

A PROGRAMOZÁS ALAPJAI 2
(BMEV8IAA03, 2024/25/2)
NAGY HÁZI FELADAT

Polinom függvények ábrázolása ASCII art-al

készítette:

Ferencz Péter

(RFG7SN)



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Mérnök-informatikus Bsc

2025 Március-Április-Május

1. fejezet

Tartalomjegyzék

1. Tartalomjegyzék	i
2. Felhasználói dokumentáció	1
2.1. A program célja	1
2.2. Megkötések	1
2.3. A program által elfogadott kapcsolók	1
3. Fejlesztői dokumentáció	3
3.1. A Console statikus osztály	3
3.2. A Command osztálycsoport	3
3.3. A Screen osztálycsoport	3
3.4. A grafikonok kirajzolása	3
3.5. A program	4
3.6. A program fejlesztéséhez használt más programok	5
4. Osztályok dokumentációja	7
4.1. AlreadyInitializedConsoleException osztályreferencia	7
4.1.1. Részletes leírás	7
4.2. Command osztályreferencia	7
4.2.1. Részletes leírás	8
4.2.2. Konstruktork és destruktorok dokumentációja	8
4.2.2.1. Command()	8
4.2.3. Tagfüggvények dokumentációja	8
4.2.3.1. exec()	8
4.2.4. Adattagok dokumentációja	9
4.2.4.1. paramCount	9
4.3. CommandManager osztályreferencia	9
4.3.1. Részletes leírás	9
4.3.2. Tagfüggvények dokumentációja	9
4.3.2.1. CaptureInput()	9

4.3.2.2.	<code>parseCLA()</code>	10
4.3.2.3.	<code>parseCommand()</code>	10
4.4.	Console osztályreferencia	11
4.4.1.	Részletes leírás	12
4.4.2.	Tagfüggvények dokumentációja	12
4.4.2.1.	<code>Clear()</code> [1/2]	12
4.4.2.2.	<code>Clear()</code> [2/2]	12
4.4.2.3.	<code>destroy()</code>	13
4.4.2.4.	<code>flush()</code>	13
4.4.2.5.	<code>getInputOfLength()</code>	13
4.4.2.6.	<code>init()</code>	14
4.4.2.7.	<code>moveCursor()</code>	14
4.4.2.8.	<code>Print()</code> [1/2]	15
4.4.2.9.	<code>Print()</code> [2/2]	15
4.4.2.10.	<code>RedirectOutput()</code>	16
4.4.2.11.	<code>setColor()</code>	16
4.5.	Expression osztályreferencia	16
4.5.1.	Részletes leírás	17
4.5.2.	Konstruktorok és destruktorok dokumentációja	17
4.5.2.1.	<code>Expression()</code>	17
4.5.3.	Tagfüggvények dokumentációja	17
4.5.3.1.	<code>getCoefficients()</code>	17
4.5.3.2.	<code>set()</code>	18
4.5.3.3.	<code>valueAt()</code>	18
4.6.	Front osztályreferencia	18
4.6.1.	Részletes leírás	19
4.6.2.	Tagfüggvények dokumentációja	19
4.6.2.1.	<code>Draw()</code>	19
4.6.2.2.	<code>getInput()</code>	19
4.7.	InfoCommand osztályreferencia	19
4.7.1.	Részletes leírás	20
4.7.2.	Tagfüggvények dokumentációja	20
4.7.2.1.	<code>exec()</code>	20
4.8.	InfoScreen osztályreferencia	20
4.8.1.	Részletes leírás	21
4.8.2.	Tagfüggvények dokumentációja	21
4.8.2.1.	<code>Draw()</code>	21
4.9.	InvalidSizeConsoleException osztályreferencia	21
4.9.1.	Részletes leírás	21
4.10.	InvalidStyleException osztályreferencia	21
4.10.1.	Részletes leírás	22
4.11.	MoveCommand osztályreferencia	22

4.11.1. Részletes leírás	22
4.11.2. Tagfüggvények dokumentációja	22
4.11.2.1. exec()	22
4.12. OffsetCommand osztályreferencia	23
4.12.1. Részletes leírás	23
4.12.2. Tagfüggvények dokumentációja	23
4.12.2.1. exec()	23
4.13. OutCommand osztályreferencia	24
4.13.1. Részletes leírás	24
4.13.2. Tagfüggvények dokumentációja	24
4.13.2.1. exec()	24
4.14. PlotCommand osztályreferencia	25
4.14.1. Részletes leírás	25
4.14.2. Tagfüggvények dokumentációja	25
4.14.2.1. exec()	25
4.15. Plotter osztályreferencia	26
4.15.1. Részletes leírás	26
4.15.2. Tagfüggvények dokumentációja	26
4.15.2.1. Draw()	26
4.16. PlotterScreen osztályreferencia	27
4.16.1. Részletes leírás	27
4.16.2. Tagfüggvények dokumentációja	27
4.16.2.1. Draw()	27
4.16.2.2. Move()	28
4.16.2.3. Offset()	28
4.16.2.4. Scale()	28
4.17. Rect< T > struktúrasablon-referencia	29
4.17.1. Részletes leírás	29
4.18. ScaleCommand osztályreferencia	29
4.18.1. Részletes leírás	30
4.18.2. Tagfüggvények dokumentációja	30
4.18.2.1. exec()	30
4.19. Screen osztályreferencia	31
4.19.1. Részletes leírás	31
4.19.2. Tagfüggvények dokumentációja	31
4.19.2.1. Draw()	31
4.20. StyleCommand osztályreferencia	31
4.20.1. Részletes leírás	32
4.20.2. Tagfüggvények dokumentációja	32
4.20.2.1. exec()	32
4.21. TUICommand osztályreferencia	32
4.21.1. Részletes leírás	33

4.21.2. Tagfüggvények dokumentációja	33
4.21.2.1. exec()	33
4.22. UnableToOpenFileException osztályreferencia	33
4.22.1. Részletes leírás	33
4.23. UninitializedConsoleException osztályreferencia	33
4.23.1. Részletes leírás	33
4.24. UnknownCommandException osztályreferencia	34
4.24.1. Részletes leírás	34
5. Jog	35

2. fejezet

Felhasználói dokumentáció

2.1. A program célja

A felhasználó polinom függvényeket ad meg a programnak, mely annak grafikonját ascii art formájában vagy egy fájlba menti, vagy interaktív módban megjeleníti a konzolon.

2.2. Megkötések

A program csak polinom függvényeket tud ábrázolni.

2.3. A program által elfogadott kapcsolók

A programot az alábbi sematika szerint kell futtatni:

```
plottr [kapcsolók] <függvény>
```

A kapcsolókat tetszőleges sorrendben, opcionálisan megadhatók.

