

A PROGRAMOZÁS ALAPJAI 2.

HÁZI FELADAT DOKUMENTÁCIÓ

FÉNYKÉP TORZÍTÁS MEGSZÜNTETÉSE

KÉSZÍTETTE: VARGA CSABA, K04JA8
v.csaba22@gmail.com

KÉSZÍTÉS FÉLÉVE: 2021/22/2

TARTALOMJEGYZÉK

Felhasználói dokumentáció	3
Osztályok statikus leírása.....	4
Image	4
Felelőssége.....	4
Attribútumok	4
Metódusok.....	4
BitMap	4
Felelőssége.....	4
Attribútumok	4
Metódusok.....	4
Egyéb függvények	5
Exception handling	5
UML osztálydiagramm	6
Összegzés	6
Mit sikerült és mit nem sikerült megvalósítani a specifikációból?	6
Mit tanultál a megvalósítás során?	6
Továbbfejlesztési lehetőségek.....	6
Képernyőképek a futó alkalmazásról.....	7

Felhasználói dokumentáció

A program 24-bites padding nélküli(4-gyel osztható a kép szélessége) BMP képek torzítását szünteti meg.

A program a standard input/output-on keresztül kommunikál a felhasználóval. Először bekéri tőle az általa módosítani kívánt kép elérési útvonalát. Itt fájlkiterjesztéssel együtt kell megadni az adott fájl nevét, elérési útvonalát. Utána a program kér a felhasználótól egy horizont vonalat, ami meghatároz 2 téglalapot és a nagyobb téglalapra lesz rátranszformálva a kész kép. A horizont vonal értéke nem lehet negatív, illetve nem lehet nagyobb, mint a kép magassága. A program még bekér négy koordinátát, amiknek formátuma (x koordináta,y koordináta) PL: (45,76)), amely koordináták által meghatározott négyszögön fogja elvégezni a transzformációt.

```
Enter the filename: text.bmp
Enter the horizon line: 282
format: (x,y)
Coords of Top-left corner: (61,73)
Coords of Bottom-left corner: (61,377)
Coords of Bottom-right corner: (394,295)
Coords of Top-right corner: (394,237)
```

Végül a program kér a felhasználótól egy fájlnevet (kiterjesztés nélkül), ahova el lesz mentve a torzításmentes kép, majd megkérdezi a felhasználót, hogy szeretne-e egy újabb képet torzítás mentesíteni.

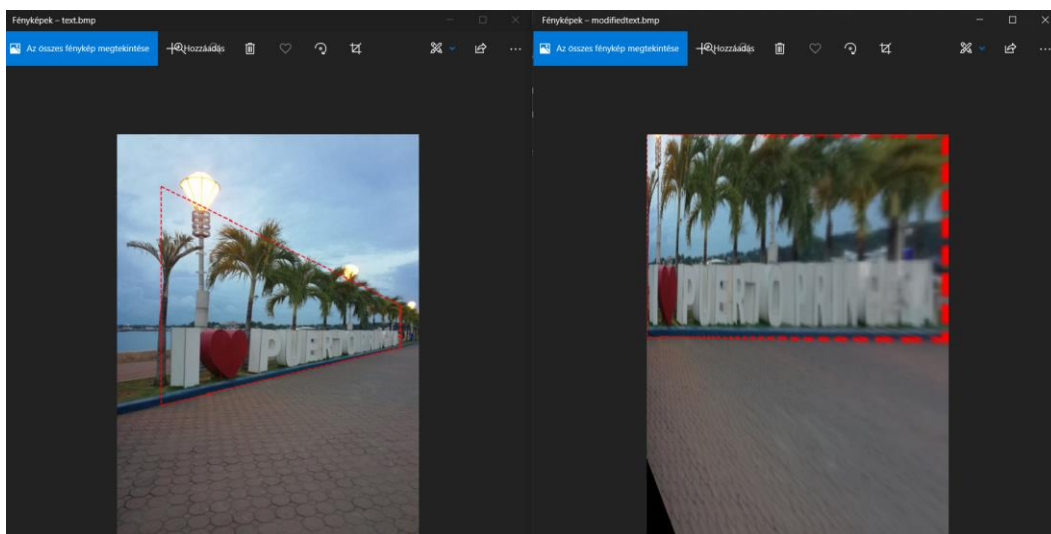
```
Enter output filename(without extension): modifiedtext
Image transformation was successfull
Do you want to transform a new image? (Y/N)
```

Az esetleges hibaüzenetek esetén a felhasználó „Y” karakter segítségével újra tudja indítani a programot, bármely más karakter bevitele eséréen pedig kilép belőle.

