



EÖTVÖS LORÁND TUDOMÁNYEGYETEM

INFORMATIKAI KAR

NUMERIKUS ANALÍZIS TANSZÉK

Deriválási szabályok alkalmazása és szemléltetése

Témavezető:

Lócsi Levente
adjunktus

Készítette:

Czvalinga Bálint
programtervező informatikus BSc

Budapest, 2025

Tartalomjegyzék

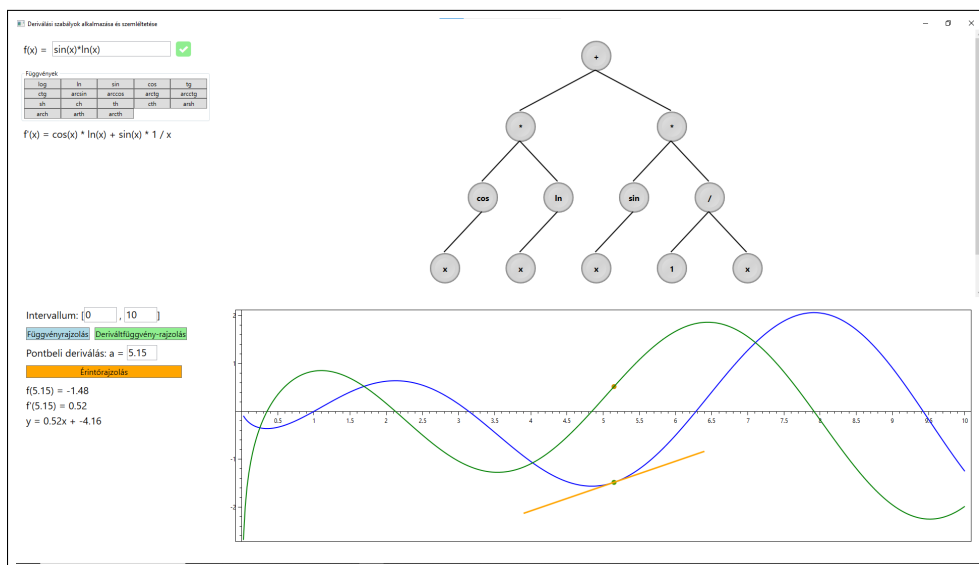
1. Bevezetés	2
2. Felhasználói dokumentáció	3
2.1. Rendszerkövetelmények	3
2.2. Telepítés	3
2.3. Felületi leírás, használat	4
2.3.1. Függvény megadása	4
2.3.2. Deriválás	6
2.3.3. Függvényrajzolás	7
3. Fejlesztői dokumentáció	9
3.1. Matematikai háttér	9
3.2. Tervezés	10
3.2.1. Megvalósítandó feladat	10
3.2.2. Használati esetek	11
3.2.3. Felhasználói történetek	11
3.3. Megvalósítás	17
3.3.1. Modell	17
3.3.2. FunctionEvaluator	24
3.3.3. Model	24
3.4. Nézetmodell	24
3.5. Nézet	24
3.6. Tesztelés	34
3.6.1. Fehér-doboz tesztelés	34
3.6.2. Fekete-doboz tesztelés	34

1. fejezet

Bevezetés

A mesterséges intelligencia világában a matematikai háttértudás elengedhetetlen az informatika mélyebb megértéséhez. Az olyan műveletek, mint a deriválás, nem csupán iskolai tananyagként jelennek meg, hanem a gyakorlati alkalmazások – speciálisan az optimalizálási feladatok – alapkövei is. A gépi tanulásban gyakran használt módszerek, például a gradiens alapú algoritmusok, a deriváltak segítségével tanulnak és hoznak döntéseket.

Szakedolgozatomban egy oktatási célú alkalmazás fejlesztésére vállalkoztam, amely képes elemi függvények deriváltját kiszámítani, valamint ábrázolni a számítás lépéseit és az eredmény grafikonját. Az informatika szemüvegén keresztül, szintaxisfákkal illusztrálni a deriválás menetét nemcsak nagyon érdekes folyamat, hanem lehetőség is arra, hogy a matematikai műveleteket strukturált, szemléletes formában mutassam be a felhasználóknak.



2. fejezet

Felhasználói dokumentáció

2.1. Rendszerkövetelmények

Az alkalmazás futtatásához Windows operációs rendszer szükséges, legalább Windows 10 vagy újabb verzió. A program a .NET 8 keretrendszert igényli, amely biztosítja a WPF technológia működéséhez szükséges környezetet.

Szoftverkövetelmények:

- **Operációs rendszer:** Windows 10/11 (x64)
- **Keretrendszer:** .NET 8

Hardverkövetelmények:

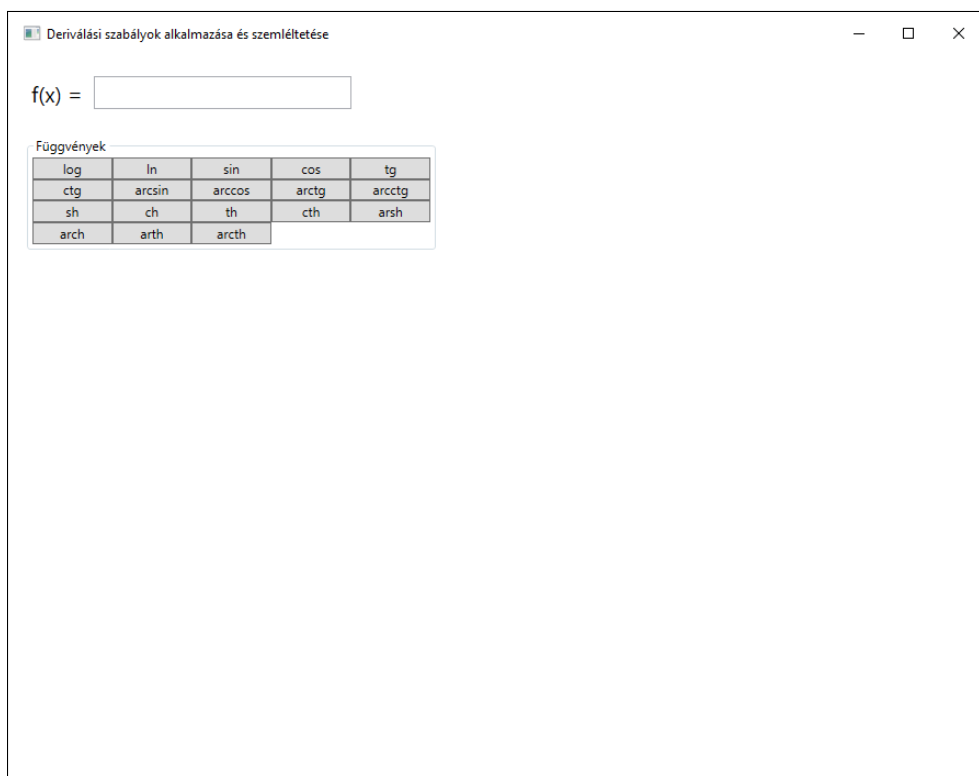
- **Processzor:** Minimum kétmagos, 1.8 GHz-es, ajánlott négymagos, legalább 2.4 GHz-es.
- **RAM:** Legalább 2 GB, ajánlott 4 GB.
- **Tárhely:** 200 MB
- **Grafikus kártya:** 512MB – DirectX 9 vagy újabb, hardveres gyorsítás támogatása ajánlott

2.2. Telepítés

A program használata nem igényel telepítést. A futtatható állomány munkára fogásához szükség van a .NET 8.0 keretrendszerre, amelyet a felhasználó letölthet a **Microsoft** oldaláról.

2.3. Felületi leírás, használat

A program indításakor megjelenik a főablak. A legfelső mezőbe lehet megadni egy függvény hozzárendelési szabályát, a mező alatt pedig a program által felismert függvények listája található. A szintaxisfa megjelenítő és a függvényrajzoló komponens ekkor még nem látható.



