth color in the rainbow (vervang X door de waarde van de kleur).
20. Maak een applicatie die een enum TemperatureFeeling heeft voor hot, warm en cold. Het is cold als de temperatuur kleiner is dan 15 graden, hot als het warmer is dan 30 graden en anders warm. Gebruik de waarden die in de enum overeenkomen met de namen als grenzen voor je if selecties (zodat we makkelijk ook hot kunnen definiëren als

35 graden en er niets anders aan de applicatie verandert). Geef voor 10 random getallen aan of ze cold, warm of hot zijn. Maak een methode die op basis van dat getal een TemperatureFeeling teruggeeft

Herhaling Programming 1:

DEEL 8: Klassen

1. Ga naar de online Tower Defense game Bloons Tower Defense 5 (<https://ninjakiwi.com/Nederlands/Games/Tower-Defense/Play/Bloons-Tower-Defense-5.html#.XFF97IxKhPY>). Ga na wat de mogelijke klassen zijn en wat de objecten in onderstaande screenshot. Welke eigenschappen en methoden zouden de klassen kunnen hebben. Hebben alle objecten van een klasse dezelfde eigenschappen?



2. Maak een klasse Mission aan, die de missie van een spel zal bijhouden. Deze Missie klasse houdt volgende gegevens bij:
 - a. Een missienaam

- b. introtext
- c. een status (ToDo, InProgress, Failed, Completed) => gebruik enum
- d. pointsWhenCompleted
- e. outro text

Daarnaast heeft de klasse

- een methode Complete, die geeft een string terug bestaande uit de outro text en heeft een out parameter, waarin wordt ingevuld hoeveel aan de score zal worden toegevoegd. Verander de status dan ook naar Completed.
- voor elk gegeven properties.
- een methode StartMission, die de status verandert naar InProgress, deze returnt de introtext
- een methode Fail, die de status verandert naar Failed

Maak in je main methode twee missies aan, vul alle gegevens in. en start de missie, complete 1 missie en fail 1 missie. Toon alles wat er gebeurt proper in de console en check met debuggen/steppen door je code of alles lukt.

3. Ontwerp een klasse voor de opslag van de gegevens van leerlingen. Sla de voornaam, de naam en het geboortjaar elk op in een objectvariabele. Voeg voor elk gegeven properties toe en maak een methode die alle gegevens als text teruggeeft. Maak in je hoofdprogramma 5 leerling objecten aan (met verschillende gegevens) en schrijf hun data uit. Maak ook een methode GetFullName die de volledige naam teruggeeft als hulpmethode (intern in de klasse).
4. Ontwerp een klasse voor de opslag van de gegevens van een rechthoek. Sla de breedte en de hoogte elk op in een objectvariabele. Maak een methode die de oppervlakte van de rechthoek als waarde teruggeeft. Gebruik deze in de main om het resultaat weer te geven. Breid de oefening uit om 1000 rechthoeken met willekeurige breedte en hoogte aan te maken en geef de gemiddelde oppervlakte weer in het console venster. Verwijder daarna alle rechthoeken uit de lijst met een oppervlakte kleiner dan 10.
5. Ontwerp een klasse CD voor de opslag van de titel en de uitvoerder. De constructor (als je dat al kan) maakt een nieuwe CD aan met de ingegeven titel en uitvoerder. Voeg twee methoden toe om de inhoud van de velden titel en uitvoerder op te vragen. Voeg nog een veld toe waarin een lijst van liedjes staat. Zorg er ook voor dat je een methode hebt waarmee je een liedje kan toevoegen. Voeg een methode toe om alle informatie over de titel, uitvoerder en de lijst van liedjes omvormt naar tekst. Maak in je Main een lijst van CD's. Maak een Shuffle methode die alle nummers van alle CD's oplist in willekeurige volgorde. Voorzie een duidelijke weergave van de informatie in je console.
6. Ter promotie van Belgische appels en peren wordt een nieuw type fruitautomaat gelanceerd. Maak een nieuwe klasse Fruitautomaat aan. Een fruitautomaat bevat appels en peren. Het maximum aantal van elk is gelimiteerd op 50 stuks.

Voeg vervolgens een aantal methoden toe om:

- appels aan te vullen;
- peren aan te vullen;
- appels te verkopen;
- peren te verkopen;
- te kijken hoeveel stukken fruit er nog in de automaat zitten;
- te controleren of een bepaalde fruitsoort uitverkocht is.

Houd er rekening mee dat je bijvoorbeeld geen appel kan verkopen als er geen appels meer in de automaat zitten.

7. Ontwerp een klasse `StudentData` voor de opslag van de naam en punten (in %) van een student. Maak properties aan om zowel de naam als de punten in te geven. Voeg een `Print`-methode toe die de naam en punten van de student afdruckt. Maak 3 verschillende variabelen aan van type `StudentData`, vul ze op met verschillende gegevens, en gebruik de `Print`-methode om de info af te printen.

8. Voeg een methode `GetGrade` toe die de graad teruggeeft van een student. Dit zijn de mogelijkheden:

- < 50: niet geslaagd;
- tussen 50 en 68: voldoende;
- tussen 68 en 75: onderscheiding;
- tussen 75 en 85: grote onderscheiding;
- > 85: grootste onderscheiding.

Druk naast de naam en de punten ook de graad af van elke student. In het Engels:

failed,
sufficient,
distinction,
great_distinction,
greatest_distinction

9. Maak een extra methode `IsFailed` aan die `true` teruggeeft als de student < 50% heeft en `false` als de student wel geslaagd is. Druk dit ook af.

10. Gebruik nu een array in plaats van losse variabelen om de `StudentData` op te slaan. Vraag eerst aan de gebruiker hoeveel studenten hij wil ingeven. Vraag vervolgens de naam en de punten van elke student in een loop, en vul zo de array op. Druk uiteindelijk dezelfde informatie af van elke student als in oefening 7.

11. Doe hetzelfde als in oefening 9, maar nu druk je de geslaagde en gefaalde studenten apart af. Dus je drukt eerst af “Deze studenten zijn geslaagd:”, gevolgd door alle geslaagde studenten in de array. Daarna druk je af “Deze studenten zijn gefaald:”, gevolgd door alle gefaalde studenten. Gebruik hiervoor 2 aparte loops.

Herhaling Programming 1: Overloading & Constructors

1. Maak 3 methodes voor het uitprinten van data met de naam PrintData. Eenmaal voor tekst, eenmaal voor gehele getallen en eenmaal voor kommagetallen. Deze printen respectievelijk TEXT : + parameter, INT + parameter, FLOAT + parameter uit in de console. Roep deze op vanuit je main met verschillende parameters, check m.b.v. debugging en *step into* dat de juiste methode wordt opgeroepen.
2. Maak methodes voor alle combinaties van het vermenigvuldigen van float en int. Dus een int * int, float * int, int * float, float * float. Het resultaat is altijd een float. Kan je alle methodes aanroepen? Probeer dit vanuit je main.
3. Ontwerp een klasse Fraction voor de opslag van de gegevens van een breuk, namelijk teller en noemer (beide ints). Voorzie 2 constructors. De eerste constructor maakt een nieuwe breuk aan met standaardwaarden (bv 1/1). De tweede constructor voorziet parameters voor het ingeven van teller en noemer. Voorzie een Print-methode in de main die de breuk afdruckt (je geeft de breuk mee als parameter). Voorzie je een methode om het quotiënt uit te rekenen. Test voor verschillende breuken. Voorzie alle klassevariabelen van een property (en zorg ervoor dat niemand het cijfer 0 in de noemer kan invullen).
4. Voeg een methode Add toe aan de Fraction-klasse, die een teller meekrijgt, en deze teller optelt bij de teller in je Fraction-object. Overload deze methode zodat je ook een teller en een noemer kunt meegeven. Je moet dan eerst de noemer van beide breuken gelijkstellen om de tellers te kunnen optellen (zie wiskunde lagere school <https://www.youtube.com/watch?v=zMJ3zWPffG8>). Voeg tenslotte een derde overloaded methode Add toe die een ander Fraction-object krijgt, en tel deze bij elkaar op op dezelfde manier als bij de vorige methode. Test met verschillende waarden en overloaded methodes, en print telkens het resultaat af.
5. Maak een klasse Color, die intern 3 integers bijhoudt, waarden tussen 0 en 255, om een kleur op te slaan voor rood/groen/blauw. Zorg dat deze 2 constructors heeft waarmee de

waarden kunnen ingevuld worden. De eerste neemt 3 integer variabelen aan als parameter, die gebruikt worden om de klassevariabelen in te vullen. De 2e neemt float variabelen aan (die van waarden tussen 0 en 1 worden omgevormd naar integers tussen 0 en 255 <https://www.youtube.com/watch?v=W5g-TiE6aZk>). Daarnaast zijn er ook 2 methoden ChangeColor, ook met een int en een float versie. Maak een methode GetColor die de waarden als string teruggeeft. Test vanuit je main meerdere MyColor objecten.