Példa egy ilyen hibaüzenetre:

```
File could not be opened
Do you want to try again? (Y - to try again | anything else to exit)
```

Példa a program működésére (bal – eredeti, jobb – módosított):



1. ábra text.bmp és textout.bmp

Osztályok statikus leírása

Image

Felelőssége

Absztrakt interfész őszosztály.

Különféle képformátumok osztályait lehet belőle örököltetni.

Attribútumok

Az Image osztálynak nincsenek tagváltozói.

Metódusok

Privát

virtual void writeImage(const char* path) = 0;

virtual void readImage() = 0;

A writeImage és a readImage függvényeket a leszármazottakban kell definiálni.

BitMap

Felelőssége

.bmp kiterjesztésű fájlok beolvasása, írása, tárolása, módosítása.

Attribútumok

Privát

unsigned char* info – A bmp fájlok file header-jét és info header-ját tárolja

cv::Mat bMap – OpenCV-ben definiált Mat (mátrix) típusú változó. A fájl bittérképét tárolja.

cv::Mat modifiedBMAP – OpenCV-ben definiált Mat (mátrix) típusú változó. A módosított bittérképet tárolja.

const char* fname – A beolvasott fájl nevét tárolja. Const mivel a konstruktor meghívásakor beállítódik és utána nem szabad megváltoztatni.

Metódusok

Public

BitMap(const char* path) – Konstruktor ami az fname tagváltozót beállítja a path értékére és minden más tagváltozónak default értéket ad.

BitMap(const BitMap& other) – Másoló konstruktor

~BitMap() – destruktork

void setInfo(unsigned char* param) – az info tagváltozót beállítja a paraméterként kapott param értékére

void setBMap(cv::Mat param) – bMap tagváltozót beállítja a paraméterként kapott param értékére

void setModifiedBMap(cv::Mat param) – a modifiedBMap értéket beállítja a paraméterként kapott param értékére

void setWidth(int param) – az info header 18-19-20-21-es indexű változóit módosítja. A paraméterként kapott param értéket jobbra kell bitshiftelni, mivel az info header típusa unsigned char és a param típusa pedig int.

void setHeight(int param) – az info header 22-23-24-25-ös indexű változóit módosítja. A paraméterként kapott param értékét jobbra kell bitshiftelni, mivel az info header típusa unsigned char és a param típusa pedig int.

unsigned char* getInfo() const – visszaadja az info tagváltozót. Konstans mert nem módosítja egyik tagváltozó értékét sem.

cv::Mat getBMap() const – visszaadja a bMap tagváltozót. Konstans, mert nem módosítja egyik tagváltozó értékét sem.

cv::Mat getModifiedBMap() const – visszaadja a modifiedBMap tagváltozót. Konstans, mert nem módosítja egyik tagváltozó értékét sem.

const char* getFname() const – visszaadja az fname tagváltozó értékét. Konstans, mert nem módosítja egyik tagváltozó értékét sem.

int getHeight() const – Kiolvassa az info tagváltozóból a magassági adatot(22-es indexű helyről egy int méretű adatot) és visszaadja azt. Konstans, mert nem módosítja egyik tagváltozó értékét sem.

int getWidth() const – Kiolvassa az info tagváltozóból a szélességi adatot(18-es indexű helyről egy int méretű adatot) és visszaadja azt. Konstans, mert nem módosítja egyik tagváltozó értékét sem.