2.1. ábra. Az üres főablak

2.3.1. Függvény megadása

Függvény megadásához bele kell kattintani a fenti szövegdobozba, és egy elemi függvény hozzárendelési szabályát kell megadni. A szövegdoboz alatt láthatóak az ismert alapfüggvények, melyeket a felhasználó szabadon komponálhat egymással, valamint számokkal és operátorokkal. A szövegdoboz

- a következő operátorokat ismeri: +, -, *, /, ^.
- a hatványozás operátorát egyaránt érti klasszikus hatványozásra és exponenciális függvény megadására.
- olyan számokat fogad el, amelyek legfeljebb két tizedesjegyet tartalmaznak és a (-100, 100) intervallumból származnak.

- az e szimbólumot is elfogadja mint szám.
- tizedesvesszőt nem fogad el, tizedespontot kell használni a törtrész beírásához.
- az implicit szorzást nem fogadja el, így például $5x$ helyett $5*x$ -et kell megadni.
- negatív előjelet csak szám előtt fogad el, így pl. $-\sin(x)$ helyett $-1*\sin(x)$ -et kell megadni.
- negatív előjelet mindig a számhoz kapcsolja, így pl. a -1^x zárójelezése $(-1)^x$ lesz, nem pedig $-(1^x)$.
- a függvény hatványozásának egyszerűsített írásmódját nem ismeri, így pl. a $\sin^2(x)$ helyett $\sin(x)^2$ -t kell megadni.
- számít a felhasználó együttműködésére, így pl. a $2+3$ bemenetet nem 5-ként fogja tárolni, nem képes kifejezések összevonására.
- maximum 20 karaktert tartalmazhat.

$f(x) =$

$\sin(x^2+2*x)$

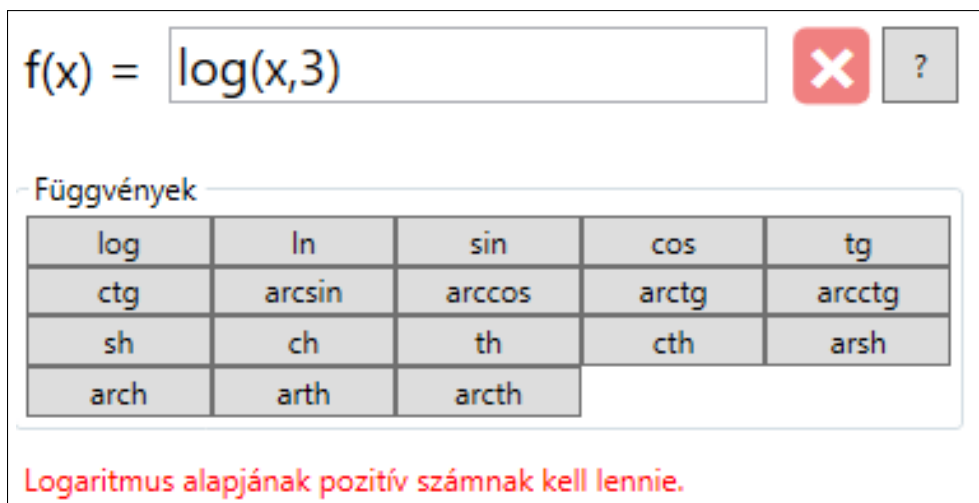
✓

Függvények

log	ln	sin	cos	tg
ctg	arcsin	arccos	arctg	arcctg
sh	ch	th	cth	arsh
arch	arth	arcth		

2.2. ábra. Példa elfogadott függvényre

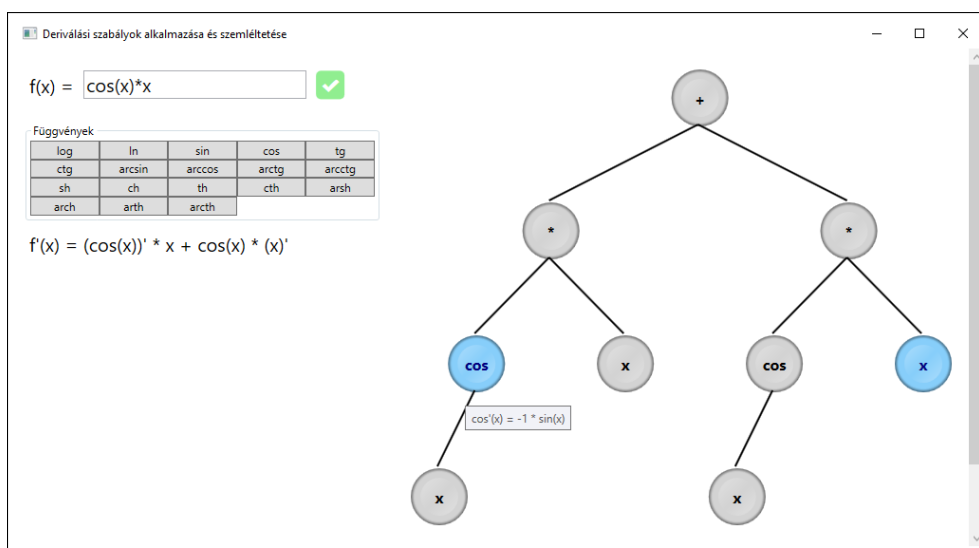
A felismert függvények listája gombokat tartalmaz, ha a felhasználó rákattint az egyikre és még befér a 20 karakterbe, akkor az adott függvény bekerül a szövegdobozba egy zárójelpárral, illetve tetszőleges alapú logaritmus esetén egy vessző is kerül a zárójelek közé. A vessző elé egy pozitív, 1-gyel nem egyenlő számot kell írni, ez lesz az alap, a vessző után pedig bármilyen kifejezés lehet, ez az argumentum. A program valós időben visszajelzést ad, hogy elfogadja-e a beírt karakterláncot. Ha nem, akkor megjelenik egy sűrű, amire rákattintva piros szöveg jelzi, hogy mi a probléma a megadott függvénnyel. A hibák teljes listáját a 3.11 táblázat tartalmazza. Ha a karakterlánc megfelelő, megjelenik a kifejezés szintaxisfája és kezdődhet a deriválás, illetve a függvény grafikonjának ábrázolására megjelenik "Függvényrajzolás" felirattal egy kék gomb.



2.3. ábra. Példa elutasított függvényre

2.3.2. Deriválás

Sikeres függvénymegadást követően megjelenik a hozzárendelési szabály szintaxisfája a képernyő jobb oldalán. Ha ezután még változtatunk a hozzárendelési szabályon, és az helytelen, akkor a fa nem változik meg, az utolsó helyes kifejezést fogja reprezentálni. A deriválható csúcsok kék színűek, kurzor érintésre megjelenik a csúcshoz tartozó deriválási szabály és kattintásra történik meg a deriválás. Az alap deriválási szabályokat a (hivatkozás) tartalmazza, emellett az alkalmazás felismeri a függvénykompozíciókat, és ez esetben az összetett függvény deriválásának (tétel hivatkozás) megfelelő szabályt adja. A deriválás folyamatát matematikai jelölésekkel is nyomon lehet követni a bal oldalon. A deriválás befejeztével megjelenik a bal oldalon egy "Egyszerűsítés" feliratú gomb, amire ha rákattintunk, a (hivatkozás)-nak megfelelő egyszerűsítéseket elvégzi a program. A gomb megnyomása után lehetőség van a deriváltfüggvény grafikonjának megjelenítésére.