Extra: Maak een enum PredefinedColors met daarin de kleuren rood (255,0,0), Green (0,255,0) en Blue (0,0,255). Zorg dat je aan je constructor en je changecolor ook een PredefinedColor variabele kan meegeven.

6. Maak voor elke klasse-oefening in deel 8 een constructor die op basis van parameters elke klassevariabele initialiseert. Wat gebeurt er bij het aanmaken van je objecten van die klasse? Los dit op door:
 - a. Eerst: een default constructor (zonder parameters) zelf te overloaden die alle klassevariabelen op default waarden initialiseert.
 - b. Daarna: door het aanmaken van de objecten te vervangen door oproepen van deze nieuwe constructor (zet de default constructor even in commentaar)
7. Maak een klasse ConsoleColor die als constructor 2 parameters aanneemt: de achtergrondkleur en de voorgrondkleur (de kleur van de tekst). Beide parameters zijn van type **ConsoleColor**. Dit is een enum die je reeds gebruikt hebt in de eerste oefeningenreeks, maar toen wist je nog niet dat dit een enum was. Sla beide parameters op in een membervariabele. Voeg een methode PrintColors toe die afdrukt wat de opgeslagen kleuren zijn. Een variabele van type **ConsoleColor** kun je converteren naar een string door er de **ToString()**-methode op te roepen. Dit kan bij elke enum! Test dit door een ConsoleColor-object aan te maken en de PrintColors-methode op te roepen. Onthoud: je kunt een enum meegeven als parameter door bv **ConsoleColor.Red** te typen.
8. Voeg een methode WriteLine toe die een string als parameter neemt, en de string afdrukt met als voorgrond- en achtergrondkleur de kleur uit de membervariabelen. Dit kun je doen door **Console.ForegroundColor = color**, **Console.BackgroundColor = color**, en **Console.WriteLine()** te gebruiken.
9. Voeg een methode ChangeColors toe die, net zoals de constructor, de achtergrond- en voorgrondkleur meekrijgt als parameters, en deze opslaat in de membervariabelen van de klasse. Gebruik deze methode om verschillende keren WriteLine op te roepen met verschillende achtergrond- en voorgrondkleurencombinaties.
10. Tenslotte, voeg een overloaded versie van WriteLine toe aan ConsoleColor die als parameters een string en twee ConsoleColor-parameters binnenkrijgt. Druk deze string

af met de achtergrond- en voorgrondkleur die je als parameter hebt meegegeven.

11. Pas beide overloaded WriteLine-methodes aan zodat je onthoudt wat de originele kleuren waren van de console (dus de originele waarden van **Console.ForegroundColor** en **Console.BackgroundColor**). Deze waarden zet je na het oproepen van **Console.WriteLine()** terug. Test dit door een aantal keer WriteLine op te roepen op een ConsoleColor-object en dan een gewone klassieke Console.WriteLine. Deze zou afgedrukt moeten worden in de oorspronkelijke klassieke kleuren van de console.

Herhaling Programming 1: Text File I/O

1.
 - a. Maak een klasse Animal met hierin
 - i. Soort
 - ii. Aantalpoten
 - iii. Een constructor
 - iv. Een methode GetText, die de animal data in tekstvorm teruggeeft
 - v. Maak een methode writeAnimalToFile die een FileStream meekrijgt als parameter.
 - b. Maak in je main een array van 5 Animal objecten en schrijf deze naar file. Check of de file ok is
 - c. Maak een methode readAnimals in je Program.cs die een filenaam meekrijgt. Deze methode leest alle animal data terug in en maakt er terug objecten van met de waarden die overeenkomen met de originele 5 animal objecten (gebruik nergens de waarde 5, deze code moet ook werken met een ander aantal animals in de file. Schrijf deze array terug uit in je main
2. Maak een nieuw project aan, voeg hier ook je klasse Animal aan toe (kopieer de file van oefening 1 naar je projectmap, en voeg toe als existing item aan je project in VS.
 - a. Maak in je main een array van 4 animal objecten
 - b. Maak een methode SaveAnimals, die de array als parameter meekrijgt en een filePath. In de file schrijf je eerst hoe groot de array is, daarna schrijf je elk animal weg met zijn writeAnimalToFile methode.
 - c. Maak een methode LoadAnimals die als out parameter een array van animals meekrijgt en een filepath. Na het uitvoeren van van LoadAnimals is de array parameter aangemaakt en ingevuld door de methode, met alle data uit de file

- d. In je main:
 - i. Schrijf elk animal uit (maak een GetText methode die zijn data als string teruggeeft, deze kan je ook in SaveAnimals gebruiken)
 - ii. Sla je originele animal array op mbv SaveAnimals op je Desktop in animals.txt.
 - iii. Declareer een nieuwe array variabele loadedAnimals
 - iv. Roep LoadAnimals met loadedAnimals en je animals.txt filepath op je desktop
 - v. Schrijf je loadedAnimals uit, en check of deze identiek zijn aan de originele
 - vi.
- 3. Kopieer je vorige oefening. Voeg aan je Animal klasse een constructor toe die een string meekrijgt als data en een object invult op basis van die string. Zorg nu dat LoadAnimals deze constructor gebruikt om objecten te maken op basis van een ingelezen string.
- 4.
 - a. Maak een klasse ConsoleSettings hierin maak je variabelen voor
 - i. Voorgrondkleur (evt als string)
 - ii. Achtergrondkleur (evt als string)
 - iii. Aantal opstartbeeps
 - b. In je hoofdprogramma
 - i. Check of je de file consolesettings.txt kan lezen
 - ii. Indien ja
 - 1. Lees de info uit file en maak een consolesetting variabele op basis van de ingelezen data
 - 2. Stel de consolesettings in, speel de beeps en toon "Hello Word" in de opgeslagen kleuren aan de gebruiker
 - iii. Indien neen
 - iv. Vraag aan de gebruiker een voorgrond en achtergrondkleur en aantal beeps en sla ze op in consolesettings.txt

Herhaling Programming 1: Collections

- 1.
 - a. Vraag aan de gebruiker een string text. Voer van de ingelezen tekst elk karakter individueel toe aan een Queue. Schrijf daarna elk element uit de queue terug uit.
 - b. Vraag aan de gebruiker een string text. Voer van de ingelezen tekst elk karakter individueel toe aan een Stack. Schrijf daarna elk element uit de stack terug uit.
 - c. Maak een methode CheckPalindrome, die test of een ingevoerde string een palindroom is (woord dat van achter naar voor hetzelfde is). Doe dit door elk karakter in een stack en in een queue te steken, daarna check je of de

elementen die je eruit haalt hetzelfde zijn. Zijn ze allemaal gelijk (en evenveel), dan heb je een palindrome (want een stack is LIFO, queue is FIFO)