void ReadImage() – Az absztrakt őssztályban deklarált virtuális függvény definiálása. Beolvassa az fname tagváltozó által jelölt fájlt és eltárolja azt a megfelelő tagváltozóiban. Ha nem sikerül megnyitni a fájlt akkor exception-t dob („File could not be opened”)

void writeImage(const char* name) – Az absztrakt őssztályban deklarált virtuális függvény definiálása. A name paraméter nevével létrehoz egy fájl .bmp kiterjesztéssel a program mappájában. Az osztály adott példányának a tagváltozóiban eltárolt adatokat kiírja a létrehozott fájlba. Ha nem sikerül létrehozni a fájlt akkor exception-t dob („Destination file could not be opened/created”)

Egyéb függvények

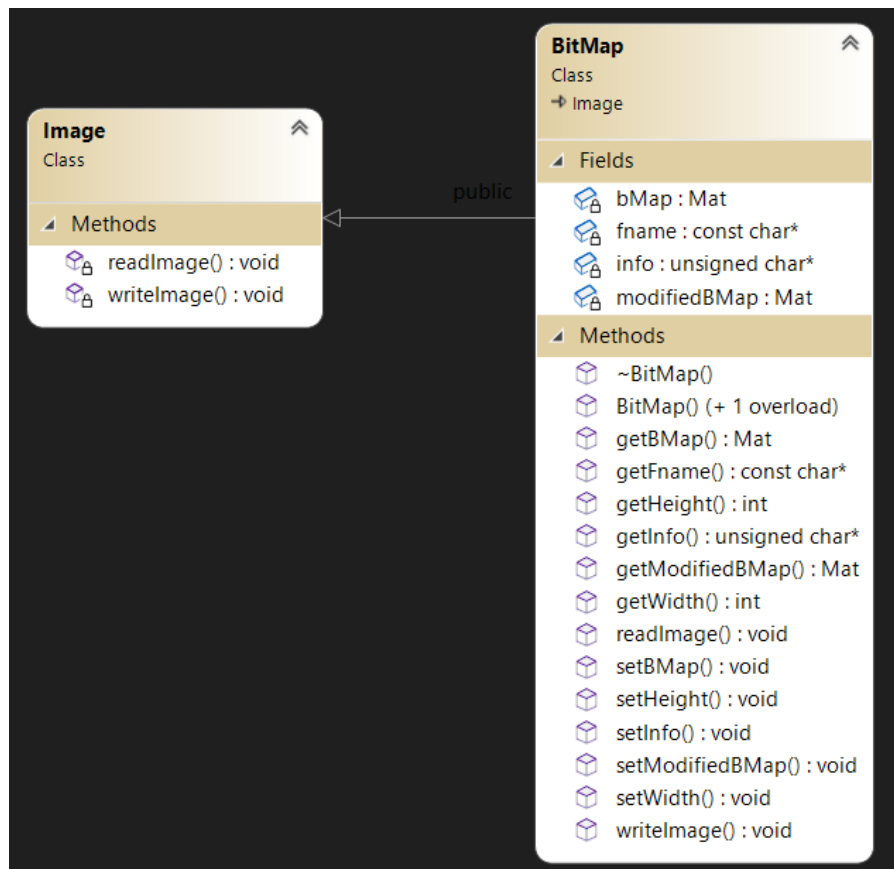
std::vector<cv::Point2f> getSourcePoints(Bitmap bmp) – visszatér egy std::vector típusú vektorral ami cv::Point2f típusú pontokat tárol. Paraméterként kap egy Bitmap osztálypéldányt. Bekér a felhasználótól 4 darab koordinátát a következő sorrend alapján: bal felső, bal alsó, jobb alsó, jobb felső. A koordináták formátuma a következő: (x,y), ahol x és y egy nullánál nagyobb, de a kép méreteinél kisebb értékek. Exceptiont dob ha a formátum nem megfelelő(„Invalid input format”), ha az X,Y koordináta negatív(„Invalid Input. X coordinate is negative”), illetve ha X,Y nagyobb mint a kép szélessége, magassága („Invalid Input. Y coordinate is greater than the width of the picture”)

std::vector<cv::Point2f> findDestination(int horizonLine, Bitmap bmp) - visszatér egy std::vector típusú vektorral ami cv::Point2f típusú pontokat tárol. Paraméterként kap egy Bitmap osztálypéldányt, illetve egy horizont vonalat. A függvény meghatározza a horizont vonallal párhuzamos legnagyobb területű téglalapot (ide fog transzformálódni a kép) és ennek a téglalapnak a négy csúcsával tér vissza. (csúcsok sorrendje: bal felső, bal alsó, jobb alsó, jobb felső)