2.4. ábra. A deriválás menete

2.3.3. Függvényrajzolás

Függvény és deriváltfüggvény

Helyes hozzárendelési szabály megadása után megjelenik egy állítható intervallum, illetve a függvény rajzolására szolgáló kék gomb. Ha a deriválás befejeződött és megtörtént az egyszerűsítés, akkor a deriváltfüggvény rajzolására megjelenik egy zöld gomb. Az intervallum megadásának a szabályai:

- Az intervallum kezdete legyen kisebb, mint a vége
- Az intervallum legyen részintervalluma a $[-50, 50]$ intervallumnak
- Mindkét számnak legfeljebb két tizedesjegye lehet
- Tizedesvesszőt nem fogad el, tizedespontot kell használni a törtrész beírásához

A két gomb bármelyikére kattintva az alkalmazás hibát jelez, ha az intervallum nem megfelelő, egyéb esetben megjeleníti a függvényt vagy deriváltfüggvényt. Bármely függvény grafikonjának egy pontjára kattintva megjelenik egy ablak, amiben a függvény hozzárendelési szabálya és az adott pont koordinátái találhatók. Ha a felhasználó kattintás közben mozgatja az egerét, az ablak is frissül a koordinátáknak megfelelően. Mindkét függvény ugyanabba a koordináta-rendszerbe kerül. A koordináta-rendszer megtartja az állapotát addig, amíg nem változik a függvény vagy nem változik az intervallum. Ha az előbb említettek közül valamelyik megváltozik, akkor bármely rajzoló gombra kattintás esetén egy új koordináta-rendszer jön létre, a korábban ábrázolt függvények pedig eltűnnek.

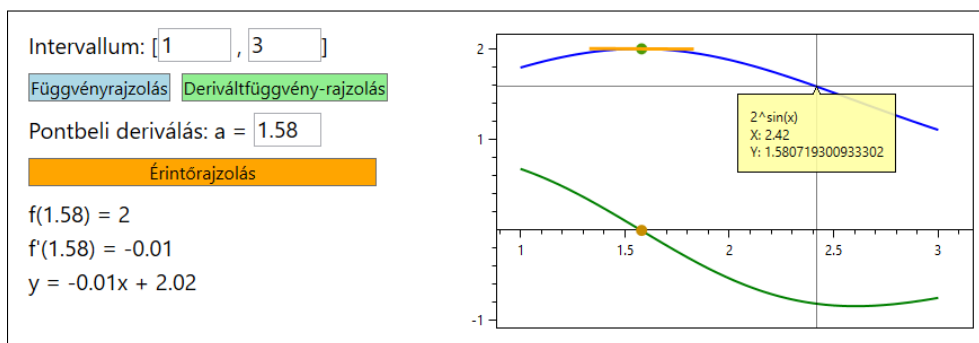
Érintő

A deriválás befejezése és a függvény ábrázolása után megjelenik egy újabb bemeneti mező, illetve egy érintő rajzolására szolgáló gomb. A mezőbe egy számot vár a program, amely az x tengely egy pontja, és amelynek a következő feltételeknek kell eleget tennie:

- Legfeljebb 2 tizedesjegyet tartalmaz
- Tizedespontot használ az egészrész és a törtrész elkülönítésére
- A megadott intervallumon belül van
- A pont benne van a függvény értelmezési tartományában
- A függvény deriválható a pontban

Ha ezen feltételek valamelyike nem teljesül, akkor az érintő rajzolásakor a program egy felugró ablakban hibát jelez. Egyéb esetben megjelenik a függvény adott pontbeli érintője, illetve a függvény grafikonján egy zöld, a deriváltfüggvény grafikonján pedig egy sárga

pont. Emellett a bal oldalon látható a függvény és a deriváltfüggvény értéke a megadott pontban, valamint az érintő egyenlete. A függvény grafikonján lévő zöld pontra kattintva, majd az egeret mozgatva a pont követi a kurzort az x tengely mentén, és a bal oldali értékek, valamint az érintő is dinamikusan frissül. Emellett új pont megadásával az érintő a megadott pontba ugrik. Ha a felhasználó változtat a függvény hozzárendelési szabályán vagy az intervallumon, akkor a pontbeli deriválás panele eltűnik.



2.5. ábra. Függvényábrázolás

Megjegyzések:

- Mozgatás közben előfordulhat, hogy olyan helyre húzza a felhasználó a pontot az egerrel, ahol a függvény vagy a deriváltfüggvény nincs értelmezve, ez esetben nem ad hibát a program, de a bal oldali értékek NaN-ok lesznek. Ezesetben a mozgatható pont, valamint az érintő is eltűnik, de új pont megadásával és az érintő rajzolásával visszahozhatók.
- Ismert hiba, hogy ha egy függvényt az értelmezési tartományánál bővebb intervallumon ábrázolunk, akkor a koordináta-rendszer az értelmezési tartományhoz igazodik, viszont a pont húzásának határa a megadott intervallum lesz, tehát a pontot lényegében lehúzhatjuk a koordináta-rendszerrel. Ekkor is eltűnik a pont és az érintő, és az előzőhöz hasonlóan szerezhetők vissza.
- A függvény szakadási helyeinek felismerése "favágó módszerrel" (hivatkozás a fejlesztői dokumentációra) történik, így az olyan függvények mint a tangens és a kotangens hibásan jelennek meg.

3. fejezet

Fejlesztői dokumentáció

3.1. Matematikai háttér

Definíció.

Legyen $\emptyset \neq A \subset \mathbb{R}$. Az $a \in A$ pont az A halmaz **belső pontja**, ha

$$\exists r > 0, \quad \text{hogy} \quad K_r(a) := (a - r, a + r) \subset A.$$

Jelölés: $\boxed{\text{int } A} := \{a \in A \mid a \text{ belső pontja } A\text{-nak}\}.$

Definíció.

Az $f \in \mathbb{R} \rightarrow \mathbb{R}$ függvény az $a \in \text{int } D_f$ pontban **deriválható**, ha

$$\exists \text{ és véges a } \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} \quad \text{határérték.}$$

Ezt $f'(a)$ -val jelöljük, és az f függvény **a pontbeli deriváltjának** nevezzük, azaz

$$f'(a) := \lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} \in \mathbb{R}.$$

Definíció.

Ha $f \in \mathbb{R} \rightarrow \mathbb{R}$ és $B := \{x \in \text{int } D_f \mid f \in D\{x\}\} \neq \emptyset$, akkor az

$$B \ni x \mapsto f'(x)$$

függvényt az f **deriváltfüggvényének** nevezzük, és az f' szimbólummal jelöljük.

Definíció.

Az $f \in \mathbb{R} \rightarrow \mathbb{R}$ függvény grafikonjának az $(a, f(a))$ pontban **van érintője**, ha $f \in D\{a\}$.

Az f függvény grafikonjának $(a, f(a))$ pontbeli **érintőjén** az

$$y = f'(a) \cdot (x - a) + f(a)$$

egyenletű egyenest értjük.

Tétel: Az algebrai műveletek és a derivált kapcsolata.

Tegyük fel, hogy $f, g \in D\{a\}$ valamilyen $a \in \text{int}(D_f \cap D_g)$ pontban. Ekkor

- 1° $c \cdot f \in D\{a\}$ és $(c \cdot f)'(a) = c \cdot f'(a)$ ($c \in \mathbb{R}$),
- 2° $f + g \in D\{a\}$ és $(f + g)'(a) = f'(a) + g'(a)$,
- 3° $f \cdot g \in D\{a\}$ és $(f \cdot g)'(a) = f'(a) \cdot g(a) + f(a) \cdot g'(a)$,
- 4° ha még a $g(a) \neq 0$ feltétel is teljesül, akkor
$$\frac{f}{g} \in D\{a\} \quad \text{és} \quad \left(\frac{f}{g}\right)'(a) = \frac{f'(a) \cdot g(a) - f(a) \cdot g'(a)}{g^2(a)}.$$

Tétel: Az összetett függvény deriválása.

Tegyük fel, hogy $f, g \in \mathbb{R} \rightarrow \mathbb{R}$, $\mathcal{R}_g \subset D_f$ és egy $a \in \text{int } D_g$ pontban $g \in D\{a\}$, továbbá $f \in D\{g(a)\}$. Ekkor $f \circ g \in D\{a\}$, és

$$(f \circ g)'(a) = f'(g(a)) \cdot g'(a).$$

Az alapfüggvények szimbolikus deriválási szabályai a 3.1. táblázatban olvashatók.

3.2. Tervezés

3.2.1. Megvalósítandó feladat

A szakdolgozatomban célja egy olyan alkalmazás fejlesztése, ami a felhasználó által megadott elemi függvények deriváltjának kiszámítását teszi lehetővé, a számítás részletes menetének és a függvények grafikonjának szemléltetésével. Szándékom az is, hogy a program az Analízis II. tárgyhoz oktatási segédletként szolgáljon.