2. Maak een programma dat checkt of een reeks haakjes uitgebalanceerd is (dus haakjes mooi in volgorde en open en gesloten. Je hebt drie soorten haakjes: (), [], {} die in elkaar geplaatst mogen worden. Je vraagt aan de gebruiker een reeks haakjes. Maak een methode `IsBalanced` die teruggeeft of ze in balans zijn of niet, Gebruik een Stack.
DIT IS EEN STEVIGE DOORDENKER
3. Maak een applicatie waar je strings mee inleest (zogezegd url's) en die uitprint als `SHOWING: url`. Je voegt ze ook toe aan een stack. Als de string een gelijk is aan `backspace`, betekent dit dat je terug wil naar de vorige url, dan toon je de vorige url (bovenste element van de stack). Als je `backspace` drukt en de stack is leeg, dan schrijf je uit `SHOWING: HOME`.
4. `HashSet` oefening. Vraag aan de gebruiker een woord, sla alle karakters uit dit woord in een `HashSet` op. Toon ofdat een letter al in de hashset zat als je het toevoegt of niet. Loop door alle chars van a-z en schrijf uit of dit karakter in het ingevoerde woord zat (als het in de hashSet zit).
5.
 - a. Maak een klasse `Bullet`, waarin een integer `damage` zit, voeg een constructor toe die je een `damage` kan meegeven.
 - b. Maak een klasse `Gun`, die een stack `magazine` heeft van `Bullets`, die is oorspronkelijk leeg. Maak een methode `AddBullet`, die een `Bullet` toevoegt aan de `magazine` variabele.
 - c. Maak een methode `Fire`, die je de bovenste bullet van `magazine` teruggeeft en verwijdert.
 - d. Maak een methode `CheckNextBullet`, die returnt de bovenste bullet van `magazine` zonder die te verwijderen.
 - e. In je `main`, maak je een `Gun` aan.
 - f. Als je op `L(oad)` drukt voeg je 5 kogels toe met random `damage` tussen 10 en 100,
 - g. druk je op `F(ire)`, dan schiet je 1 kogel af. Print af, "Shot bullet with damage X" waarin X de `damage` van die kogel is
 - h. Druk je op `C(heckMagazine)`, dan schrijf je uit wat de volgende kogel zou zijn die je afschiet.
6.
 - a. Maak een methode die een string meekrijgt en een `hashset` returnt met daarin alle karakters die voorkomen in de tekst
 - b. Vraag aan de gebruiker een string, en voer de methode uit. Check of elk letter in het alfabet voorkomt in de tekst of niet en schrijf dit uit

- c. Vraag aan de gebruiker een filenaam, lees nu heel de tekst uit de file in en doe hetzelfde voor die.
- d. Maak een nieuwe methode, die hetzelfde doet, maar in plaats van individuele karakters slaat hij alle woorden op in de hashset. Splitten van heel je tekst kan bvb zo (enkele alternatieven in comment):

```
//string[] words = text.Split(); //splits on all whitespaces
char[] separators = new char[] { ',', '.', '!', '?', '\\', '\\\"', ' ', '\\n', '\\t' };
string[] words = text.Split(separators);
//string[] words = Regex.Split(text, @"\\s+");
```

- e. Toon alle woorden van de file in de console.
7. Maak een Dictionary aan waarbij een char een key is en een getal als value
- a. Hermaak nu vorige oefening waarbij je in de dictionary bijhoudt hoe vaak elk karakter voorkomt in een ingevoerde string. Toon in de console
 - b. Hermaak nu vorige oefening waarbij je in de dictionary bijhoudt hoe vaak elk woord voorkomt in een file.

Geavanceerdere oefeningen

Deze oefeningen lijken op wat we op het examen zullen vragen. Het gaat hem over complexere problemen die wat meer logisch denken vragen. Deel elke opgave op in kleine stukjes en tackle dan stukje per stukje. Dat lukt meestal niet in een keer. Dus ga stap per stap te werk.

1 Twitter sentiment analyse

Tweets kan je downloaden naar een csv-file: <https://twitter-sentiment-csv.herokuapp.com/>. Kies zelf een onderwerp, bijvoorbeeld "Trump". We kunnen deze file inlezen en tweet per tweet nagaan of de tekst positief is of negatief (of neutraal). Dat staat op zich ook in de file zelf, maar we willen dit zelf kunnen. We zullen alle tweets analyseren met woorden uit een woordenboek die voorzien zijn van een sentimentscore:

https://github.com/cjhutto/vaderSentiment/blob/master/vaderSentiment/vader_lexicon.txt

Gebruik deze woorden om voor elke tweet een score te berekenen (gebruik de woorden uit de eerste kolom en de score uit de tweede kolom).

Dit kan er dan zo uit zien:



```
D:\Data\02 KdG\04 Programmeren\2018-2019\Programming1\Les\bin\Debug\Les.exe

"RT @EdKrasen: INSTEAD OF GIVING $5 BILLION FOR TRUMP'S WALL\n\n- Fix Flint's water crisis ~ $55 mil\n- Feed all homeless vets 3 meals/day for."
Constains crisis so adds -3,1 to score
Constains giving so adds 1,4 to score
Total score for tweet: -1,7

"RT @EdKrasen: Stop saying 'Donald Trump is not racist.'\n\n'Not racist' people don't joke about mass genocide of another race of people."
Constains joke so adds 1,2 to score
Constains stop so adds -1,2 to score
Total score for tweet: 0

"RT @ezralevin: In 2000
Total score for tweet: 0

"RT @TheUSASingers: I have never been more embarrassed and ashamed of anyone or anything than I am to have Donald Trump be the President of."
Constains ashamed so adds -2,1 to score
Constains embarrassed so adds -1,5 to score
Total score for tweet: -3,6

"RT @tedlieu: Dear @SecPompeo: I look forward to you testifying before the Foreign Affairs Committee. Some questions:\n\n-Why are you ignoring."
Constains dear so adds 1,6 to score
Total score for tweet: 1,6

"ok
Total score for tweet: 0
```


2 Roman substitution cypher hacken

Met volgende tool kan je een tekst encrypteren volgens een Roman substitution cypher:

<https://www.xarg.org/tools/caesar-cipher/>. De sleutel is dan hoeveel plaatsen er zijn opgeschoven. Als de sleutel "3" is, dan wil dat zeggen alle letters met 3 opschuiven (dus een a wordt een d) en andersom om te decrypteren (een d wordt een a).

Maak zelf een tool om een tekstfile (die je zelf kiest) de encrypteren naar een nieuwe tekstfile. Geef de geëncrypteerde tekst aan aan medestudent, maar vertel die niet wat de sleutel is. Aan hem om de originele tekst opnieuw tevoorschijn te halen.

Tip: gebruik een dictionary met <char, double> op basis van

<http://www.gutenberg.org/files/5200/5200.txt> (lange Engelse tekst). Bij char zet je alle letters van het alfabet. Bij double hoeveel keer ze voorkomen in een grote tekst (% kans, dus <a,10> wil zeggen van alle karakters komt de a in 10% van de gevallen voor). Als je deze dictionary niet hebt uit vorige oefeningen, dan moet je die nog zelf maken. Zoek online zelf voor lange teksten op bijvoorbeeld wikipedia.

Tip: als <char, double> niet goed werkt, probeer dan <string, double> en zoek op woorden ipv individuele characters.

Dit werkt uiteraard enkel voor grote teksten.



3. Ontwerp een klasse CD voor de opslag van de titel en de uitvoerder. De constructor maakt een nieuwe CD aan met de ingegeven titel en uitvoerder. Voeg twee methoden toe om de inhoud van de velden titel en uitvoerder op te vragen. Voeg nog een veld toe waarin een lijst van liedjes staat. Zorg er ook voor dat je een methode hebt waarmee je een liedje kan toevoegen. Voeg een methode toe om alle informatie over de titel, uitvoerder en de lijst van liedjes omvormt naar tekst.