Exception handling

A dobott exception-öket a main függvényen belül kezelem. Amennyiben valamelyik exception-t elkapom, ezt tudatom a felhasználóval és megkérdezem tőle, hogy szeretne-e újra próbálkozni az adatok bevitelével. Ha nemmel válaszol akkor a main visszatér, a program leáll.

UML osztálydiagramm



Felhasznált külső libraryk

OpenCV – <https://opencv.org/>

Windows.h

Összegzés

Mit sikerült és mit nem sikerült megvalósítani a specifikációból?

A specifikációban foglaltak sikerült teljesítenem.

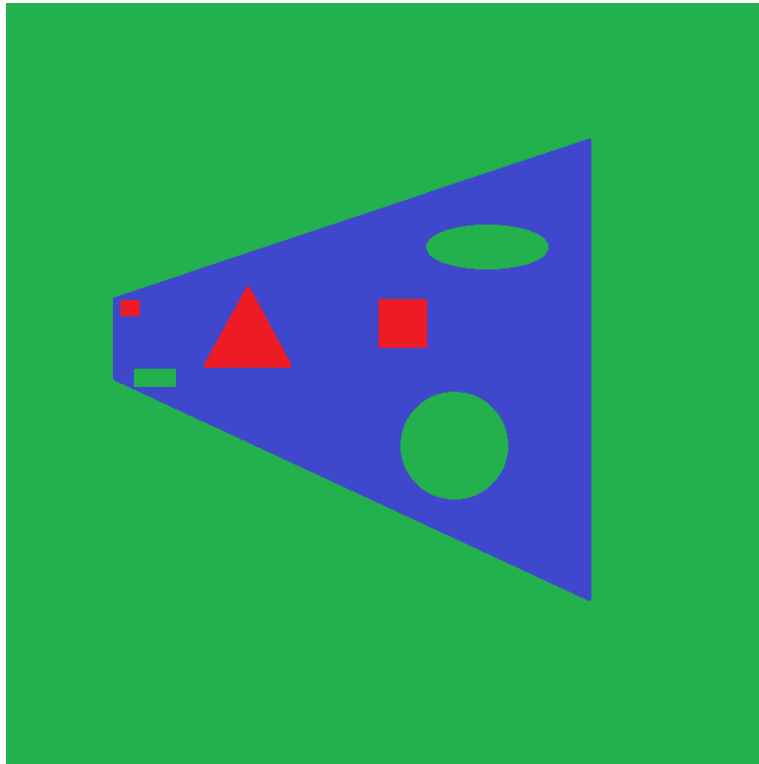
Mit tanultál a megvalósítás során?

Megtanultam a .bmp kiterjesztésű fájlok hexa kódját értelmezni és megismertem a fájl felépítését. Az exception handling eleinte nehézkes volt.