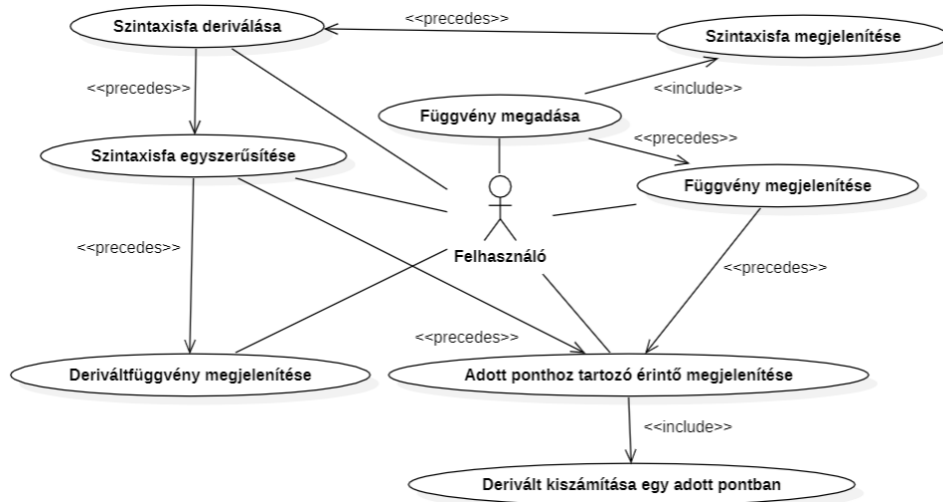
Az alkalmazás a következő szempontoknak kell megfeleljen:

- A felhasználó által megadott hozzárendelési szabályon végezzen szintaktikai elemzést és adjon visszajelzést annak eredményéről.
- A függvénynek jelenjen meg a szintaxisfája, valamint a szintaxisfán keresztül, lépésenként lehessen eljutni a deriváltfüggvényhez úgy, hogy az adott lépésben használt deriválási szabály is látható legyen.
- A deriválás lépései szövegesen is jelenjenek meg.
- Lehessen ábrázolni a függvényt, a deriváltfüggvényt és egy adott pontbeli érintőt koordináta-rendszerben.

Lépésenkénti deriválás alatt azt értem, hogy az alkalmazás ne rögtön a deriváltfüggvényt adja vissza, hanem a matematikai bevezetőben tárgyalt tételek és szabályokból egyszerre csak egyet-egyet alkalmazzon. Például az $f(x) = 2 + \sin(x^2)$ függvény deriválása a következő folyamatként legyen megfigyelhető: $(2 + \sin(x^2))' = 2' + \sin'(x^2) = 0 + \sin'(x^2) = 0 + \cos(x^2) \cdot (x^2)' = 0 + \cos(x^2) \cdot 2x^1 = \cos(x^2) \cdot 2x$

3.2.2. Használati esetek

A következő diagram tartalmazza a felhasználó által elindítható, és az azok következményeként bekövetkező főbb folyamatokat.



3.1. ábra. Használati eset diagram

3.2.3. Felhasználói történetek

A következő táblázatokban a megvalósítandó program funkcióinak részletes bemutatása szerepel felhasználói szemszögből.

Alkalmazás indítása

3.2. táblázat. Alkalmazás indításának története

GIVEN	Az alkalmazás Windows 10 vagy újabb operációs rendszeren megtalálható
-------	---

WHEN	A felhasználó elindítja az alkalmazást
THEN	Megjelenik az üres főablak

Kilépés

3.3. táblázat. Kilépés története

GIVEN	Az alkalmazás fut
WHEN	A felhasználó a bezáró ikonra kattint
THEN	Az alkalmazás leáll

Függvénymegadás

3.4. táblázat. Függvénymegadás története

GIVEN	A felhasználó belekattint a függvénymegadás szövegdobozába
WHEN	A felhasználó helytelenül adja meg egy függvény hozzárendelési szabályát (hivatkozás)
THEN	Megjelenik a súgó
GIVEN	A felhasználó belekattint a függvénymegadás szövegdobozába

WHEN	A felhasználó helyesen megadja egy függvény hozzárendelési szabályát
THEN	Megjelenik a hozzárendelési szabály szintaxisfája, az intervallum megadó panel és a „Függvényrajzolás” feliratú gomb

Súgó használata

3.5. táblázat. Súgó használatának története

GIVEN	A súgó látható a függvénymegadás szövegdozoza mellett
WHEN	A felhasználó rákattint a súgó gombra
THEN	Megjelenik piros színnel egy szöveg, ami jelzi, hogy mi a probléma a hozzárendelési szabállyal

Deriválás

3.6. táblázat. Deriválás használatának története

GIVEN	Megjelent a hozzárendelési szabály szintaxisfája
WHEN	A felhasználó rákattint az egyik deriválandó (kék) csúcsra
THEN	A csúcshoz tartozó részfa kicserélődik a derivált részfára
GIVEN	A felhasználó elkezdte a lépésenkénti deriválást

WHEN	Nincs több deriválandó csúcs
THEN	Megjelenik az „Egyszerűsítés” feliratú gomb

Egyszerűsítés

3.7. táblázat. Egyszerűsítés használatának története

GIVEN	Megjelent az „Egyszerűsítés” feliratú gomb
WHEN	A felhasználó rákattint az „Egyszerűsítés” feliratú gombra
THEN	A szintaxisfa a (hivatkozás). táblázatban leírt szabályoknak megfelelően egyszerűsödik és megjelenik a „Deriváltfüggvény-rajzolás” feliratú gomb

Függvényrajzolás

3.8. táblázat. Függvényrajzolás használatának története

GIVEN	Megjelent a függvényrajzolás intervalluma és a „Függvényrajzolás” feliratú gomb
WHEN	A felhasználó beállít egy rossz intervallumot (hivatkozás hogy mi a megfelelő intervallum) és rákattint a „Függvényrajzolás” feliratú gombra
THEN	Az alkalmazás felugró ablakban jelzi a problémát

GIVEN	Megjelent a függvényrajzolás intervalluma és a „ <i>Függvényrajzolás</i> ” feliratú gomb
WHEN	A felhasználó beállít egy megfelelő intervallumot (hivatkozás hogy mi a megfelelő intervallum) és rákattint a „ <i>Függvényrajzolás</i> ” feliratú gombra
THEN	A függvény grafikonja megjelenik a koordináta-rendszerben

Deriváltfüggvény-rajzolás

3.9. táblázat. Deriváltfüggvény-rajzolás használatának története

GIVEN	Megjelent a „ <i>Deriváltfüggvény-rajzolás</i> ” feliratú gomb
WHEN	A felhasználó beállít egy rossz intervallumot (hivatkozás hogy mi a megfelelő intervallum) és rákattint a „ <i>Deriváltfüggvény-rajzolás</i> ” feliratú gombra
THEN	Az alkalmazás felugró ablakban jelzi a problémát
GIVEN	Megjelent a „ <i>Deriváltfüggvény-rajzolás</i> ” feliratú gomb
WHEN	A felhasználó beállít egy megfelelő intervallumot (hivatkozás hogy mi a megfelelő intervallum) és rákattint a „ <i>Deriváltfüggvény-rajzolás</i> ” feliratú gombra
THEN	A deriváltfüggvény grafikonja megjelenik a koordináta-rendszerben