Maak een klasse Jukebox die een collectie van CD's heeft. Elke cd kan maar 1x voorkomen. Deze heeft een methode om cd's toe te voegen en om zichzelf naar file te schrijven.

Zorg ervoor dat je de CD met elk van hun nummers naar file kan schrijven en terug kan inladen in een nieuw JukeBox object.

Maak in je Main een lijst van CD's. Maak een Shuffle methode die alle nummers van alle CD's oplijst in willekeurige volgorde. Voorzie een duidelijke weergave van de informatie in je console.

4. Maak een klasse ConsoleSettings hierin maak je variabelen voor

- v. Voorgrondkleur (evt als string)
- vi. Achtergrondkleur (evt als string)
- vii. Aantal opstartbeeps

Maak een klasse MultipleSettings, die een lijst bevat van consolesettings

- Zorg dat je een AddConsoleSettings methode hebt
- Deze klasse bevat een methode WriteToFile die het hele object naar file schrijft
- Deze klasse bevat een constructor die een filenaam meekrijgt en via de filenaam het object initialiseert. Gebruik een hulpmethode ReadFromFile

Maak in je main meerdere settings aan, zorg dat je ze kan opslaan in 1 multiplesettings object, en sla dit op in een file. Maak een tweede object aan die je creeert op basis van je file.

Maak een menu die je toelaat te vragen de hoeveelste setting je wil tonen en toon deze setting in tekstvorm in de console.

Advanced Oefeningen :

Overerven en overriden

8. Maak op basis van je klasse Mission uit deel 8: Klassen een overervende klasse BonusMission. Deze voegt extra variabelen toe voor een aantal bonuspunten en een boolean die bijhoudt of die bonuspunten behaald zijn of niet. Maak voor beide ook properties. Overschrijf hierin de methode Complete van de basisklasse. Deze overschrijvende returnt net hetzelfde als de basisklasse, maar in de out parameter voeg je de bonuspunten toe als de bonuspunten behaald zijn. Let op met de constructors! Test je nieuwe klasse uit in de console (aanmaken, starten, completen).
9. Werk verder op basis van vorige oefening. Zorg dat in je basisklasse Mission enkel een constructor is die alle klassevariabelen initialiseert (dus geen default constructor). Maak nu een nieuwe constructor aan in de BonusMission klasse, die ook alle variabelen

initialiseert. De variabelen van de Mission klasse worden doorgegeven aan de constructor van die klasse. Test of je nog een constructor kan schrijven zonder oproepen van de base constructor. Test het aanmaken van een bonusmission met die nieuwe constructor en step doorheen de code om te zien wat er in de constructors gebeurt (step into).

10. Maak een klasse Animal.

- a. Elk Animal heeft een aantal poten en een boolean die bijhoudt of het longen heeft of niet. Zorg voor een constructor met parameters en properties voor de klassevariabelen. Daarnaast heeft de klasse ook een ToString methode die als text teruggeeft (om in main uit te schrijven): "Animal with X paws and lungs", of no longs als er geen longen zijn. X vervang je door het aantal poten.
- b. Maak een klasse Fish die overerft van Animal en een variabele voor de kleur toevoegt. Maak hier een constructor met parameters. Daarnaast overschrijf je de methode ToString, zodat Animal vervangt door Fish, altijd no longs uitschrijft en achteraan "my color is..." + de kleur toevoegt
- c. Maak een klasse Dog, die overerft van Animal en een boolean heeft die bijhoudt of de hond lang haar heeft of niet. Maak een methode ToString. Deze overschrijft ToString van de basisklasse, hij gebruikt eerst alles uit dat de basisklasse ToString uitschrijft, en daarachter voegt hij toe: "Barking dog with long/short hair", long of short hangt af van de boolean variabele.
- d. Maak in je console applicatie een array van 5 animals. Een gewoon animal, 2 vissen en 2 honden. Kies je parameters zelf, en loop met een lusje door je array om van elk dier de ToString in de console te tonen.
- e. Voeg in je Dog klasse naast de constructor met parameters ook een default constructor zelf aan (zonder parameters). Zorg dat deze ook een default constructor van de basisklasse kan oproepen. Zorg wel dat alle variabelen geïnitieerd zijn (dat mag steeds op dezelfde waarde zijn). Test het uit met nieuwe objecten.
- f. Vervang je array door een 2D array met daarin random dieren (zet een random getal om in een dier, een 0 => animal, een 1 een fish, een 2 een dog. Gebruik hiervoor een switch in een lus. En schrijf daarna voor elke animal de ToString uit

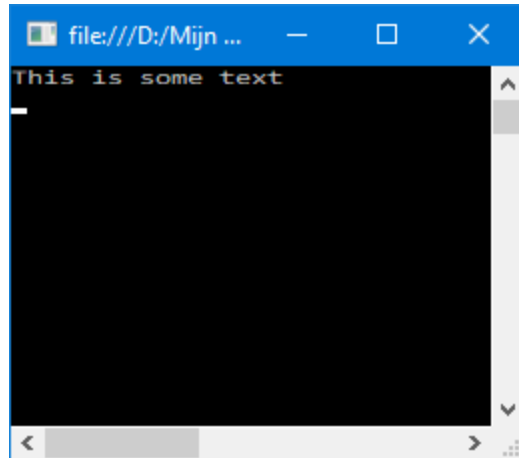
11. Ontwerp een klasse Point voor de opslag van twee velden, namelijk de x- en de y-positie. Geef de klasse een default constructor die een punt (0,0) aanmaakt en een constructor waaraan je de x- en de y-coördinaat kan meegeven als parameter. Voorzie een methode om de coördinaten op te vragen (een string met daarin beide coördinaten). Ontwerp daarna de klasse Dot. Deze klasse erft de eigenschappen van de klasse Point. Het point is namelijk de positie van de Dot (een punt heeft geen dikte). Voeg aan de klasse Dot het veld "radius" (straal) toe. Voorzie een methode om het aantal pixels van een Dot te bepalen (afgeronde oppervlakte). Ook deze klasse heeft 2 constructors: 1

met en 1 zonder parameters. De standaard straal is 5. Test met verschillende Dot objecten die met verschillende constructors zijn aangemaakt in je main methode. Gebruik een List om objecten in op te slaan.

Zorg dat je constructor van de Dot klasse gebruik maakt van de klasse Point constructor met parameters om zijn x en y waarden in te stellen. Override de methode die de coördinaten teruggeeft als tekst om ook de oppervlakte terug te geven. Maak hierbij gebruik van de base om de x en y al mooi als tekst terug te krijgen.

12. Ontwerp een systeem via overerving voor een platenzaak die cd's en dvd's verkoopt.
 - a. Van elke cd en dvd wordt de titel, de prijs en het aantal stuks in voorraad bijgehouden. Van de cd's wordt ook bijgehouden hoeveel liedjes erop staan.
 - b. Maak een aantal willekeurige objecten aan. Je slaat alle objecten op in een lijst van een basisklasse.
 - c. De zaakvoerder wenst een overzicht van het aantal stuks in voorraad en de totale verkoopwaarde van de hele voorraad van ieder artikel. Programmeer deze laatste 2 methodes in je console applicatie.
 - d. Breidt het assortiment uit met games, deze hebben naast dezelfde basiselementen ook een pegi rating 12+, 16+, 18+. Gebruik hiervoor een enum. Voeg een aantal game objecten toe aan je lijst.
 - e. Maak in je hoofdprogramma een methode die alle eigenschappen van alle items in het gamma uitschrijft. Doe dit met een methode GetText die je in de basisklasse al aanmaakt, en overschrijft en uitbreidt met de extra variabelen (gebruik makende van de base).
13. We gaan een object-geïntendeerde aanpak met polymorfisme maken om een reeks classes te schrijven die tekst op verschillende manieren afdruckt, vergelijkbaar met oefening 7/8 uit de vorige oefeningenreeks. Nu gaan we gebruik maken van inheritance om een hele familie van printers te definiëren, die elk op hun eigen unieke manier tekst naar de console wegschrijven.
 - a. Maak een basisklasse Printer aan. Deze klasse bevat enkel een virtual methode Print die een string als parameter krijgt, en deze afdruckt in de console met Console.WriteLine.