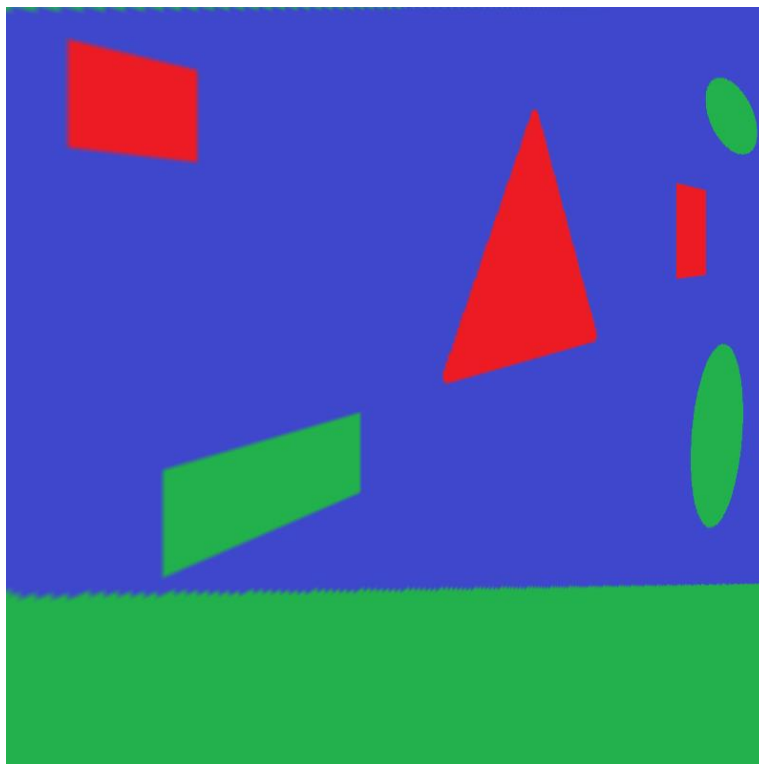
Továbbfejlesztési lehetőségek

A programot lehetne bővíteni a támogatott képformátumok terén. Újabb képmódosítási opciókat lehetne megvalósítani.