Érintőrajzolás

3.10. táblázat. Érintőrajzolás használatának története

GIVEN	A derivált egyszerűsítése megtörtént
WHEN	A felhasználó rákattint a „ <i>Függvényrajzolás</i> ” feliratú gombra
THEN	Megjelenik a pontbeli deriválás szövegdozoza és az „ <i>Érintőrajzolás</i> ” feliratú gomb
GIVEN	A felhasználó lerajzolta a függvény grafikonját
WHEN	A felhasználó leegyszerűsíti a derivált szintaxisfáját
THEN	Megjelenik a pontbeli deriválás szövegdozoza és az „ <i>Érintőrajzolás</i> ” feliratú gomb
GIVEN	A felhasználó megadott egy megfelelő (hivatkozás) abszcisszát
WHEN	A felhasználó rákattint az „ <i>Érintőrajzolás</i> ” feliratú gombra
THEN	A függvény grafikonján (és a deriváltfüggvényén, ha már szerepel a koordináta-rendszerben) megjelenik a megadott abszcisszához tartozó pont és a hozzátartozó érintő, valamint megjelenik a függvény adott pontbeli értéke, a deriváltfüggvény pontbeli értéke és az érintő egyenlete
GIVEN	Megjelent az érintő a függvényen
WHEN	A felhasználó rákattint az érintőre, lenyomva tartja a bal klikket és elkezdi mozgatni a kurzorát

THEN	A függvényen (és a deriváltfüggvényen, ha már felrajzolta a felhasználó) a pont és az érintő követi a kurzort az x-tengely mentén, valamint a megadott pont értéke, a függvény pontbeli értéke, a deriváltfüggvény pontbeli értéke és az érintő egyenlete is valós időben változnak
------	---

3.3. Megvalósítás

Az alkalmazást eredetileg MVVM architektúrában terveztem elkészíteni, azonban az irányelvek jelentősen megnehezítették a megvalósítást, pontosabban a bináris fa megjelenítést és a függvényrajzolást. Ezért az alkalmazás az MVVM architektúrának megfelelő komponensekkel rendelkezik, de a nézetmodell közvetlenül a programkódból rajzolja a függvényeket, valamint a bináris fák megjelenítéséért felelő `BinaryTreeView` osztály is kódban hozza létre a fát.

3.3.1. Modell

TokenType

A *TokenType* felsoroló típus definiálja a lehetséges token típusokat.

Token

A *Token* osztály egy *string* értékkel és egy *TokenType* típusú besorolással reprezentál egy lexikai egységet a matematikai kifejezésből. A következő táblázat a matematikai bevezetőnek megfelelően összefoglalja a lehetséges lexikai egységeket.

Token típusa	Lexikai egységek
Szám	Legfeljebb két tizedesjeggyel rendelkező, (-100, 100) intervallumbeli szám vagy az e szimbólum
Változó	x
Operátor	$+$, $-$, $*$, $/$, $^$
Függvény	\sin , \cos , tg , ctg , arcsin , arccos , arctg , arcctg , sh , ch , th , cth , arsh , arch , arth , arcth , \log , \ln
Nyitó zárójel	(
Berekesztő zárójel)
Vessző	,

ASTNode

Az `ASTNode` osztály egy matematikai kifejezést reprezentáló absztrakt szintaxisfa (AST) egy csúcsát modellezi, amely lehet szám, változó, operátor vagy függvény. Minden csúcs rendelkezik egy értékkel (`Value`), valamint opcionálisan bal (`Left`) és jobb (`Right`) rész-fával és egy `Locator` azonosítóval. A `ToBeDifferentiated` tulajdonság jelzi, hogy az adott csúcs deriválásra vár-e, a `DiffRule` pedig a csúcshoz tartozó, a matematikai bevezetőnek megfelelő deriválási szabály szöveges alakját tartalmazza. A `DeepCopy()` metódus egy csúcs mély másolására szolgál, mely során a másolt csúcs megőröklí az eredeti `ToBeDifferentiated` és `Locator` értékét. A `HasDifferentiationNode(ASTNode)` statikus függvény a paraméterben kapott szintaxisfáról visszaadja, hogy van-e benne olyan csúcs, melynek `ToBeDifferentiated` értéke igaz. A `ToString()`, `SubtreeToString()`, `AddParenthesesIfNeeded(ASTNode, string)` és `NeedsParentheses(ASTNode, string)` metódusok a szintaxisfát ember által olvasható, a matematikai jelöléseknek megfelelő szöveggé alakítják. A `Simplify()` metódus egyszerűsíti az AST-t ismert algebrai azonosságok mentén, melyeket a (hivatkozás) táblázat tartalmaz. Az `IsOperator()` és `IsFunction()` me-

tóduosok segítségével megállapítható, hogy a csúcs egy operátort vagy ismert függvényt tartalmaz-e.

Tokenizer

A *Tokenizer* osztály célja egy matematikai kifejezést tartalmazó szöveg feldarabolása tokenekre. A `Tokenize()` függvény a konstruktorban megadott `string`-et karakterenként dolgozza fel:

- A `whitespace` karaktereket figyelmen kívül hagyja.
- Ha talál egy szám karaktert, meghívja a `TokenizeNumber()` függvényt, ami egy szám token létrehozásáért felel.
- Ha talál egy betű karaktert, meghívja a `TokenizeFunctionOrVariableOrE()` függvényt, ami egy függvény, az `x` változó és az `e` szám létrehozásáért felel.
- Ha talál egy operátort, vesszőt, nyitó vagy berekesztő zárójelet, akkor létrehozza az annak megfelelő token.

Parser

A *Parser* osztály célja, hogy a *Tokenizer* által előállított tokenekből *AST*-t építsen fel, amely matematikai kifejezések struktúráját tükrözi. A `ParseExpression()` metódus elindítja az elemzési folyamatot, amely rekurzív módon dolgozza fel az összeadás, kivonás, szorzás, osztás, hatványozás, zárójelezés, változók, számok és függvények struktúráját.

Differentiator

A *Differentiator* osztály célja egy matematikai kifejezést reprezentáló *AST* lépésenkénti deriválása. Az osztály a konstruktorban megkap egy `ASTNode` fát, melynek a gyökerét megjelöli deriválásra és beteszi a `differentiationSteps` listába, amely a deriválás lépéseinek szintaxisfáit tárolja. A `Differentiate(int)` metódus egy `Locator` értéket kap, és a következő lépéseket hajtja végre:

1. A `differentiationSteps` listából lekérdezi az utolsó deriválási lépésben megkapott fát, amit a `CurrentTree` számított mező tartalmaz (nevezzük ezt a fát eredeti fának).
2. Az eredeti fát és a `Locator` értéket továbbadja a `FindDifferentiationNode(ASTNode, int)` függvénynek, ami megkeresi és visszaadja a deriválni kívánt részfat (deriválandó részfa).
3. A deriválandó részfat továbbadja a `DifferentiateOnce(ASTNode)` metódusnak, ami elvégzi a szimbolikus deriválás egy lépését a (hivatkozás) szabályoknak megfelelően, majd kijelöli a következő lépéseket a `FlagNode(ASTNode)` függvény segítségével és visszaadja a kapott fát (derivált részfa).

4. Az eredeti fában a deriválandó részfat kicseréli a derivált részfára a `ReplaceNode(ASTNode, ASTNode, ASTNode)` módszerrel.
5. Hozzáadja a `differentiationSteps` listához az így kapott új fat.

Az `UnflagNode(ASTNode)` segédfüggvényt a `DifferentiateOnce(ASTNode)` függvény használja azokon a helyeken, ahol a deriválandó szintaxisfa megjelenik a derivált szintaxisfában is, például $(e^x)' = e^x$.