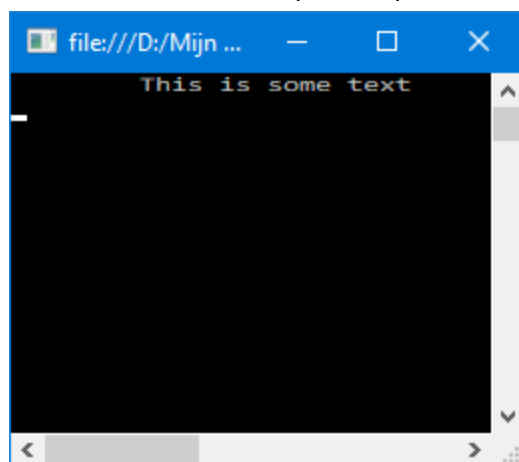
Test dit in de main-methode door een object van dit type aan te maken en Print op te roepen.

A screenshot of a Windows-style console window. The title bar is blue and contains the text 'file:///D:/Mijn ...' followed by standard window control icons (minimize, maximize, close). The console area has a black background. The text 'This is some text' is printed in a monospaced font at the top left. Below the text, there is a small white cursor line. The bottom of the window shows a horizontal scrollbar.

- b. Maak een nieuwe klasse `TabPrinter` aan die afgeleid is van de basisklasse `Printer`. `TabPrinter` heeft een constructor die 1 parameter aanneemt: een `int` met het aantal spaties dat afgedrukt moet worden VOOR de tekst. Sla deze waarde op in een membervariabele.

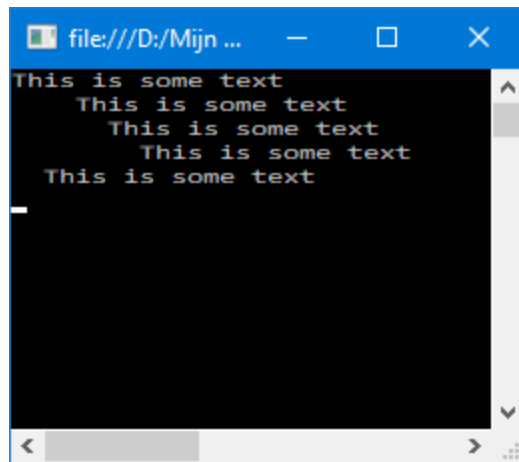
Override vervolgens de methode `Print`: in plaats van gewoon de tekst af te drukken, druk je eerst het aantal opgegeven spaties af voor de tekst, zodat de tekst een beetje naar rechts staat in de console.

Test dit door een object van type `TabPrinter` aan te maken in de `main`-methode en de `Print`-methode op te roepen:

A screenshot of a Windows-style console window, similar to the one above. The title bar is blue and contains the text 'file:///D:/Mijn ...'. The console area has a black background. The text 'This is some text' is printed in a monospaced font, but it is indented to the right, preceded by several spaces. Below the text, there is a small white cursor line. The bottom of the window shows a horizontal scrollbar.

- c. Maak een `List` aan waarin een object van basisklasse `Printer` steekt. Maak vervolgens een paar objecten van type `TabPrinter` aan met verschillende `space`-parameters, en steek die allemaal in dezelfde lijst. Loop over de lijst, en roep op elk object in de lijst de `Print`-methode op. Merk op dat nu dezelfde tekst

wordt afgedrukt, maar met meer en minder spaties voor de lijn.



```
file:///D:/Mijn ...  
This is some text  
    This is some text  
        This is some text  
            This is some text  
                This is some text
```

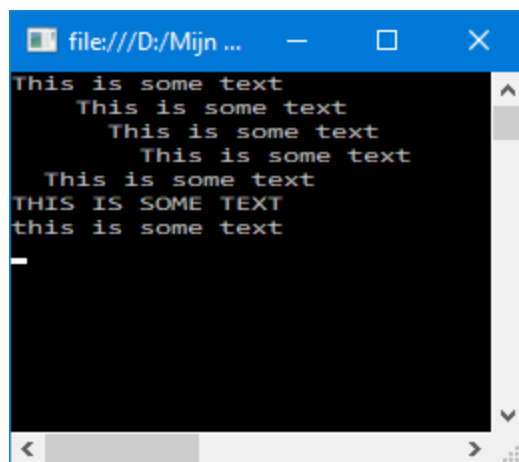
- d. Maak een class CapsPrinter aan die afgeleid is van Printer. CapsPrinter moet een string volledig in kleine of grote letters afdrukken, afhankelijk van de parameter van de constructor. De constructor neemt een bool-parameter, en als deze true is, moet de string volledig IN CAPS worden afgedrukt, anders volledig in kleine letters.

In de Print-methode die je override, kun je text.ToUpper() en text.ToLower() gebruiken om een string te converteren naar caps of kleine letters.

Voor meer info, zie:

[https://msdn.microsoft.com/en-us/library/ewddd6aed\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ewddd6aed(v=vs.110).aspx)

Test dit door een CapsPrinter-object toe te voegen aan de bestaan de lijst. Hij zal nu automatisch mee afgedrukt worden door de loop:

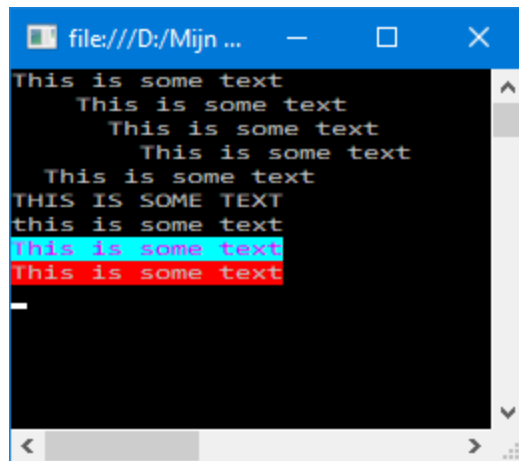


```
file:///D:/Mijn ...  
This is some text  
    This is some text  
        This is some text  
            This is some text  
                This is some text  
THIS IS SOME TEXT
```

- e. Maak een nieuwe klasse `ColorPrinter` aan die afgeleid is van de basisklasse `Printer`. `ColorPrinter` heeft een constructor die 2 parameters aanneemt: de achtergrondkleur en de voorgrondkleur (de kleur van de tekst). Beide parameters zijn van type `ConsoleColor`. Dit is een enum die je reeds gebruikt hebt in de eerste oefeningenreeks, maar toen wist je nog niet dat dit een enum was. Sla beide parameters op in een membervariabele.

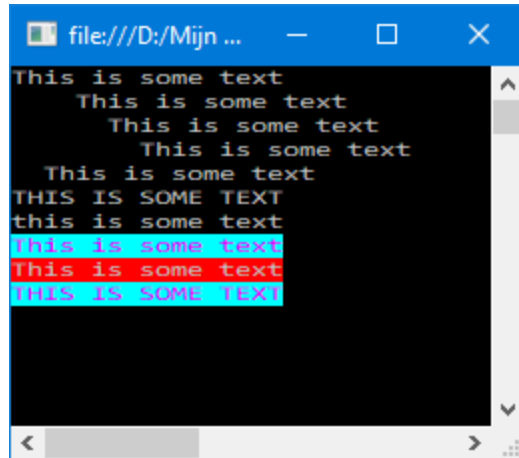
Override de methode `Print` uit de basisklasse, zodat je de tekst afdruckt in de achtergrond- en voorgrondkleur die je hebt meegegeven met de constructor. Dit kun je doen door `Console.ForegroundColor = color, Console.BackgroundColor = color` te gebruiken.