Név	Flag	Rövidítés	Magyarázat
Interaktív mód	-tui	-t	Interaktív módba állítja a programot
Mozgatás	-move <x:num> <y:num>	-m	Adott x, y koordinátákra mozgatja a nézetet
Elmozdulás	-offset <x:num> <y:num>	-o	A jelenlegi nézettől mérten relatívan mozog
Méretarány	-scale <w:num> <h:num>	-s	A kijelző szélességét és magasságát adjuk meg
Információ kijelző	-info	-i	A függvényről jelenít meg információkat
Függvény megadása	-plot <exp...:number>	-p	A függvény együtthatóit lehet beállítani
Stílusok	-style <basic ascii>	-s	A függvény megjelenítését állíthatjuk
Mentés	-out <f:file>	-o	Kiment a függvényt fájlba

2.1. táblázat A program által elfogadott kapcsolók

3. fejezet

Fejlesztői dokumentáció

3.1. A Console statikus osztály

A *Console* osztály fő feladata az interaktív módban történő felhasználói bemenet és kimenet kezelésére szolgáló *ncurses* könyvtárral való kommunikáció lebonyolítására szolgáló statikus osztály. Ezt az osztályt az *init* és a *destroy* függvényekkel kell inicializálni és dekonstruálni a program végén.

3.2. A Command osztálycsoport

Az interaktív mód során kapott parancsokat és parancssori futtatás által adott kapcsolókat a *ConsoleManager* osztály kezeli, amely a konstruktorában kapott *Command* absztrak osztály leszármazottait tartalmazza és hívja meg azok *exec* függvényeit a megadott paraméterekkel.

3.3. A Screen osztálycsoport

Az interaktív mód során kapott parancsokat a program a *Front* osztályon keresztül fogja megkapni, mely a különböző *Screen*-ek váltogatását kezeli.

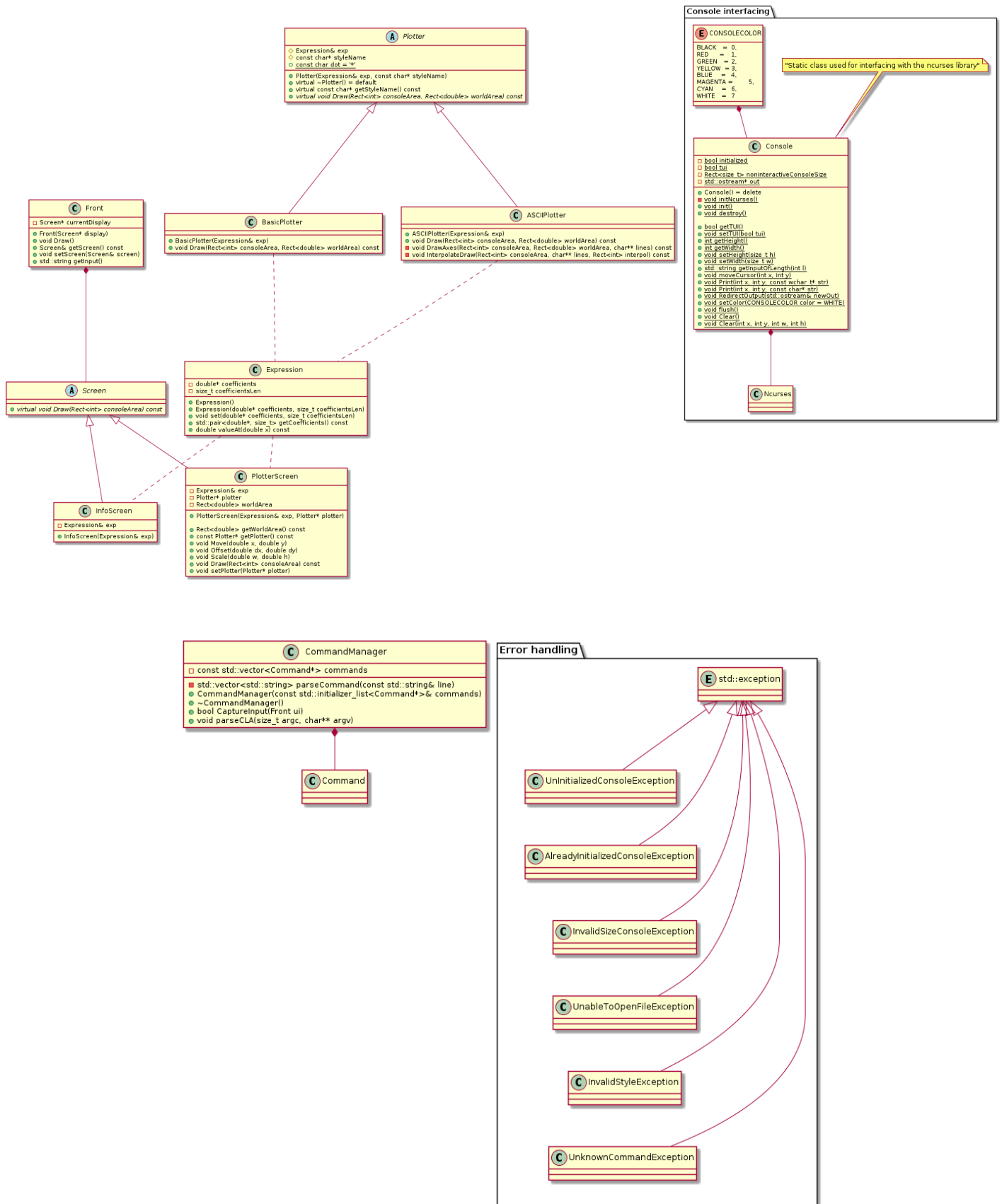
3.4. A grafikonok kirajzolása

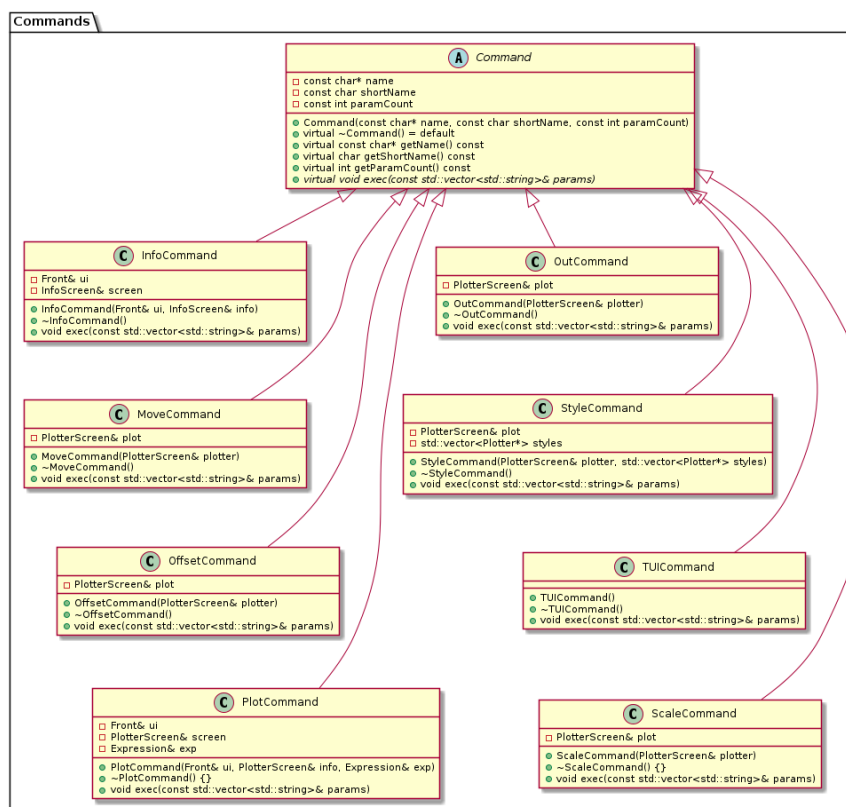
A grafikonok kirajzolását a *Plotter* osztály leszármazottjai teszik, akik különböző módon ábrázolnak:

SimplePlotter: Ez az osztály a képernyő minden helyéhez hozzárendel egy, a függvénynek megfelelt *y* értéket.