Példák

A következőkben bemutatom a `Differentiator` osztály működését két példán keresztül. Ha egy a csúcs `ToBeDifferentiated` értéke igaz, azt így jelölöm: $(a)'$

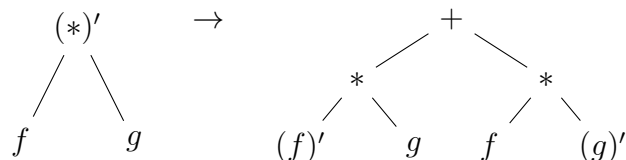
Az első példa legyen a szorzás általános esete, tegyük fel, hogy az $f * g$ kifejezés szintaxisfáját megadjuk a `Differentiator` osztály konstruktorának, ahol f és g két tetszőleges részfa. Ekkor a fa gyökerét $(*)$ a konstruktor megjelöli a `FlagNode(ASTNode)` függvénnyel, hogy az a deriválás következő (első) lépése, és a `differentiationSteps` első eleme a következő fa lesz:



Ezután tegyük fel, hogy a $(*)'$ csúcs `Locator` értékével meghívjuk a `Differentiate(int)` függvényt, és koncentráljunk arra, hogy a `DifferentiateOnce(ASTNode)` függvényben mi történik. Mivel a deriválandó szintaxisfa gyökere a szorzás operátor, ezért a következő kódrészlet fog lefutni:

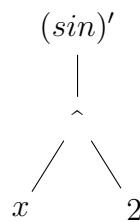
```
case "*": // (f*g)' = f'*g + f*g'
    return new ASTNode("+",
        new ASTNode("*", FlagNode(left.DeepCopy(), right.
            DeepCopy())) ,
        new ASTNode("*", left.DeepCopy(), FlagNode(right.
            DeepCopy())));
```

A kódban található `left` az analógiánkban az f , a `right` pedig a g . A `DifferentiateOnce` függvény lényegében a következő transzformációt végzi el:



A deriválás folyamata úgy folytatódna, hogy vagy az $(f)'$, vagy a $(g)'$ csúcs `Locator` értékét megadjuk a `Differentiator(int)` függvénynek, ám mivel most nem tudjuk, hogy mi pontosan f és g , így be kell érünk ennyivel. Azt kaptuk tehát, hogy $(fg)' = f'g + fg'$, ami megfelel a matematikai bevezetőben láttotaknak.

A második példában tegyük fel, hogy a $\sin(x^2)$ függvény szintaxisfáját megadjuk a `Differentiator` osztály konstruktorának. Ekkor a fa gyökerét (\sin) a konstruktor megjelöli a `FlagNode(ASTNode)` függvénnyel, hogy az a deriválás következő (első) lépése, és a `differentiationSteps` első eleme a következő fa lesz:



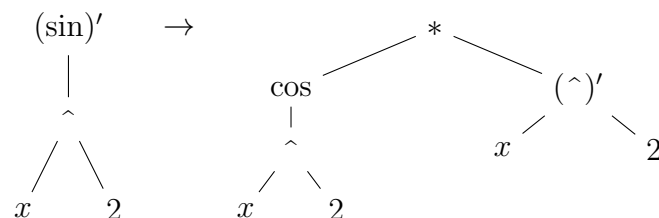
Ezután tegyük fel, hogy a $(\sin)'$ csúcs `Locator` értékével meghívjuk a `Differentiate(int)` függvényt, és koncentráljunk arra, hogy a `DifferentiateOnce(ASTNode)` függvényben mi történik. Mivel a deriválandó szintaxisfa gyökere a szinuszfüggvény, valamint az argumentuma nem x , ezért a következő kódrészlet fog lefutni:

```

case "sin": // sin'(f) = cos(f)*f'
    ...
    return new ASTNode("*",
        new ASTNode("cos", left.DeepCopy()),
        FlagNode(left.DeepCopy()));

```

A kódban található `left` az analógiánkban az x^2 . A `DifferentiateOnce` függvény lényegében a következő transzformációt végzi el:



Ezután tegyük fel, hogy a $(^)'$ csúcs `Locator` értékével meghívjuk a `Differentiate(int)` függvényt, és nézzük meg részletesen, hogy mi történik.

```

public ASTNode Differentiate(int locator)
{

```

```

ASTNode lastTree = CurrentTree.DeepCopy();

ASTNode nodeToDifferentiate = FindDifferentiationNode(lastTree,
    locator);

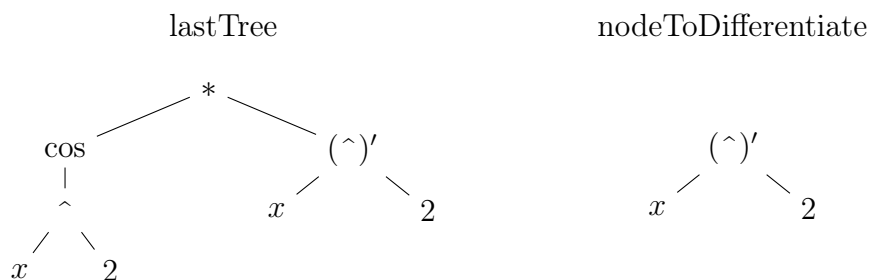
ASTNode differentiatedNode = DifferentiateOnce(
    nodeToDifferentiate);

ASTNode newTree = ReplaceNode(lastTree, nodeToDifferentiate,
    differentiatedNode);

differentiationSteps.Add(newTree);
return newTree;
}

```

Itt a `lastTree` az előző lépésben kapott fa, amiben a `FindDifferentiationNode(lastTree, locator)` hívással a program visszaadja azt a részfát, ami deriválandó, és a `Locator` értéke megegyezik a paraméterként kapott `locator` értékkel. Tehát a `lastTree` és a `nodeToDifferentiate` a következőképpen néznek ki:



A `differentiatedNode` fát úgy kapjuk, hogy a `nodeToDifferentiate` részfának folytatjuk a deriválását. Mivel ennek a részfának a gyökere a hatványozás operátor, a bal gyermeke az x változó, a jobb gyermeke pedig a 2 szám, ezért a `DifferentiateOnce` függvényben a következő kódrészlet fog lefutni:

```

case "^(n)":
    ...
    else if ((double.TryParse(right.Value, NumberStyles.Any,
        CultureInfo.InvariantCulture, out double num1) || right.
        Value == "e") && left.Value == "x") // (x^n)' = n*x^(n-1)
    {
        ...
        return new ASTNode("*(n)",
            right.DeepCopy(),

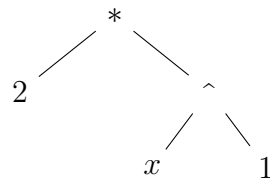
```

```

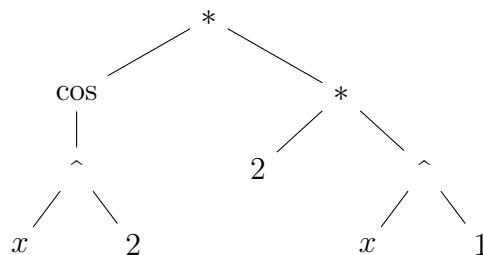
new ASTNode("^",
    left.DeepCopy(),
    new ASTNode((num1 - 1).ToString(CultureInfo.
        InvariantCulture))));
}
...

```

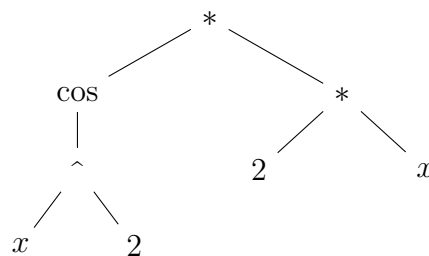
A példában `num1` értéke 2, `right` az x csúcs, `left` pedig a 2 csúcs. Jól látható, hogy a kódrészletben nincs `FlagNode` hívás, így ennek a részfának a deriválása véget ért. A `differentiatedNode` így néz ki:



Az utolsó lépés, hogy az eredeti `lastTree` fában a `nodeToDifferentiate` részfat kicseréljük a `differentiatedNode` fára. Ezt a műveletet végzi a `ReplaceNode(lastTree, nodeToDifferentiate, differentiatedNode)` hívás, ami visszaadja az új, `newTree` fat, ami így néz ki:



Mivel nincs több deriválandó csúcs, így a deriválás folyamata véget ért. Azonban jól látható, hogy a fat lehetne egyszerűsíteni, hiszen $x^1 = x$. Tegyük fel, hogy meghívjuk erre a fára az `ASTNode` osztály `Simplify` metódusát, ami az ismert egyszerűsítéseket (hivatkozás) elvégezve a következő fával fog visszatérni:



A példa végére azt kaptuk, hogy $(\sin(x^2))' = \cos(x^2) \cdot (x^2)' = \cos(x^2) \cdot 2x^1 = \cos(x^2) \cdot 2x$, ami megfelel a matematikai bevezetőben látottaknak.