Test dit door twee verschillende `ColorPrinters` toe te voegen aan de lijst:

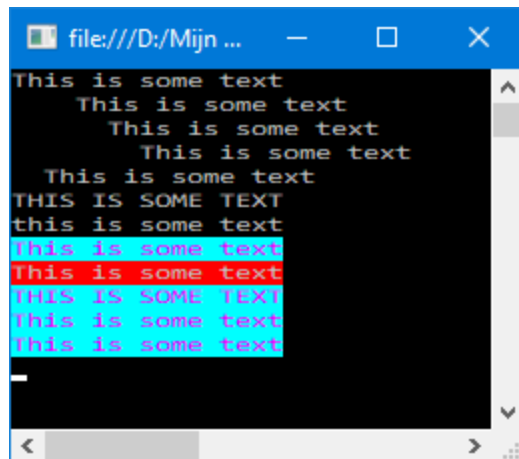


- f. Maak een nieuwe class `CapsColorPrinter` aan die afgeleid is van `ColorPrinter`, niet van `Printer`! Hierdoor krijg je een derde niveau in de OO-hiërarchie: `CapsColorPrinter` is afgeleid van `ColorPrinter` en `ColorPrinter` is afgeleid van `Printer`. Let op: omdat `ColorPrinter` een constructor gedefinieerd heeft, moet je er ook één definiëren in `CapsColorPrinter`, want de default constructor is vervallen. Maak een nieuwe constructor aan met de twee kleuren-parameters, en geef deze door aan de constructor van `ColorPrinter` via `base`.

`CapsColorPrinter` override de methode `Print` opnieuw, waardoor de `Print`-methode van `ColorPrinter` ook overschreven wordt. `CapsColorPrinter` moet eerst de tekst converteren naar uppercase, zodat de tekst zowel in uppercase als in kleur wordt afgedrukt. Roep na het converteren naar uppercase de `Print`-methode van `ColorPrinter` op door `base.Print(text)` te doen.



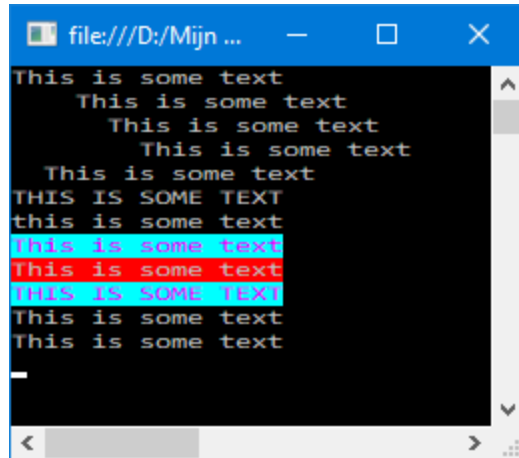
- g. Er zit nog een bug in ColorPrinter. Als je na ColorPrinter bv een TabPrinter gebruikt, zal de TabPrinter ook in dezelfde kleur blijven afdrukken als de laatste ColorPrinter! Test dit door in de List op het einde een TabPrinter toe te voegen:



Dit komt omdat we `Console.ForegroundColor` veranderen, maar het daarna niet terugzetten naar de oorspronkelijke waarde. Pas de methode `Print` aan in `ColorPrinter`, zodat de originele waarde van `Console.ForegroundColor` en `Console.BackgroundColor` eerst opgeslagen wordt in een tijdelijke variabele, zodat je daarna de kleuren terug naar de oorspronkelijke waarden kunt zetten.

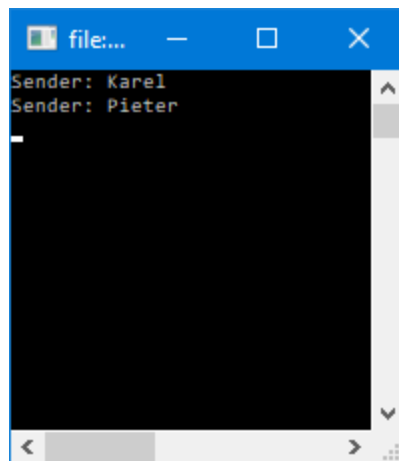
Test dit en merk op dat de bug ook opgelost is voor `CapsColorPrinter`! Dit is de kracht van OO: door af te leiden van `ColorPrinter` erft `CapsColorPrinter` alle functionaliteit (en bug fixes) van de basisklasse: handig!

Definitieve output:



```
file:///D:/Mijn ...
This is some text
  This is some text
    This is some text
      This is some text
        This is some text
THIS IS SOME TEXT
this is some text
This is some text
This is some text
THIS IS SOME TEXT
This is some text
This is some text
```


14. In deze oefening schrijven we een mailbox-systeem (real-life, geen e-mail syteem). Hiervoor gaan we een klasse-hiërarchie maken van verschillende soorten zaken die je op de post kunt doen (pakketjes, brieven) en een klasse voor de mailbox.
- a. Schrijf een class Mail die de basisklasse vormt voor onze hiërarchie. Deze class bevat een membervariabele sender die de verzender bevat van de mail, in een string. Maak een virtuele methode ShowContent die de verzender afdruckt. Test dit door een aantal mail-objecten aan te maken en ShowContent op te roepen.



```
file:...
Sender: Karel
Sender: Pieter
```

- b. Maak twee afgeleide classes Letter en Package. Letter bevat een extra membervariabele text, en Package bevat extra membervariabeles weight en content. Override de methode ShowContent om de extra info af te drukken. Vergeet niet dat je ook de verzender wilt afdrukken voor zowel Package als Letter!

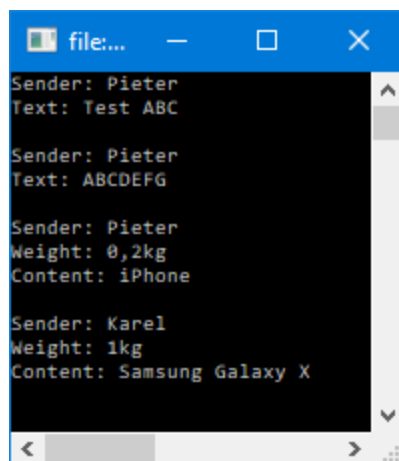
Test dit door enkele Letter- en Package-objecten aan te maken en ShowContent op te roepen.



```
file:...\nSender: Karel\nText: Hallo, dit is een brief\n\nSender: Pieter\nWeight: 0,2kg\nContent: iPhone X
```

- c. Maak een class Mailbox. Mailbox heeft als membervariabele een string owner die de eigenaar bevat van deze mailbox. Vul deze op met een constructor. Mailbox bevat ook een tweede membervariabele: een List van type Mail. Schrijf een methode Post op mailbox die een Mail-object aanneemt en in de lijst zet. Schrijf een andere methode Open, die over de lijst gaat, en voor elk element ShowContent oproept. Omdat we de mailbox dan geleegd hebben, moeten de mails ook uit de lijst gehaald worden nadat ze zijn afgedrukt met ShowContent.