ASCIIPlotter: A grafikon kirajzolása csak ASCII karakterekkel történik, ezen felül pontok helyett vonalakkal dolgozik.

3.5. A program





3.6. A program fejlesztéséhez használt más programok

- Projektgenerátor: [premake5](#)
- Cpp fordító: [GNU Compiler Collection \(g++\)](#)
- Fordítás: [GNU Make](#)
- Dokumentáció: [Doxygen](#)
- Verziókövetés: [Git](#) és [Github](#)

A programot helyes működésének elérése érdekében folyamatos, fejlesztés közbeni manuális tesztet, ezen felül a *gtest lite* tesztkönyvtárat, és a *memtrace* memóriaszivárgás-ellenőrző programot alkalmazzuk.

4. fejezet

Osztályok dokumentációja

4.1. AlreadyInitializedConsoleException osztályreferencia

Akkor dobjuk, ha a program már inicializálta a konzolt, de azt még egyszer megtenné a program.

```
#include <Console.hpp>
```

4.1.1. Részletes leírás

Akkor dobjuk, ha a program már inicializálta a konzolt, de azt még egyszer megtenné a program.

4.2. Command osztályreferencia

A felhasználó által adott parancsok őssztálya. Ez az osztály az összes parancs alapja, amelyet a felhasználó futtathat. A parancsokat az osztály név és rövid név alapján azonosítjuk. Az osztály az argumentumok számát is tárolja, és meghatározza, hogy hány paramétert fogad el egy parancs. A parancsokat egy `exec` virtuális metódus segítségével lehet végrehajtani.

```
#include <Command.hpp>
```

Publikus tagfüggvények

- `Command` (const char *`name`, const char `shortName`, const int `paramCount`)
Konstruktor, amely beállítja a parancs nevét, rövid nevét és paraméterek számát.
- virtual void `exec` (const std::vector< std::string > ¶ms)=0
Végrehajtja a parancsot.

Privát attribútumok

- const char * `name`
A név, amellyel a felhasználó kiválasztja melyik parancsot futtatja.
- const char `shortName`
Olyan mint `name`, csak egy karakterrel.
- const int `paramCount`
A parancsok adható paraméterek száma.

4.2.1. Részletes leírás

A felhasználó által adott parancsok őssztálya. Ez az osztály az összes parancs alapja, amelyet a felhasználó futtathat. A parancsokat az osztály név és rövid név alapján azonosítjuk. Az osztály az argumentumok számát is tárolja, és meghatározza, hogy hány paramétert fogad el egy parancs. A parancsokat egy `exec` virtuális metódus segítségével lehet végrehajtani.

4.2.2. Konstruktorok és destruktorok dokumentációja

4.2.2.1. Command()

```
Command::Command (
    const char * name,
    const char shortName,
    const int paramCount ) [inline]
```

Konstruktor, amely beállítja a parancs nevét, rövid nevét és paraméterek számát.

Paraméterek

<i>name</i>	A parancs neve
<i>shortName</i>	A parancs rövid neve
<i>paramCount</i>	A parancsnak adható paraméterek száma (-1 ha bármennyi paramétert elfogad)

4.2.3. Tagfüggvények dokumentációja

4.2.3.1. exec()

```
virtual void Command::exec (
    const std::vector< std::string > & params ) [pure virtual]
```

Végrehajtja a parancsot.

Paraméterek

<i>params</i>	A felhasználó által a parancsnak adott paraméterek
---------------	--

Megvalósítják a következők: [TUICommand](#), [StyleCommand](#), [ScaleCommand](#), [PlotCommand](#), [OutCommand](#), [OffsetCommand](#), [MoveCommand](#) és [InfoCommand](#).

4.2.4. Adattagok dokumentációja

4.2.4.1. paramCount

```
const int Command::paramCount [private]
```

A parancsnak adható paraméterek száma.

Megjegyzés

Ha -1, akkor bármennyi paraméterrel hívható

4.3. CommandManager osztályreferencia

A [Command](#) osztályok kezelése, és a felhasználó által megadott parancsok végrehajtását szolgáló osztály. A [CommandManager](#) osztály felelős a parancsok tárolásáért és kezeléséért. A parancsokat egy heterogén kollekcióban tárolja, és biztosítja azok végrehajtását a felhasználó által adott bemeneti utasítások alapján.

```
#include <CommandManager.hpp>
```

Publikus tagfüggvények

- bool [CaptureInput](#) ([Front ui](#))
Parancsot kér a felhasználótól, melyet ha tud futtat is.
- void [parseCLA](#) (size_t argc, char **argv)
Parancssori kapcsolókat dolgoz fel, melyeket a bekérés sorrendjében futtat is.

Privát tagfüggvények

- std::vector< std::string > [parseCommand](#) (const std::string &line)
Adott sort tördel fel, és adja vissza azokat egy sztring vektorban.

Privát attribútumok

- const std::vector< [Command](#) * > [commands](#)
[Command](#)-ok heterogén kollekciója, melyet elfogadunk

4.3.1. Részletes leírás

A [Command](#) osztályok kezelése, és a felhasználó által megadott parancsok végrehajtását szolgáló osztály. A [CommandManager](#) osztály felelős a parancsok tárolásáért és kezeléséért. A parancsokat egy heterogén kollekcióban tárolja, és biztosítja azok végrehajtását a felhasználó által adott bemeneti utasítások alapján.

4.3.2. Tagfüggvények dokumentációja

4.3.2.1. CaptureInput()

```
bool CommandManager::CaptureInput (  
    Front ui )
```

Parancsot kér a felhasználótól, melyet ha tud futtat is.

Paraméterek

<i>ui</i>	Az adott felület, ahonnan a parancsot kéri
-----------	--

Visszatérési érték

Megmondja, hogy a felhasználó ki akar-e lépni a programból (false: igen, true: nem)

Kivételek

<i>UnknownCommandException</i> , <i>ha</i>	a <code>commands</code> nem tartalmazza a parancsot.
--	--

4.3.2.2. parseCLA()

```
void CommandManager::parseCLA (
    size_t argc,
    char ** argv )
```

Parancssori kapcsolókat dolgoz fel, melyeket a bekérés sorrendjében futtat is.