3.3.2. FunctionEvaluator

3.3.3. Model

3.4. Nézetmodell

3.5. Nézet

3.11. táblázat. Függvénymegadás hibái

Kiváltó osztály	Hibaüzenet	Példa bemenet
Tokenizer	Több tizedespont nem lehet egy számban.	2.3.4
Tokenizer	A tizedespont után csak 2 szám engedélyezett.	2.354
Tokenizer	A szám nem végződhet tizedesponttal.	2.
Tokenizer	A szám abszolút értékének maximuma 99.99 lehet.	100
Tokenizer	Nem várt karakter: <karakter>	, ", @, !
Tokenizer	Ismeretlen token: <token>. Csak az 'x' változó és az ismert függvények engedélyezettek.	sec(x), cos(y)
Parser	A(z) <operátor> bináris operátornak hiányzik a jobb oldali operandusa.	x *

Folytatás a következő oldalon

Kiváltó osztály	Hibaüzenet	Példa bemenet
Parser	Nullával való osztás nem engedélyezett.	$2 / (x - x)$
Parser	A '^' hatványozás operátornak hiányzik a kitevője.	$2 ^$
Parser	A 0^0 nem értelmezett.	$(x-x)^0$
Parser	A hatványozás alapja nem lehet negatív szám vagy 0.	$(-1)^x$
Parser	A negatív előjelet ('-') számnak kell követnie.	$-x$
Parser	Hiányzó operátor a <negatív szám> után.	$-1x$
Parser	Hiányzó operátor a(z) <szám> után.	$5x$
Parser	A <függvéynév> függvény után nyitó zárójelnek kell következnie: '('.	$\sin x$
Parser	A(z) <függvéynév> függvénynek hiányzik az argumentuma.	$\sin()$
Parser	A logaritmus függvényből hiányzik az alap.	$\log(,x)$

Folytatás a következő oldalon

Kiváltó osztály	Hibaüzenet	Példa bemenet
Parser	A logaritmus alapjának pozitív számnak kell lennie.	$\log(x, 2)$
Parser	A logaritmus alapja nem lehet 1.	$\log(1, x)$
Parser	A logaritmus alapja nem lehet 0.	$\log(0, 2)$
Parser	A logaritmus alapja után vesszőnek kell következnie: ','.	$\log(2 \ x)$
Parser	A logaritmus függvényből hiányzik az argumentum a vessző után.	$\log(2,)$
Parser	A <függvéynév> függvény argumentuma után berekesztő zárójelnek kell következnie: ')'	$\sin(x$
Parser	A <függvéynév> függvény berekesztő zárójele után operátornak kell következnie.	$\sin(x)2$
Parser	A kifejezésből hiányzik egy berekesztő zárójel: ')'	$(x+2$
Parser	A berekesztő zárójel után operátornak kell következnie.	$(x+2)2$
Parser	Nem várt token a kifejezésben.	,

3.13. táblázat. Felhasználói történetek

#	Felhasználói eset		Leírás
1.	Alkalmazás indítása	GIVEN	Az alkalmazás Windows 10 vagy újabb operációs rendszeren megtalálható
		WHEN	A felhasználó elindítja az alkalmazást
		THEN	Megjelenik az üres főablak
2.	Kilépés	GIVEN	Az alkalmazás fut
		WHEN	A felhasználó a bezáró ikonra kattint
		THEN	Az alkalmazás leáll
3.	Függvénymegadás	GIVEN	A felhasználó belekattint a függvénymegadás szövegdobozába
		WHEN	A felhasználó helytelenül adja meg egy függvény hozzárendelési szabályát (hivatkozás)
		THEN	Megjelenik a súgó
		GIVEN	A felhasználó belekattint a függvénymegadás szövegdobozába
		WHEN	A felhasználó helyesen megadja egy függvény hozzárendelési szabályát

#	Felhasználói eset		Leírás
		THEN	Megjelenik a hozzárendelési szabály szintaxisfája
		GIVEN	A felhasználó belekattint a függvénymegadás szövegdobozába
		WHEN	A felhasználó helyesen megadja egy függvény hozzárendelési szabályát
		THEN	Megjelenik a függvényrajzoláshoz szükséges intervallum és a függvényrajzolás gomb
4.	Súgó használata	GIVEN	A súgó látható a függvénymegadás szövegdozoza mellett
		WHEN	A felhasználó rákattint a súgó gombra
		THEN	Megjelenik piros színnel egy szöveg, ami jelzi, hogy mi a probléma a hozzárendelési szabállyal
		GIVEN	Megjelent a hozzárendelési szabály szintaxisfája
		WHEN	A felhasználó ráviszi a kurzorát az egyik deriválandó (kék) csúcsra
		THEN	Megjelenik a csúcshoz tartozó deriválási szabály

5. Deriválás

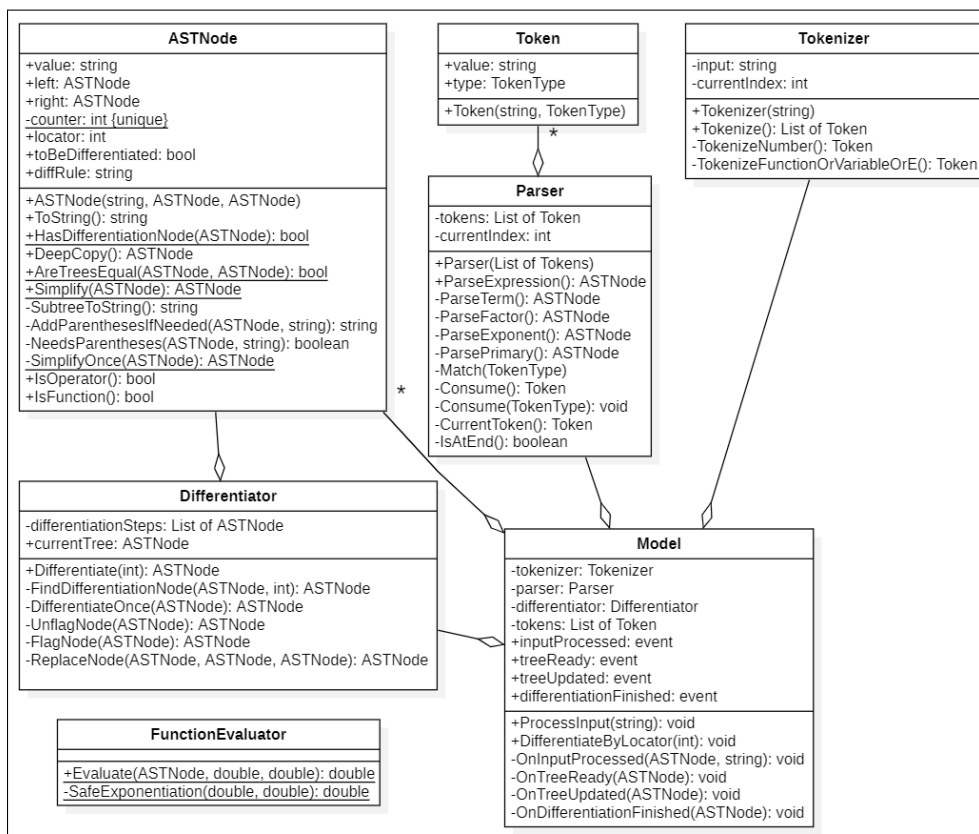
#	Felhasználói eset		Leírás
		GIVEN	Megjelent a hozzárendelési szabály szintaxisfája
		WHEN	A felhasználó rákattint az egyik deriválandó (kék) csúcsra
		THEN	A csúcshoz tartozó részfa kicserélődik a derivált részfára
		GIVEN	A felhasználó elkezdte a lépésenkénti deriválást
		WHEN	Nincs több deriválandó csúcs
		THEN	Megjelenik az Egyszerűsítés feliratú gomb
6.	Egyszerűsítés	GIVEN	Megjelent az "Egyszerűsítés" feliratú gomb
		WHEN	A felhasználó rákattint az "Egyszerűsítés" feliratú gombra
		THEN	A szintaxisfa a (hivatkozás). táblázatban leírt szabályoknak megfelelően egyszerűsödik
		GIVEN	Megjelent az "Egyszerűsítés" feliratú gomb
		WHEN	A felhasználó rákattint az "Egyszerűsítés" feliratú gombra

