**Algemene opmerkingen (zeker lezen voor je start)**:

* Lees grondig de opgaven, en voer ze uit zoals vermeld. Een voorbeeldscreenshot van de output vind je onder de opgave.
* Het is niet omdat je één onderdeel niet volledig kan oplossen, dat andere onderdelen niet zullen werken of je er geen punten voor krijgt. **Zorg wel dat je onderdelen compileren/builden of iets simuleren zodat de rest van je opdracht wel werkt.** Je verliest punten als je project niet compileert.
* Schrijf altijd propere gestructureerde code. Vermijd dubbele code en pas voor alle code, en variabele- en methodenamen de aangeleerde regels toe (alles in het Engels, duidelijke namen, camel case,…). Voeg commentaar toe als je iets wilt verduidelijken aan de code.
* **Als er iets niet duidelijk is, vraag dan om verduidelijking van de aanwezige docent.**
* Afsluiten! Geef aan de docent/toezichter aan dat je wilt afsluiten en inleveren. Sla alles op en sluit je Visual Studio. Dan maak je een zipfile met je **familienaam.zip** als naam. Hierin plaats je **volledige map/folder met je solution, project en examencode**. PAS DAN MAG JE INTERNET INSCHAKELEN!!!
* Check nog eens of dit uw volledige solution en code bevat is voor en upload via Canvas.
* Check nog voor het verlaten van het lokaal je upload, check zeker dat dit ok is, indien niet, raadpleeg de toezichter/docent.

**SOLUTION EN OPSTELLING**

Maak een nieuwe solution aan met daarin een console applicatie project en geef als naam: je ***Achternaam22P4***, zonder spaties. Bijvoorbeeld voor de naam Pieter Jorissen zal de solution de volgende naam hebben: “Jorissen22P4”. Het project mag dezelfde naam hebben.

Ga naar je project en open de file Program.cs. Schrijf **boven** de Main methode het commentaar: *This is the exam made by [Voornaam] [Achternaam].*

**PixelArt.cs**

Maak een nieuwe klasse aan voor PixelArt

1. In dezelfde file maak je een enum ColorValue aan, deze bevat:
   * Red (waarde 1)
   * Green (waarde 2)
   * Blue (waarde 3)
   * Yellow (waarde 4)
2. Maak klasse variabelen:
   * *author* die een naam kan bevatten,
   * *height* en *width* variabelen die de hoogte en breedte van onze pixelart kunnen bevatten (geen kommagetallen)
   * Een lijst van ColorValue elementen genoemd pixelColors.

We willen geen van deze variabelen rechtstreeks toegang geven vanuit in andere klassen.

1. Maak een property voor de height die toegang geeft tot de variabele (opvragen en invullen), en bij het invullen controleert of er geen negatieve waarde wordt ingevuld. Als dit wel het geval is maak je deze positief. (Dus -8 wordt 8).
2. Maak een property voor de width die toegang geeft tot de variabele (opvragen en invullen), en bij het invullen controleert of er geen negatieve waarde wordt ingevuld. Als dit wel het geval is maak je deze positief. (Dus -8 wordt 8).
3. Maak een property voor de author en pixelColors, die rechtstreeks de variabelen invult/teruggeeft.
4. Maak een constructor die als parameter een auteur, een breedte en een hoogte als parameters meekrijgt.
   * Gebruik deze om **alle** klassevariabelen in te vullen / initialiseren.
   * Maak voor pixelColors een nieuwe lijst aan en vul die in met random waarden die niet minder kunnen zijn dan het minimum (Red) (en niet meer kunnen zijn dan het maximum (Yellow). Het aantal elementen is width x height?
5. Maak een methode in de PixelArt klasse *DrawPixel()* die 1 pixel kan tekenen in ons consolevenster in de juiste kleur. Hiervoor geef je een parameter mee van het type ColorValue.

* Maak een switch op basis van de ColorValue parameter
* Afhankelijk van de kleurwaarde hierin zet je de Console.ForegroundColor op een waarde die ermee overeenkomt uit consolecolor

(dus bvb als de parameter ColorValue.Red heeft, zet je Console.ForegroundColor = ConsoleColor.Red enz.. voor de andere kleuren).

Daarna print je een ‘\*’ gevolgd door een spatie naar het scherm (gebruik geen NewLine aan het einde).

1. Maak een methode *DrawPixelArt()* die de alle pixels uittekent in de juiste kleuren in de console.
   * Maak een dubbele lus waar je in de buitentste lus de hoogte doorloopt, en in de binnenste lus de breedte
   * Hou een teller bij om elementen uit de pixelColors lijst te halen (of gebruik een berekening)
   * Gebruik DrawPixel om de huidige pixel uit te tekenen naar het scherm (geef de huidige colorvalue uit de lijst mee als parameter)
   * Na het doorlopen van de hele breedte voeg je een NewLine toe (en start je op de volgende regel)
   * Zorg dat de afmetingen van hoogte en breedte juist zijn.
   * Vergeet niet om wanneer alle pixels getekend zijn de tekstkleur terug op wit te zetten.
2. Overschrijf de *ToString()* methode van de PixelArt klasse zodat deze de volgende string teruggeeft:  
     
   **“This HHH x BBB is made by CCC.”**  
     
   Waar je bij HHH de hoogte invult, bij BBB de breedte en bij CCC de naam van de auteur.

**Program.cs**

1. Maak een variabele om je voornaam in te vullen die je initialiseert met je voornaam en een variabele om je achternaam in te vullen die je initialiseert met je achternaam.
2. Maak een object aan van de klasse PixelArt. Voor de auteur geef je je naam *spatie* achternaam mee (deze waarden haal je uit de vorige 2 variabelen). Voor je hoogte geef je het aantal letters van je voornaam en voor de breedte het aantal letters van je achternaam (de lengtes vraag je ook op aan de variabelen uit het vorige puntje). Je mag spaties, als die in je naam voorkomen gewoon meetellen als letter.
3. Maak een lus waarin je de gebruiker vraagt wat die wil doen:
   1. Create,
   2. Show,
   3. Save,
   4. Load
   5. Quit
4. Je houdt bij wat je gebruiker kiest (Hoe? Wat vult de gebruiker in? Bijv. “C”?) en gaat op basis daarvan het volgende doen:

* **Create:** Schrijf “Creating new pixelart” naar de console. Vervang je PixeArt-object met een nieuw PixelArt-object met behulp van de constructor. Geef opnieuw je voor- en achternaam als parameter voor de auteur, het aantal letters in je voornaam als hoogte en het aantal letters in je achternaam als breedte.
* **Show:** Schrijf het PixelArt-object uit naar de console met behulp van de ToString() methode. Teken daarna de pixels uit in de console met behulp van de DrawPixelArt() methode.
* **Save:** Schrijf “Saving current pixelart to AAA” waar je bij AAA de fileName invult. Save je object als json in de debug folder van je solution via een methode SaveJSON die het PixelArt object meekrijgt als parameter. De filenaam is je voornaam.json
* **Load:** Schrijf “Loading pixelart from AAA” waar je bij AAA het de fileName invult. Laad je json file terug in in je pixelart object via een methode LoadJSON die als out parameter je PixelArt object meekrijgt en invult (lukt dit niet met een out parameter, gebruik dan een return).
* **Quit:** Stop de loop en sluit de applicatie (doe je geen quit, dan blijft dit loopen).  
    
  Zorg ervoor dat het niet uitmaakt of de gebruiker hoofdletters gebruikt of niet.

**Voorbeeld output (let op sterretjes zijn hier ruitjes, dit mag je negeren)**

**Text

Description automatically generated**