Paraméterek

<i>argc</i>	A <code>argv</code> elemeinek száma
<i>argv</i>	C-sztringek a kapcsolókra

Kivételek

<i>UnknownCommandException</i> , <i>ha</i>	a <code>commands</code> nem tartalmazza a parancsot.
--	--

4.3.2.3. parseCommand()

```
std::vector< std::string > CommandManager::parseCommand (
    const std::string & line ) [private]
```

Adott sort tördel fel, és adja vissza azokat egy sztring vektorban.

Paraméterek

<i>line</i>	A feltördelendő sor
-------------	---------------------

Visszatérési érték

A feltördelt sor

4.4. Console osztályreferencia

A szabványos kimenetre írást felügyelő osztály. A program semelyik más része nem írhat és olvashat a konzolról. Ncurses könyvtárbeli függvényeket is csak ez az osztály hívhat meg.

```
#include <Console.hpp>
```

Statikus publikus tagfüggvények

- static void **init** ()
Inicializálja a konzolt. Ez a függvény előkészíti a konzolt a használatra. Ha a program interaktív módban fut, akkor meghívja az ncurses függvényeit.
- static void **destroy** ()
A konzol erőforrásainak felszabadítása. Ez a függvény befejezi a konzolhasználatot, törli a képernyőt és lezárja a terminál kezelést (ncurses könyvtárból visszatér).
- static std::string **getInputOfLength** (int l)
Bekér egy adott hosszúságú szöveget a felhasználótól. Ez a függvény legfeljebb l hosszúságú bemenetet olvas be a felhasználótól. TUI módban az ncurses getch függvényét használja, egyébként a szabványos bemenetről olvas std::getline segítségével.
- static void **moveCursor** (int x, int y)
A kurzor mozgatása a megadott (x, y) pozícióra. A függvény a kurzort áthelyezi a megadott koordinátákra, ha a konzol inicializálva van.
- static void **Print** (int x, int y, const wchar_t *str)
Szöveg kiírása a megadott pozícióra. A megadott szöveget kiírja a (x, y) koordinátára TUI módban. Ha nem TUI módban van, akkor a szöveget a kimeneti adatfolyamba írja új sorral.
- static void **Print** (int x, int y, const char *str)
Szöveg kiírása a megadott pozícióra. A megadott szöveget kiírja a (x, y) koordinátára TUI módban. Ha nem TUI módban van, akkor a szöveget a kimeneti adatfolyamba írja új sorral.
- static void **RedirectOutput** (std::ostream &newOut)
Átírányítja a kimenetet egy új adatfolyamra. Ezzel a funkcióval beállítható egy új kimeneti adatfolyam. Az összes további kimenet az új adatfolyamba lesz irányítva.
- static void **setColor** (CONSOLECOLOR color=WHITE)
Beállítja a színt a konzolon. A függvény beállítja a konzol színét a megadott color értékre. Ha a terminál nem támogatja a színeket, a függvény nem hajt végre semmilyen műveletet.
- static void **flush** ()
Kiüríti a kimenetet, frissíti a terminált. A függvény frissíti a konzolt, ha a tui mód engedélyezve van.
- static void **Clear** ()
Törli a képernyőt. A függvény törli a képernyőt, ha a tui mód engedélyezve van.
- static void **Clear** (int x, int y, int w, int h)
Törli a kijelölt területet a képernyőn. A függvény törli a képernyőn az adott (x, y) pozíciótól kezdődően a megadott szélességű (w) és magasságú (h) területet.

Statikus privát tagfüggvények

- static void **initNcurses** ()
Inicializálja a Ncurses könyvtárat. A függvény inicializálja az Ncurses könyvtárat, beállítja a billentyűzet kezelést, a színeket és a kimeneti módot. Továbbá, beállítja az alapértelmezett színpárokat, ha a terminál támogatja a színeket. A konzol ezután törlődik.

Statikus privát attribútumok

- static bool `initialized` = false
Inicializáltuk-e a konzolt az `init` fv. meghívásával.
- static bool `tui` = false
A program interaktív módban fut-e. (használja-e az ncurses könyvtárat)
- static `Rect< size_t > noninteractiveConsoleSize` = {0, 0, 20, 10}
Ha a program nem interaktív módban fut, megmondja hogy mekkora területre írhat ki.
- static `std::ostream * out` = `&std::cout`
A kimenet, melyre az összes függvény ír.

4.4.1. Részletes leírás

A szabványos kimenetre írást felügyelő osztály. A program semelyik más része nem írhat és olvashat a konzolról. Ncurses könyvtárbeli függvényeket is csak ez az osztály hívhat meg.

Kivételek

<code>std::exception, ha</code>	példányosítani próbáljuk
---------------------------------	--------------------------

4.4.2. Tagfüggvények dokumentációja

4.4.2.1. `Clear()` [1/2]

```
void Console::Clear ( ) [static]
```

Törli a képernyőt. A függvény törli a képernyőt, ha a `tui` mód engedélyezve van.

Kivételek

<code>UninitializedConsoleException</code>	Ha a konzol nincs inicializálva
--	---------------------------------

4.4.2.2. `Clear()` [2/2]

```
void Console::Clear (
    int x,
    int y,
    int w,
    int h ) [static]
```

Törli a kijelölt területet a képernyőn. A függvény törli a képernyőn az adott (x, y) pozíciótól kezdődően a megadott szélességű (w) és magasságú (h) területet.

Paraméterek

<i>x</i>	A terület bal felső sarkának vízszintes koordinátája.
<i>y</i>	A terület bal felső sarkának függőleges koordinátája.
<i>w</i>	A törlendő terület szélessége.
<i>h</i>	A törlendő terület magassága.

Kivételek

<i>UninitializedConsoleException</i>	Ha a konzol nincs inicializálva
--	---------------------------------

4.4.2.3. **destroy()**

```
void Console::destroy ( ) [static]
```

A konzol erőforrásainak felszabadítása. Ez a függvény befejezi a konzolhasználatot, törli a képernyőt és lezárja a terminál kezelést (ncurses könyvtárból visszatér).

Kivételek

<i>UninitializedConsoleException</i>	Ha a konzol még nem lett inicializálva.
--	---

4.4.2.4. **flush()**

```
void Console::flush ( ) [static]
```

Kiüríti a kimenetet, frissíti a terminált. A függvény frissíti a konzolt, ha a `tui` mód engedélyezve van.

Kivételek

<i>UninitializedConsoleException</i>	Ha a konzol nincs inicializálva
--	---------------------------------

4.4.2.5. **getInputOfLength()**

```
std::string Console::getInputOfLength (
    int l ) [static]
```

Bekér egy adott hosszúságú szöveget a felhasználótól. Ez a függvény legfeljebb `l` hosszúságú bemenetet olvas be a felhasználótól. TUI módban az `ncurses getch` függvényét használja, egyébként a szabványos bemenetről olvas `std::getline` segítségével.

Paraméterek

/	A beolvasandó karakterek maximális száma.
---	---

Visszatérési érték

A beolvasott szöveg.