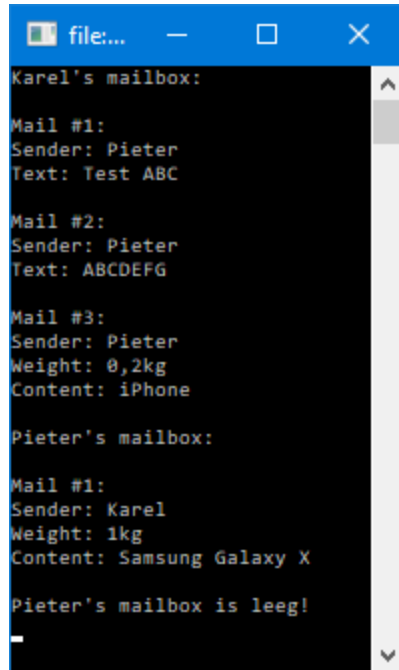
Test dit door een paar objecten van type Mailbox aan te maken, en een aantal verschillende Letter/Package-objecten te posten op deze mailboxes met de Post-methode. Roep vervolgens Open op de mailboxes en kijk of de inhoud goed wordt afgedrukt.



```
file:...\nSender: Pieter\nText: Test ABC\n\nSender: Pieter\nText: ABCDEFG\n\nSender: Pieter\nWeight: 0,2kg\nContent: iPhone\n\nSender: Karel\nWeight: 1kg\nContent: Samsung Galaxy X
```

- d. Kuis de manier waarop de mailbox wordt afgeprint op. Print, voor de inhoud van de lijst, eerst af van wie de mailbox is. Print bij elk Mail-object ook af het

hoeveelste in de brievenbus dat is. Voeg whitespace toe zodat het duidelijker is wat de verschillende pakketjes zijn. Voeg ook een bericht toe als de mailbox leeg is. Als je dus twee keer na elkaar Open oproept op dezelfde mailbox, moet hij de tweede keer melden dat de mailbox leeg is. De output moet er zo uit zien:

A screenshot of a terminal window with a blue title bar containing the text 'file:...' and standard window controls. The terminal has a black background with white text. The output is as follows:

```
Karel's mailbox:

Mail #1:
Sender: Pieter
Text: Test ABC

Mail #2:
Sender: Pieter
Text: ABCDEFG

Mail #3:
Sender: Pieter
Weight: 0,2kg
Content: iPhone

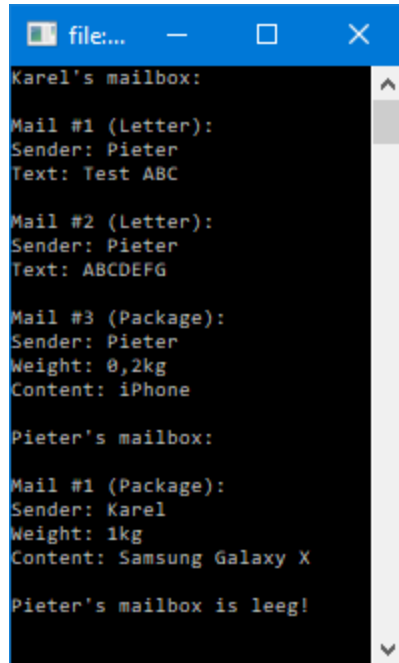
Pieter's mailbox:

Mail #1:
Sender: Karel
Weight: 1kg
Content: Samsung Galaxy X

Pieter's mailbox is leeg!
```

- e. Voeg een virtuele methode GetMailType toe aan de basisklasse Mail. Deze methode geeft als een string het soort terug (Package of Letter of gewoon Mail). Override deze methode in Package en Letter zodat er respectievelijk "Package" en "Letter" teruggegeven wordt.

Gebruik deze methode om in de Open-methode van Mailbox af te drukken wat voor type pakket het is, als volgt:

A screenshot of a Windows file explorer window with a blue title bar. The window displays a text file with the following content:

```
Karel's mailbox:

Mail #1 (Letter):
Sender: Pieter
Text: Test ABC

Mail #2 (Letter):
Sender: Pieter
Text: ABCDEFG

Mail #3 (Package):
Sender: Pieter
Weight: 0,2kg
Content: iPhone

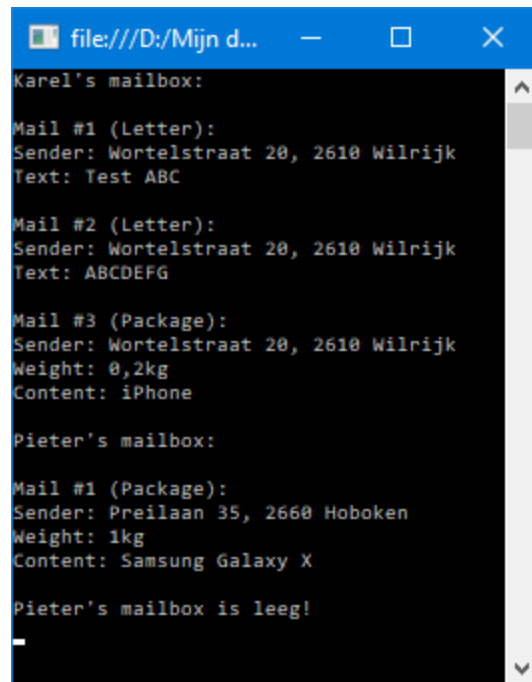
Pieter's mailbox:

Mail #1 (Package):
Sender: Karel
Weight: 1kg
Content: Samsung Galaxy X

Pieter's mailbox is leeg!
```

- f. Schrijf tenslotte een klasse Address die onze sender-string moet vervangen. Address bevat 4 membervariabelen: street, streetNumber, postCode, city. Vul deze op met een constructor. Schrijf een GetDescription-methode die een string teruggeeft die het adres bevat (bv Straat 10, 2610 Wilrijk). Vervang nu overal in Mail, Letter en Package de sender-string door een Address object. In de ShowContent-methode van Mail zul je GetDescription van Address gebruiken om de sender af te drukken.

Je zult vervolgens je Main-methode moeten aanpassen zodat je eerst een Address-object aanmaakt voor elke Letter/Package die je verzendt. Je zou nu een project moeten hebben met de volgende files/classes: Mail, Letter, Package, Address, Mailbox. De finale output van deze oefening zou er dan als volgt uit moeten zien:



```
file:///D:/Mijn d...
Karel's mailbox:
Mail #1 (Letter):
Sender: Wortelstraat 20, 2610 Wilrijk
Text: Test ABC

Mail #2 (Letter):
Sender: Wortelstraat 20, 2610 Wilrijk
Text: ABCDEFG

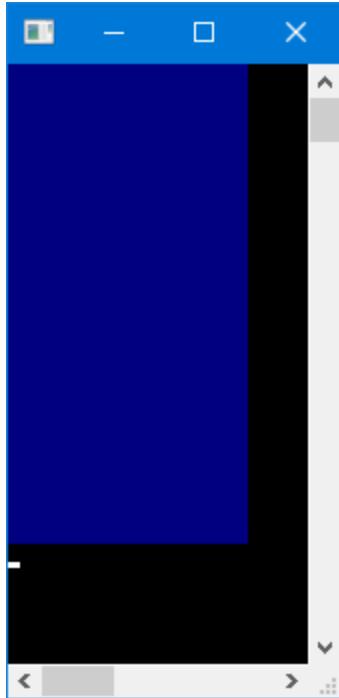
Mail #3 (Package):
Sender: Wortelstraat 20, 2610 Wilrijk
Weight: 0,2kg
Content: iPhone

Pieter's mailbox:
Mail #1 (Package):
Sender: Preillaan 35, 2660 Hoboken
Weight: 1kg
Content: Samsung Galaxy X

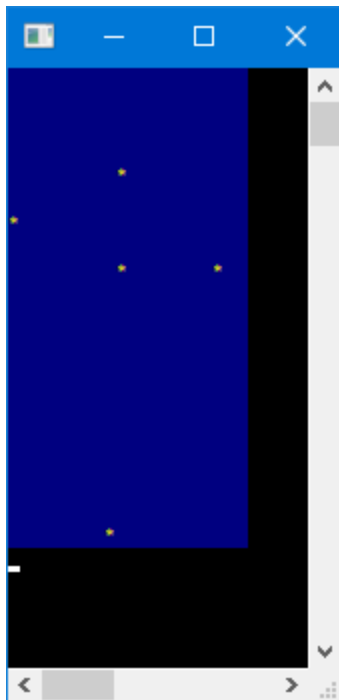
Pieter's mailbox is leeg!
```