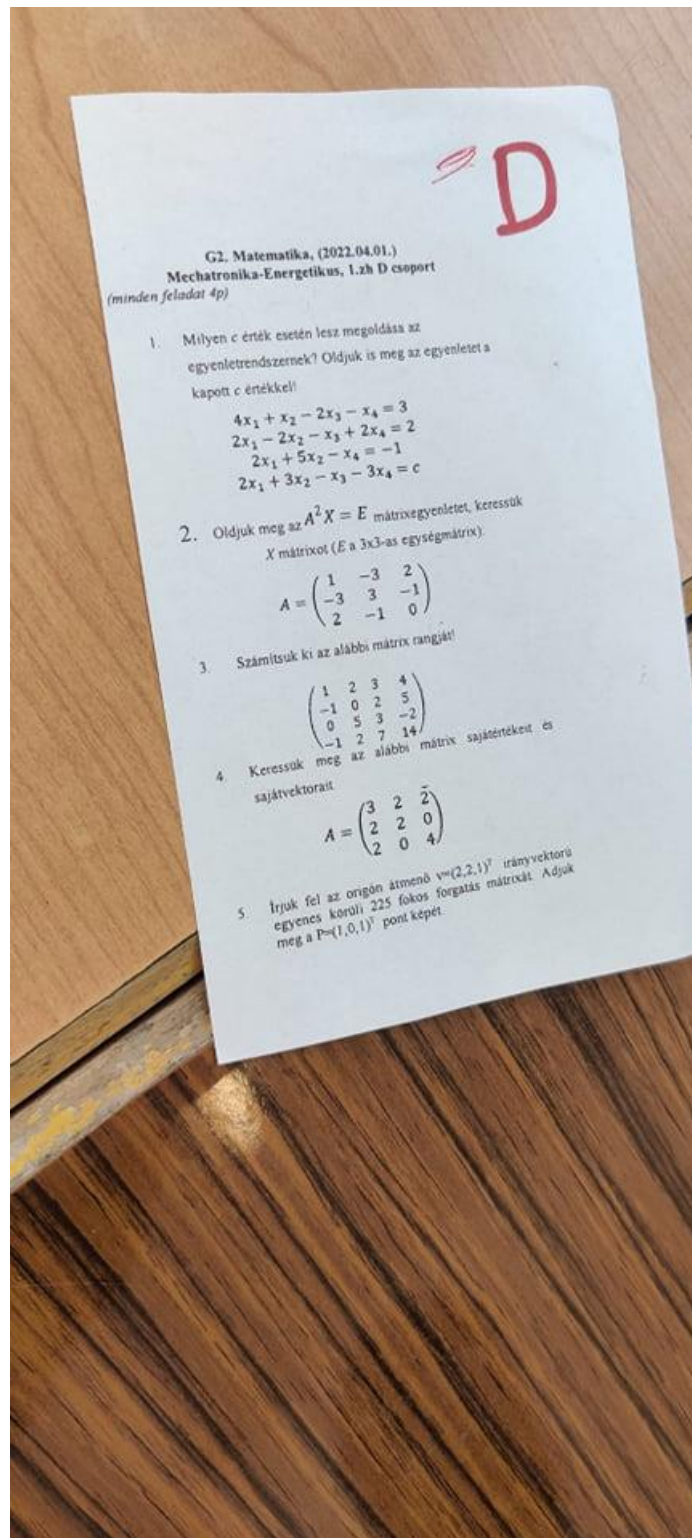
Képernyőképek a futó alkalmazásról



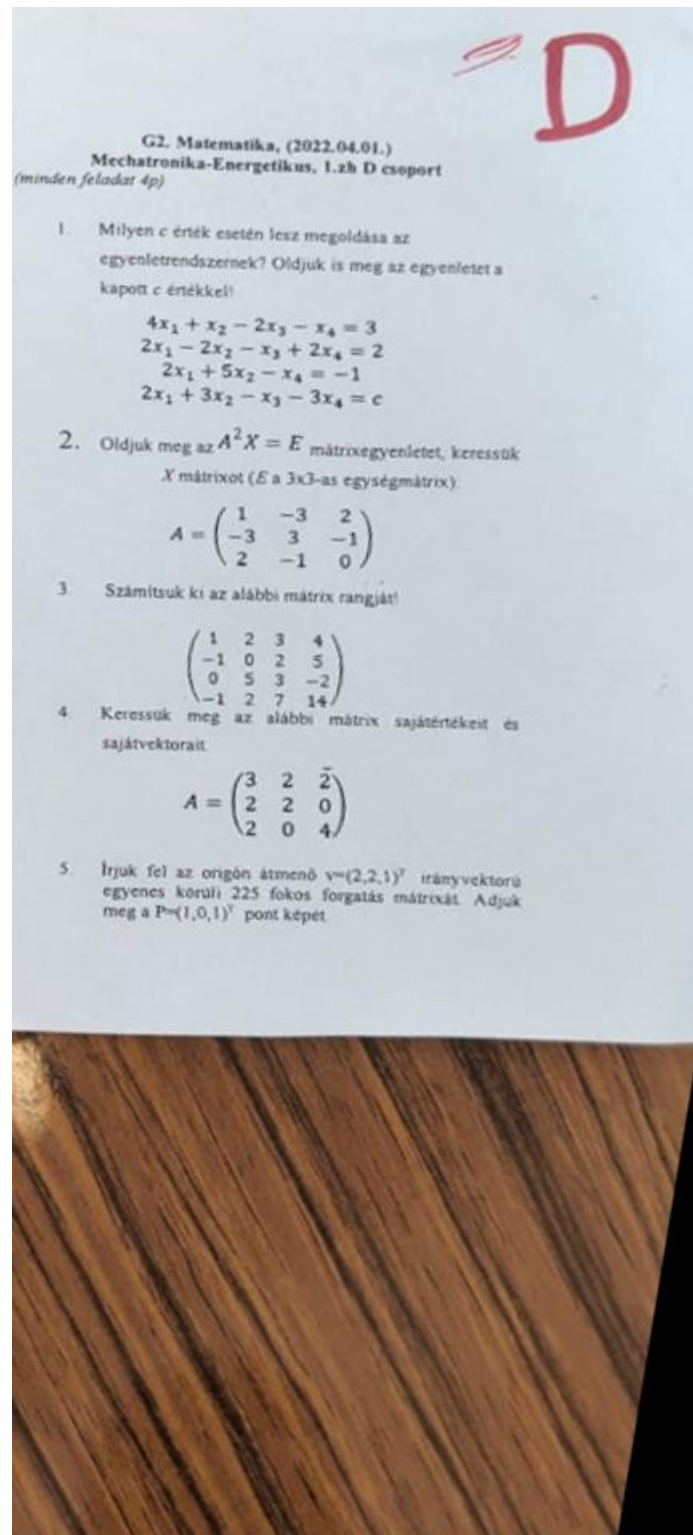
2. ábra bmptestgreen.bmp



3. ábra bmptestgreenout.bmp



4. ábra zh.bmp



5. ábra zhout.bmp