Kivételek

<i>UninitializedConsoleException</i>	ha a konzol nincs inicializálva.
--	----------------------------------

Megjegyzés

Ha nem TUI módban EOF érkezik, a program kilép.

4.4.2.6. init()

```
void Console::init ( ) [static]
```

Inicializálja a konzolt. Ez a függvény előkészíti a konzolt a használatra. Ha a program interaktív módban fut, akkor meghívja az ncurses függvényeit.

Kivételek

<i>AlreadyInitializedConsoleException</i>	Ha a konzol már egyszer inicializálva lett.
---	---

4.4.2.7. moveCursor()

```
void Console::moveCursor (
    int x,
    int y ) [static]
```

A kurzor mozgatása a megadott (x, y) pozícióra. A függvény a kurzort áthelyezi a megadott koordinátákra, ha a konzol inicializálva van.

Megjegyzés

Csak TUI módban működik.

Paraméterek

x	Az oszlop pozíció (vízszintes koordináta).
y	A sor pozíció (függőleges koordináta).

Kivételek

<i>UninitializedConsoleException</i>	ha a konzol nincs inicializálva.
--	----------------------------------

4.4.2.8. Print() [1/2]

```
void Console::Print (
    int x,
    int y,
    const char * str ) [static]
```

Szöveg kiírása a megadott pozícióra. A megadott szöveget kiírja a (x, y) koordinátára TUI módban. Ha nem TUI módban van, akkor a szöveget a kimeneti adatfolyamba írja új sorral.

Paraméterek

<i>x</i>	Az oszlop pozíció (vízszintes koordináta).
<i>y</i>	A sor pozíció (függőleges koordináta).
<i>str</i>	A kiírandó szöveg.

Kivételek

<i>UninitializedConsoleException</i>	ha a konzol nincs inicializálva.
--	----------------------------------

4.4.2.9. Print() [2/2]

```
void Console::Print (
    int x,
    int y,
    const wchar_t * str ) [static]
```

Szöveg kiírása a megadott pozícióra. A megadott szöveget kiírja a (x, y) koordinátára TUI módban. Ha nem TUI módban van, akkor a szöveget a kimeneti adatfolyamba írja új sorral.

Paraméterek

<i>x</i>	Az oszlop pozíció (vízszintes koordináta).
<i>y</i>	A sor pozíció (függőleges koordináta).
<i>str</i>	A kiírandó szöveg.

Kivételek

<i>UninitializedConsoleException</i>	ha a konzol nincs inicializálva.
--	----------------------------------

4.4.2.10. RedirectOutput()

```
void Console::RedirectOutput (
    std::ostream & newOut ) [static]
```

Átirányítja a kimenetet egy új adatfolyamra. Ezzel a funkcióval beállítható egy új kimeneti adatfolyam. Az összes további kimenet az új adatfolyamba lesz irányítva.

Paraméterek

<i>newOut</i>	Az új adatfolyam, amelyre a kimenet irányítva lesz.
---------------	---

4.4.2.11. setColor()

```
void Console::setColor (
    CONSOLECOLOR color = WHITE ) [static]
```

Beállítja a színt a konzolon. A függvény beállítja a konzol színét a megadott `color` értékre. Ha a terminál nem támogatja a színeket, a függvény nem hajt végre semmilyen műveletet.

Paraméterek

<i>color</i>	A beállítandó szín, amelyet a <code>CONSOLECOLOR</code> típus reprezentál.
--------------	--

4.5. Expression osztályreferencia

Polinomiális kifejezést reprezentáló osztály. A kifejezés formája: $c[0] + c[1]*x + c[2]*x^2 + \dots$, ahol a `c[i]` értékeket a `coefficients` tömb tartalmazza.

```
#include <Expression.hpp>
```

Publikus tagfüggvények

- [Expression](#) (double *[coefficients](#), size_t [coefficientsLen](#))
Adott együtthatósorozattal létrehozott kifejezés.
- void [set](#) (double *[coefficients](#), size_t [coefficientsLen](#))
Új együtthatók beállítása.
- std::pair< double *, size_t > [getCoefficients](#) () const
Az együtthatók lekérdezése.
- double [valueAt](#) (double x) const
A kifejezés kiértékelése egy adott x értéknél.

Privát attribútumok

- `double * coefficients`
A polinom együtthatóinak tömbje.
- `size_t coefficientsLen`
Az együtthatók tömbjének hossza.

4.5.1. Részletes leírás

Polinomiális kifejezést reprezentáló osztály. A kifejezés formája: $c[0] + c[1]*x + c[2]*x^2 + \dots$, ahol a $c[i]$ értékeket a `coefficients` tömb tartalmazza.

4.5.2. Konstruktorkok és destruktorkok dokumentációja

4.5.2.1. Expression()

```
Expression::Expression (  
    double * coefficients,  
    size_t coefficientsLen ) [inline]
```

Adott együtthatósorozattal létrehozott kifejezés.

Paraméterek

<code>coefficients</code>	A polinom együtthatói.
<code>coefficientsLen</code>	Az együtthatók száma.

4.5.3. Tagfüggvények dokumentációja

4.5.3.1. getCoefficients()

```
std::pair<double*, size_t> Expression::getCoefficients ( ) const [inline]
```

Az együtthatók lekérdezése.

Visszatérési érték

A `coefficients` tömb pointere és hossza egy `std::pair`-ben.

4.5.3.2. set()

```
void Expression::set (
    double * coefficients,
    size_t coefficientsLen )
```

Új együtthatók beállítása.

Paraméterek

<i>coefficients</i>	A polinom új együtthatói.
<i>coefficientsLen</i>	Az új együtthatók száma.

Megjegyzés

Ez a függvény nem másolja le a tömböt, csak a pointert tárolja.

4.5.3.3. valueAt()

```
double Expression::valueAt (
    double x ) const
```

A kifejezés kiértékelése egy adott *x* értéknél.

Paraméterek

<i>x</i>	Az <i>x</i> változó értéke, ahol a kifejezést kiértékeljük.
----------	---

Visszatérési érték

A polinom értéke az adott *x*-nél.

4.6. Front osztályreferencia

A felhasználói felület és a kirajzolás vezérlője. Ez az osztály felelős a képernyő tartalmának megjelenítéséért, valamint a felhasználói bemenet bekéréséért és a képernyőváltásért.

```
#include <Front.hpp>
```

Publikus tagfüggvények

- void `Draw ()`
A képernyő kirajzolása.
- std::string `getInput ()`
Felhasználói szövegbevitel lekérése.

Privát attribútumok

- `Screen * currentDisplay`

A jelenleg megjelenített képernyő.

4.6.1. Részletes leírás

A felhasználói felület és a kirajzolás vezérlője. Ez az osztály felelős a képernyő tartalmának megjelenítéséért, valamint a felhasználói bemenet bekéréséért és a képernyőváltásért.

4.6.2. Tagfüggvények dokumentációja

4.6.2.1. Draw()

```
void Front::Draw ( )
```

A képernyő kirajzolása.