15. We gaan een sterrenhemel implementeren die we kunnen printen in de console.

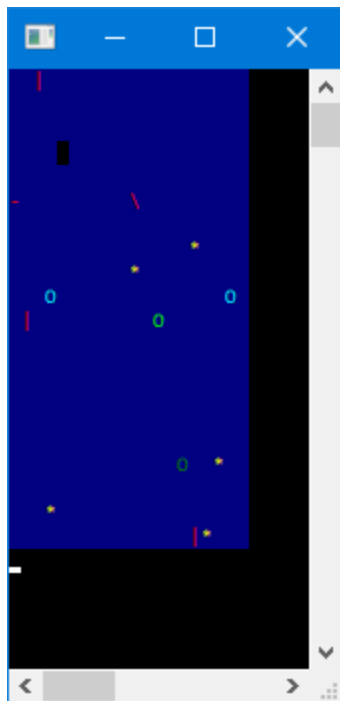
- a. Maak een klasse SkyElement die een deel van de sterrenhemel moet voorstellen. Een SkyElement-object staat voor een leeg stuk in de lucht, en drukt dus gewoon een spatie af. Schrijf een Print methode in SkyElement die deze spatie afdruckt met achtergrondkleur DarkBlue.
- b. Maak een klasse NightSky die als membervariabele een 2D-array van SkyElement-objecten bevat. De constructor van NightSky krijgt een int size mee, die bepaalt hoe groot deze 2D-array zal zijn. Vul in de constructor deze 2D-array op met nieuwe SkyElement-objecten. Schrijf een Print methode die door de array loopt en de sterrenhemel afdruckt door op elk SkyElement-object de Print-methode op te roepen. Maak in je Main methode een object van type NightSky aan, en roep daar de Print-methode van op. Dit zou er dan zo uit moeten zien:



- c. Schrijf nu een klasse `Star` die afgeleid is van `SkyElement`. Maak de `Print`-methode van `SkyElement` virtual, en override deze methode in `Star` zodat er in de plaats van een spatie, een gele `*` wordt afgedrukt. In de constructor van `NightSky` maak je 5 `Star`-objecten aan op random posities in de array. Roep de `Print`-methode van `NightSky` op om te zien of dit gelukt is:



- d. Maak nu een nieuwe klasse Planet, ook afgeleid van SkyElement. Een Planet krijgt in zijn constructor een kleur mee. In de Print-methode druk je een O af in de meegegeven kleur. Voeg aan NightSky een aantal Planet-objecten toe op random posities in de lucht, en geef ze ook een random kleur, te kiezen uit Cyan, DarkGreen en paars.
- e. Maak een enum CometDirection aan, die vier waardes kan hebben: Horizontal, Vertical, LeftDown of RightDown. Maak een nieuwe klasse Comet aan, afgeleid van SkyElement, die in de constructor een CometDirection-waarde meekrijgt. In de Print-methode druk je dan respectievelijk -, |, / of \ af in het rood, afhankelijk van welke CometDirection je hebt meegegeven.
- f. Maak een klasse BlackHole aan, afgeleid van SkyElement. Een BlackHole drukt in de Print-methode gewoon een spatie af maar dan met een zwarte achtergrond. Voeg 1 BlackHole-object toe aan je sterrenhemel. Belangrijk: vlak rond een BlackHole kan zich geen ander hemelobject bevinden, want dat zouden opgezogen zijn door het zwarte gat! Zorg er dus voor dat er geen Star, Planet of Comet dichter dan 3 vakjes bij je zwarte gat is. De finale sterrenhemel ziet er dan bv zo uit:



- g. BONUS:** Schrijf een Move-methode in NightSky die heel de lucht 1 vakje naar rechts opschuift. Zorg dat de elementen die rechts uit de 2D-array vallen, links terug verschijnen. In de Main-methode schrijf je een loop, en in die loop roep je telkens de Move-methode op, waarna je Console.Clear() oproept (dit wist alle

inhoud van je Console), en vervolgens de Print-methode op NightSky. Gebruik de code `System.Threading.Thread.Sleep(1000)` om na elke Move-call 1 seconde (1000ms) te wachten. Hierdoor zul je een looping animatie krijgen van de sterrenhemel die verschuift in de Console. Dit zou er dan zo uit moeten zien:

<https://i.imgur.com/mzTr6Kh.gifv>

Advanced Oefeningen: Static

1. Voeg aan je `PersonalData` (of `Animal`) klasse een static teller variabele toe die telt hoeveel objecten er ooit van zijn aangemaakt. Zorg dat al je constructors deze variabele verhogen. Voeg ook een property toe die toelaat om de waarde hiervan aan de klasse op te vragen. Maak in je main 2 `PersonalData` (of `Animal`) objecten aan, 1 `StudentData` object (of 1 `Fish`) en 1 `TeacherData` (of 1 `Dog`). Vraag op en schrijf uit hoeveel objecten er zijn van `PersonalData` (`Animal`).
2. Maak je eigen static Math klasse, bvb `MyMath` met daarin wat static methodes. Gebruik gewoon in die methodes de klassieke math methodes, maar maak er bvb variaties op. Bvb je methode `sin` print de sinus uit ipv ze te returnen,... Bedenk 5 eigen methodes en voer ze uit vanuit je main.
3. Maak een Klasse `StaticTest` die een methode `Sum` heeft die 2 kommagetallen optelt en een tweede die 2 gehele getallen optelt. Beiden returnen het getal. Voer beiden uit vanuit de main.
4. Maak een Klasse `Inventory`. Deze heeft een static list van strings met elementen die erin zitten. Voeg static methodes toe om:
 - a. Een element aan de list toe te voegen
 - b. Een element uit de list te verwijderen
 - c. Vragen hoeveel elementen erin zitten
 - d. Alle elementen eruit te verwijderen
 - e. Een element eruit weer te geven via `console.WriteLine`, op basis van de index
 - f. De volledige list te tonen.
 - g. Te checken of een element erin zit en zo ja op welke index (geef -1 terug als het er niet inzit).

Advanced Oefeningen: Herhaling Ref en Out parameters

Oefeningen op out en ref parameter binding (zie later)

27. Herschrijf de vorige oefening, maar gebruik een tweede array parameter waarin het resultaat van de verdubbeling komt, geef deze door als out parameter.
28. Schrijf een methode FillArray, die een out parameter array opvult met de getallen van 1 t.e.m. de lengte van de array. Geef ook door hoe groot de array moet worden. (elk element van de array heeft na uitvoer van de methode dus de waarde van zijn index +1. Test door op te roepen vanuit je main en de array uit te schrijven. Test met verschillende arraygroottes.
29. Hermaak oefening 6,7,8 zodanig dat er geen return waarde gebruikt wordt, maar een tweede out parameter, waarin het resultaat zal worden opgeslagen vanuit de methode.
30. Schrijf een methode CircleDimensions, die bij een gegeven straal de oppervlakte en de omtrek van een cirkel berekent en opslaat in out parameters. De oppervlakte van een cirkel wordt gegeven door de formule $\text{Math.PI} * r * r$. Hoewel we ook een benadering als 3.14 zouden kunnen gebruiken, levert Math.PI een nauwkeuriger waarde op. De omtrek is $2 * \text{Math.PI} * r$. Test vanuit je main met meerdere waarden.
31. Herschrijf oefening 10 zodanig dat er aparte methodes zijn voor de omtrek en oppervlakte, die het resultaat retourneren. CircleDimensions gebruikt dan deze 2 methodes om zijn resultaten te berekenen en in zijn parameters het resultaat op te slaan.
32. Herschrijf vorige oefening zodanig dat er geen string geretourneerd wordt, maar dat het resultaat in de eerste parameter (die daardoor een ref parameter moet worden) wordt opgeslagen.