Megjegyzés

Amennyiben interaktív módban fut a program, keretet is rajzol.

4.6.2.2. getInput()

```
std::string Front::getInput ( )
```

Felhasználói szövegbevitel lekérése.

Visszatérési érték

A felhasználó által beírt szöveg.

4.7. InfoCommand osztályreferencia

Az információs képernyőt megjelenítő parancs. Ez a parancs felelős az [InfoScreen](#) aktiválásáért, amikor a felhasználó beírja az "info" parancsot vagy az "i" rövidítést.

```
#include <InfoCommand.hpp>
```

Publikus tagfüggvények

- void `exec` (const std::vector< std::string > ¶ms)

A parancs végrehajtása.

Privát attribútumok

- [Front & ui](#)

A felhasználói felület kezelője.

- [InfoScreen & screen](#)

Az információs képernyő, amit meg kell jeleníteni.

4.7.1. Részletes leírás

Az információs képernyőt megjelenítő parancs. Ez a parancs felelős az [InfoScreen](#) aktiválásáért, amikor a felhasználó beírja az "info" parancsot vagy az "i" rövidítést.

4.7.2. Tagfüggvények dokumentációja

4.7.2.1. `exec()`

```
void InfoCommand::exec (
    const std::vector< std::string > & params ) [virtual]
```

A parancs végrehajtása.

Paraméterek

<i>params</i>	A parancs paraméterei.
---------------	------------------------

Megjegyzés

Ez a függvény beállítja az InfoScreen-t az aktuális képernyőként.

Megvalósítja a következőket: [Command](#).

4.8. InfoScreen osztályreferencia

Információs képernyő osztály, amely az aktuális kifejezésről jelenít meg információkat. Az [InfoScreen](#) osztály célja, hogy szöveges formában megjelenítse az aktuálisan használt matematikai kifejezést, amelyet a felhasználó éppen rajzoltat.

```
#include <InfoScreen.hpp>
```

Publikus tagfüggvények

- void [Draw](#) ([Rect](#)< int > consoleArea) const

A képernyő tartalmának kirajzolása a konzolra.

Privát attribútumok

- [Expression](#) & [exp](#)

Az aktuálisan ábrázolt kifejezés.

4.8.1. Részletes leírás

Információs képernyő osztály, amely az aktuális kifejezésről jelenít meg információkat. Az [InfoScreen](#) osztály célja, hogy szöveges formában megjelenítse az aktuálisan használt matematikai kifejezést, amelyet a felhasználó éppen rajzoltat.

4.8.2. Tagfüggvények dokumentációja

4.8.2.1. Draw()

```
void InfoScreen::Draw (
    Rect< int > consoleArea ) const [virtual]
```

A képernyő tartalmának kirajzolása a konzolra.

Paraméterek

<code>consoleArea</code>	A konzolon megjelenítendő terület, ahova a képernyőt írjuk.
--------------------------	---

Megvalósítja a következőket: [Screen](#).

4.9. InvalidSizeConsoleException osztályreferencia

Akkor dobjuk, ha a program érvénytelen méretet szeretne valamelyik komponensnek beállítani.

```
#include <Console.hpp>
```

4.9.1. Részletes leírás

Akkor dobjuk, ha a program érvénytelen méretet szeretne valamelyik komponensnek beállítani.

4.10. InvalidStyleException osztályreferencia

Kivétel, ha érvénytelen stílusnevet ad meg a Ez a kivétel akkor kerül dobásra, ha a felhasználó által megadott stílusnév nem szerepel az elérhető stílusok között.

```
#include <StyleCommand.hpp>
```

4.10.1. Részletes leírás

Kivétel, ha érvénytelen stílusnevet ad meg a Ez a kivétel akkor kerül dobásra, ha a felhasználó által megadott stílusnév nem szerepel az elérhető stílusok között.

4.11. MoveCommand osztályreferencia

Parancs, amely elmozdítja a kirajzolt grafikon nézetét. A [MoveCommand](#) segítségével a felhasználó eltolhatja a kirajzolt területet a megadott x és y irányú értékek szerint.

```
#include <MoveCommand.hpp>
```

Publikus tagfüggvények

- void [exec](#) (const std::vector< std::string > ¶ms)
A parancs végrehajtása.

Privát attribútumok

- [PlotterScreen](#) & [plot](#)
A kirajzolandó képernyő, amelyen a mozgás történik.

4.11.1. Részletes leírás

Parancs, amely elmozdítja a kirajzolt grafikon nézetét. A [MoveCommand](#) segítségével a felhasználó eltolhatja a kirajzolt területet a megadott x és y irányú értékek szerint.

4.11.2. Tagfüggvények dokumentációja

4.11.2.1. exec()

```
void MoveCommand::exec (  
    const std::vector< std::string > & params ) [virtual]
```

A parancs végrehajtása.

Paraméterek

<i>params</i>	A parancs paraméterei: két szám, amelyek az x és y irányú eltolást adják meg.
---------------	---

Kivételek

<code>std::invalid_argument</code>	Ha a paraméterek nem konvertálhatók számokká.
------------------------------------	---

Megvalósítja a következőket: [Command](#).

4.12. OffsetCommand osztályreferencia

Parancs, amely eltolja a világterületet relatív mértékben. Az [OffsetCommand](#) segítségével a felhasználó relatív eltolást adhat meg a grafikon kirajzolási világterületére vonatkozóan.

```
#include <OffsetCommand.hpp>
```

Publikus tagfüggvények

- void [exec](#) (const std::vector< std::string > ¶ms)
A parancs végrehajtása.

Privát attribútumok

- [PlotterScreen](#) & [plot](#)
A kirajzolt nézet, amelyet eltolunk.

4.12.1. Részletes leírás

Parancs, amely eltolja a világterületet relatív mértékben. Az [OffsetCommand](#) segítségével a felhasználó relatív eltolást adhat meg a grafikon kirajzolási világterületére vonatkozóan.

4.12.2. Tagfüggvények dokumentációja

4.12.2.1. exec()

```
void OffsetCommand::exec (  
    const std::vector< std::string > & params ) [virtual]
```

A parancs végrehajtása.

Paraméterek

<i>params</i>	Két számot tartalmazó vektor: az x és y irányú relatív eltolás értékei.
---------------	---

Kivételek

<code>std::invalid_argument</code>	Ha a paraméterek nem konvertálhatók számokká.
------------------------------------	---

Megvalósítja a következőket: [Command](#).

4.13. OutCommand osztályreferencia

Parancs, amely a grafikon kimenetét fájlba menti. Lehetővé teszi, hogy a kirajzolt grafikon szöveges formában fájlba mentсүк.

```
#include <OutCommand.hpp>
```

Publikus tagfüggvények

- void [exec](#) (const std::vector< std::string > ¶ms)
A grafikon fájlba mentésének végrehajtása. A paraméterként megadott fájlba kerül a grafikon mentésre.

Privát attribútumok

- [PlotterScreen](#) & [plot](#)
A rajzoló képernyő, amelyet fájlba írunk ki.

4.13.1. Részletes leírás

Parancs, amely a grafikon kimenetét fájlba menti. Lehetővé teszi, hogy a kirajzolt grafikon szöveges formában fájlba mentсүк.

4.13.2. Tagfüggvények dokumentációja

4.13.2.1. exec()

```
void OutCommand::exec (  
    const std::vector< std::string > & params ) [virtual]
```

A grafikon fájlba mentésének végrehajtása. A paraméterként megadott fájlba kerül a grafikon mentésre.

Paraméterek

<i>params</i>	Egyetlen fájlnev stringet tartalmazó vektor.
---------------	--

Kivételek

UnableToOpenFileException	ha nem sikerül megnyitni a fájlt írásra.
---	--

Megvalósítja a következőket: [Command](#).

4.14. PlotCommand osztályreferencia

Parancs, amely beállítja a kirajzolandó polinomot és átvált a grafikon nézetre. A [PlotCommand](#) lehetővé teszi új polinom kifejezés beállítását a megadott együtthatók alapján, valamint átvált a grafikon megjelenítésére szolgáló képernyőre.

```
#include <PlotCommand.hpp>
```

Publikus tagfüggvények

- void [exec](#) (const std::vector< std::string > ¶ms)
A parancs végrehajtása, amely új polinomot állít be. Ha nem adunk meg paramétert, csak átvált a nézetre anélkül, hogy módosítaná a jelenlegi polinomot.

Privát attribútumok

- [Front](#) & [ui](#)
A kezelőfelület referenciája.
- [PlotterScreen](#) & [screen](#)
A megjelenítésre szolgáló képernyő, ahol a grafikon látható.
- [Expression](#) & [exp](#)
A polinom kifejezést reprezentáló objektum, amelyet frissítünk.

4.14.1. Részletes leírás

Parancs, amely beállítja a kirajzolandó polinomot és átvált a grafikon nézetre. A [PlotCommand](#) lehetővé teszi új polinom kifejezés beállítását a megadott együtthatók alapján, valamint átvált a grafikon megjelenítésére szolgáló képernyőre.

4.14.2. Tagfüggvények dokumentációja

4.14.2.1. exec()

```
void PlotCommand::exec (
    const std::vector< std::string > & params ) [virtual]
```

A parancs végrehajtása, amely új polinomot állít be. Ha nem adunk meg paramétert, csak átvált a nézetre anélkül, hogy módosítaná a jelenlegi polinomot.

Paraméterek

<i>params</i>	Együtthatók tömbje karakterláncként (string), amiket double-é konvertál.
---------------	--

Kivételek

<i>std::invalid_argument</i>	Ha valamelyik paraméter nem alakítható át double típusra.
------------------------------	---

Megvalósítja a következőket: [Command](#).

4.15. Plotter osztályreferencia

Absztrakt őszosztály grafikus kirajzolásokhoz. A [Plotter](#) osztály egy interfész, amely különféle kirajzolási stílusok implementálására szolgál. Minden származtatott osztálynak meg kell valósítania a Draw függvényt.

```
#include <Plotter.hpp>
```

Publikus tagfüggvények

- virtual void [Draw](#) ([Rect](#)< int > consoleArea, [Rect](#)< double > worldArea) const =0

Kirajolja a kifejezést a megadott konzol- és világkoordináták alapján. Ezt a függvényt a származtatott osztályok valósítják meg. A megadott területre (konzol és világ) rajzolják ki a kifejezést.

Védett attribútumok

- [Expression](#) & [exp](#)

Az ábrázolandó kifejezés.

- const char * [styleName](#)

A stílus neve, amely megkülönbözteti a megjelenítési módokat.

4.15.1. Részletes leírás

Absztrakt őszosztály grafikus kirajzolásokhoz. A [Plotter](#) osztály egy interfész, amely különféle kirajzolási stílusok implementálására szolgál. Minden származtatott osztálynak meg kell valósítania a Draw függvényt.

4.15.2. Tagfüggvények dokumentációja

4.15.2.1. Draw()

```
virtual void Plotter::Draw (
    Rect< int > consoleArea,
    Rect< double > worldArea ) const [pure virtual]
```

Kirajolja a kifejezést a megadott konzol- és világkoordináták alapján. Ezt a függvényt a származtatott osztályok valósítják meg. A megadott területre (konzol és világ) rajzolják ki a kifejezést.

Paraméterek

<code>consoleArea</code>	A konzol területe (karakterkoordinátákban).
<code>worldArea</code>	A világ területe (valós koordinátákban).

4.16. PlotterScreen osztályreferencia

A grafikon kirajzolásáért felelős képernyő. A [PlotterScreen](#) osztály kezeli a kirajzolt világ, a hozzá tartozó kifejezést, valamint a [Plotter](#) példányt, amely a tényleges kirajzolást végzi.

```
#include <PlotterScreen.hpp>
```

Publikus tagfüggvények

- void [Move](#) (double x, double y)
A világterület abszolút pozíciójának beállítása.
- void [Offset](#) (double dx, double dy)
A világterület relatív eltolása.
- void [Scale](#) (double w, double h)
A világterület méretének (léptékének) beállítása.
- void [Draw](#) ([Rect](#)< int > consoleArea) const
Kirajolás végrehajtása a konzol adott területére.

Privát attribútumok

- [Expression](#) & [exp](#)
A kirajzolandó kifejezés.
- [Plotter](#) * [plotter](#)
A kirajzolásért felelős objektum.
- [Rect](#)< double > [worldArea](#)
A világterület, amely meghatározza, mit látunk a grafikonból.

4.16.1. Részletes leírás

A grafikon kirajzolásáért felelős képernyő. A [PlotterScreen](#) osztály kezeli a kirajzolt világ, a hozzá tartozó kifejezést, valamint a [Plotter](#) példányt, amely a tényleges kirajzolást végzi.

4.16.2. Tagfüggvények dokumentációja

4.16.2.1. Draw()

```
void PlotterScreen::Draw (
    Rect< int > consoleArea ) const [virtual]
```

Kirajolás végrehajtása a konzol adott területére.

Paraméterek

<i>consoleArea</i>	A konzol terület, amelyre rajzolunk.
--------------------	--------------------------------------

Megvalósítja a következőket: [Screen](#).

4.16.2.2. Move()

```
void PlotterScreen::Move (
    double x,
    double y )
```

A világterület abszolút pozíciójának beállítása.

Paraméterek

<i>x</i>	Az új bal szélső x-koordináta.
<i>y</i>	Az új alsó y-koordináta.

4.16.2.3. Offset()

```
void PlotterScreen::Offset (
    double dx,
    double dy )
```

A világterület relatív eltolása.

Paraméterek

<i>dx</i>	Eltolás x irányban.
<i>dy</i>	Eltolás y irányban.

4.16.2.4. Scale()

```
void PlotterScreen::Scale (
    double w,
    double h )
```

A világterület méretének (léptékének) beállítása.

Paraméterek

<i>w</i>	Új szélesség.
<i>h</i>	Új magasság.

4.17. Rect< T > struktúrasablon-referencia

Általános téglalap-struktúra sablon.

```
#include <Rect.h>
```

Publikus attribútumok

- *T x*
A bal felső sarok X koordinátája.
- *T y*
A bal felső sarok Y koordinátája.
- *T w*
A téglalap szélessége.
- *T h*
A téglalap magassága.

4.17.1. Részletes leírás

```
template<typename T>
struct Rect< T >
```

Általános téglalap-struktúra sablon.

Sablon paraméterek

<i>T</i>	A koordináták és méretek típusa.
----------	----------------------------------

Megjegyzés

A téglalap bal-felső sarka az (x, y) koordináta, szélessége w, magassága h.

4.18. ScaleCommand osztályreferencia

Méretezési parancs a [PlotterScreen](#) számára. A parancs beállítja a világkoordináta-rendszer szélességét és magasságát.

```
#include <Scalecommand.hpp>
```

Publikus tagfüggvények

- void `exec` (const std::vector< std::string > ¶ms)
A parancs végrehajtása.

Privát attribútumok

- `PlotterScreen` & `plot`
A megjelenítendő grafikon képernyője, amelyen a változtatás történik.

4.18.1. Részletes leírás

Méretezési parancs a `PlotterScreen` számára. A parancs beállítja a világkoordináta-rendszer szélességét és magasságát.

Megjegyzés

Ez nem módosítja a konzol méretét.

Kivételek

<code>std::invalid_argument</code>	ha a paraméterek nem számokká konvertálhatók.
------------------------------------	---

4.18.2. Tagfüggvények dokumentációja

4.18.2.1. `exec()`

```
void ScaleCommand::exec (
    const std::vector< std::string > & params ) [virtual]
```

A parancs végrehajtása.

Paraméterek

<code>params</code>	A parancshoz tartozó paraméterek: szélesség, magasság.
---------------------	--

Kivételek

<code>std::invalid_argument</code>	Ha nem szám értékeket kap.
------------------------------------	----------------------------

Megvalósítja a következőket: `Command`.

4.19. Screen osztályreferencia

Absztrakt alaposztály a képernyőfelületekhez. Ez az osztály határozza meg az interfészt minden képernyő-típushoz, amely valamilyen módon megjelenítést végez a konzolon.

```
#include <Screen.h>
```

Publikus tagfüggvények

- virtual void Draw (Rect< int > consoleArea) const =0

A képernyő kirajzolása a megadott konzolterületre.

4.19.1. Részletes leírás

Absztrakt alaposztály a képernyőfelületekhez. Ez az osztály határozza meg az interfészt minden képernyő-típushoz, amely valamilyen módon megjelenítést végez a konzolon.

4.19.2. Tagfüggvények dokumentációja

4.19.2.1. Draw()

```
virtual void Screen::Draw (  
    Rect< int > consoleArea ) const [pure virtual]
```

A képernyő kirajzolása a megadott konzolterületre.

Paraméterek

<i>consoleArea</i>	A konzolon használt téglalap alakú terület (x, y, szélesség, magasság).
--------------------	---

Megvalósítják a következők: [PlotterScreen](#) és [InfoScreen](#).

4.20. StyleCommand osztályreferencia

A stílusváltást megvalósító parancsosztály.

```
#include <StyleCommand.hpp>
```

Publikus tagfüggvények

- void exec (const std::vector< std::string > ¶ms)

Végrehajtja a stílusváltást a megadott név alapján.

Privát attribútumok

- [PlotterScreen](#) & [plot](#)
A megjelenítendő grafikon képernyő.
- `std::vector< Plotter * > styles`
Az elérhető rajzolási stílusok.

4.20.1. Részletes leírás

A stílusváltást megvalósító parancsosztály.

4.20.2. Tagfüggvények dokumentációja

4.20.2.1. `exec()`

```
void StyleCommand::exec (
    const std::vector< std::string > & params ) [virtual]
```

Végrehajtja a stílusváltást a megadott név alapján.

Paraméterek

<i>params</i>	Egyetlen paraméterként a kiválasztott stílus neve.
---------------	--

Kivételek

InvalidStyleException	Ha a stílus nem található.
---------------------------------------	----------------------------

Megvalósítja a következőket: [Command](#).

4.21. TUICommand osztályreferencia

A parancs, amely ki- vagy bekapcsolja a szöveges felhasználói felületet (TUI). Ezzel a paranccsal a felhasználó be- vagy kikapcsolhatja a TUI megjelenítést. Ha a TUI engedélyezve van, akkor keretek és fejlécek jelennek meg; ha nem, akkor nyers konzolra történik a rajzolás.

```
#include <tuiCommand.hpp>
```

Publikus tagfüggvények

- void `exec` (const std::vector< std::string > ¶ms)
Végrehajtja a TUI mód be- vagy kikapcsolását. A [Console](#) osztály statikus módszerét használja az aktuális állapot lekérdezésére és annak megfordítására.

4.21.1. Részletes leírás

A parancs, amely ki- vagy bekapcsolja a szöveges felhasználói felületet (TUI). Ezzel a paranccsal a felhasználó be- vagy kikapcsolhatja a TUI megjelenítést. Ha a TUI engedélyezve van, akkor keretek és fejlécek jelennek meg; ha nem, akkor nyers konzolra történik a rajzolás.

4.21.2. Tagfüggvények dokumentációja

4.21.2.1. exec()

```
void TUICommand::exec (
    const std::vector< std::string > & params ) [virtual]
```

Végrehajtja a TUI mód be- vagy kikapcsolását. A [Console](#) osztály statikus metódusát használja az aktuális állapot lekérdezésére és annak megfordítására.

Paraméterek

<i>params</i>	Nincs paramétere.
---------------	-------------------

Megvalósítja a következőket: [Command](#).

4.22. UnableToOpenFileException osztályreferencia

Kivétel, ha nem sikerül megnyitni a fájl írásra.

```
#include <OutCommand.hpp>
```

4.22.1. Részletes leírás

Kivétel, ha nem sikerül megnyitni a fájl írásra.

4.23. UninitializedConsoleException osztályreferencia

Akkor dobjuk, ha a program hamarabb hívna [Console](#) függvényeket, minthogy azt inicializálná (init)

```
#include <Console.hpp>
```

4.23.1. Részletes leírás

Akkor dobjuk, ha a program hamarabb hívna [Console](#) függvényeket, minthogy azt inicializálná (init)

4.24. UnknownCommandException osztályreferencia

Akkor dobjuk, ha a `ConsoleManager`-ben nem lett regisztrálva a parancs.

```
#include <CommandManager.hpp>
```

4.24.1. Részletes leírás

Akkor dobjuk, ha a `ConsoleManager`-ben nem lett regisztrálva a parancs.

5. fejezet

Jog

A programot Ferencz Péter írta (kivétel: gtest lite, szabadon felhasználható; memtrace.h / memtrace.cpp, Peregi Tamas 2011, Szeberenyi Imre 2013).

A projekt a *GNU GENERAL PUBLIC LICENSE* alatt fut. Minden jog fenntartva.