#	Felhasználói eset		Leírás
		THEN	Megjelenik a "Deriváltfüggvény-rajzolás" feliratú gomb
7.	Függvényrajzolás	GIVEN	Megjelent a függvényrajzolás intervalluma és a függvényrajzolás gomb
		WHEN	A felhasználó beállít egy megfelelő intervallumot (hivatkozás hogy mi a megfelelő intervallum)
		THEN	A függvény grafikonja megjelenik a koordináta-rendszerben
		GIVEN	Megjelent a függvényrajzolás intervalluma és a függvényrajzolás gomb
		WHEN	A felhasználó beállít egy rossz intervallumot (hivatkozás hogy mi a megfelelő intervallum) és rákattint a függvényrajzolás gombra
		THEN	Az alkalmazás felugró ablakban jelzi a problémát
8.	Deriváltfüggvény-rajzolás	GIVEN	Megjelent a "Deriváltfüggvény-rajzolás" feliratú gomb
		WHEN	A felhasználó beállít egy megfelelő intervallumot (hivatkozás hogy mi a megfelelő intervallum)

#	Felhasználói eset		Leírás
		THEN	A deriváltfüggvény grafikonja megjelenik a koordináta-rendszerben
		GIVEN	Megjelent a "Deriváltfüggvény-rajzolás" feliratú gomb
		WHEN	A felhasználó beállít egy rossz intervallumot (hivatkozás hogy mi a megfelelő intervallum) és rákattint a "Deriváltfüggvény-rajzolás" feliratú gombra
		THEN	Az alkalmazás felugró ablakban jelzi a problémát
9.	Érintő rajzolás	GIVEN	A derivált egyszerűsítése megtörtént
		WHEN	A felhasználó rákattint a „ <i>Függvényrajzolás</i> ” feliratú gombra
		THEN	Megjelenik a pontbeli deriválás szövegdoboza és az "Érintő rajzolás" gomb
		GIVEN	A felhasználó lerajzolta a függvény grafikonját
		WHEN	A felhasználó leegyszerűsíti a derivált szintaxisfáját
		THEN	Megjelenik a pontbeli deriválás szövegdoboza és az "Érintő rajzolás" gomb

#	Felhasználói eset		Leírás
		GIVEN	A felhasználó megadott egy megfelelő (hivatkozás) abszcisszát
		WHEN	A felhasználó rákattint az "Érintő rajzolás" gombra
		THEN	A függvény grafikonján megjelenik a megadott abszcisszához tartozó pont és a hozzátartozó érintő, valamint megjelenik a függvény adott pontbeli értéke, a deriváltfüggvény pontbeli értéke és az érintő egyenlete.
		GIVEN	A felhasználó megadott egy megfelelő (hivatkozás) abszcisszát és már a deriváltfüggvény is szerepel a koordináta-rendszerben
		WHEN	A felhasználó rákattint az "Érintő rajzolás" gombra
		THEN	A függvény grafikonján megjelenik a megadott abszcisszához tartozó pont és a hozzátartozó érintő, és a deriváltfüggvényen is megjelenik a megadott abszcisszához tartozó pont, valamint megjelenik a függvény adott pontbeli értéke, a deriváltfüggvény pontbeli értéke és az érintő egyenlete.
		GIVEN	Megjelent az érintő a függvényen

#	Felhasználói eset		Leírás
		WHEN	A felhasználó rákattint az érintőre, lenyomva tartja a bal klikket és elkezd mozgatni a kurzorát
		THEN	A függvényen (és a deriváltfüggvényen, ha már felrajzolta a felhasználó) a pont és az érintő követi a kurzort az x-tengely mentén, valamint a megadott pont értéke, a függvény pontbeli értéke, a deriváltfüggvény pontbeli értéke és az érintő egyenlete is dinamikusan változnak.



3.2. ábra. Modell osztálydiagramja

3.6. Tesztelés

3.6.1. Fehér-doboz tesztelés

3.6.2. Fekete-doboz tesztelés

3.1. táblázat. Szimbolikus deriválási szabályok

$f(x)$	$f'(x)$	$f(x)$	$f'(x)$
c ($c \in \mathbb{R}$)	0	$\operatorname{tg}(x)$	$\frac{1}{\cos^2(x)}$
x^n ($n \in \mathbb{Z}$)	nx^{n-1}	$\operatorname{arctg}(x)$	$\frac{1}{1+x^2}$
$\frac{1}{x}$	$-\frac{1}{x^2}$	$\operatorname{ctg}(x)$	$-\frac{1}{\sin^2(x)}$
x^α ($\alpha \in \mathbb{R}$)	$\alpha x^{\alpha-1}$	$\operatorname{arcctg}(x)$	$-\frac{1}{1+x^2}$
e^x	e^x	$\operatorname{sh}(x)$	$\operatorname{ch}(x)$
$\ln(x)$	$\frac{1}{x}$	$\operatorname{arsh}(x)$	$\frac{1}{\sqrt{x^2+1}}$
a^x ($a \in (0, +\infty)$)	$a^x \ln(a)$	$\operatorname{ch}(x)$	$\operatorname{sh}(x)$
$\log_a(x)$ ($a \in \mathbb{R}^+ \setminus \{1\}$)	$\frac{1}{x \ln(a)}$	$\operatorname{arch}(x)$	$\frac{1}{\sqrt{x^2-1}}$
$\sin(x)$	$\cos(x)$	$\operatorname{th}(x)$	$\frac{1}{\operatorname{ch}^2(x)}$
$\arcsin(x)$	$\frac{1}{\sqrt{1-x^2}}$	$\operatorname{arth}(x)$	$\frac{1}{1-x^2}$
$\cos(x)$	$-\sin(x)$	$\operatorname{cth}(x)$	$-\frac{1}{\operatorname{sh}^2(x)}$
$\arccos(x)$	$-\frac{1}{\sqrt{1-x^2}}$	$\operatorname{arcth}(x)$	$\frac{1}{1-x^2}$

3.12. táblázat. Egyszerűsítési szabályok

Eredeti kifejezés	Egyszerűsített kifejezés
$0 + 0$	0
$f + 0 \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	f
$0 + f \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	f
$f - 0 \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	f
$f - f \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	0
$0 \cdot f \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	0
$f \cdot 0 \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	0
$1 \cdot f \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	f
$f \cdot 1 \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	f
$0/f \quad (f \in \mathbb{R} \rightarrow \mathbb{R}, f \neq 0)$	0
$f/1 \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	f
$f/f \quad (f \in \mathbb{R} \rightarrow \mathbb{R}, f \neq 0)$	1
$1^f \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	1
$f^1 \quad (f \in \mathbb{R} \rightarrow \mathbb{R})$	f
$f^0 \quad (f \in \mathbb{R} \rightarrow \mathbb{R}, f \neq 0)$	1
$0^f \quad (f \in \mathbb{R} \rightarrow \mathbb{R}, f \neq 0)$	0

Ábrák jegyzéke

2.1. Az üres főablak	4
2.2. Példa elfogadott függvényre	5
2.3. Példa elutasított függvényre	6
2.4. A deriválás menete	6
2.5. Függvényábrázolás	8
3.1. Használati eset diagram	11
3.2. Modell osztálydiagramja	33

Táblázatok jegyzéke

3.2. Alkalmazás indításának története	11
3.3. Kilépés története	12
3.4. Függvénymegadás története	12
3.5. Súgó használatának története	13
3.6. Deriválás használatának története	13
3.7. Egyszerűsítés használatának története	14
3.8. Függvényrajzolás használatának története	14
3.9. Deriváltfüggvény-rajzolás használatának története	15
3.10. Érintőrajzolás használatának története	16
3.11. Függvénymegadás hibái	24
3.13. Felhasználói történetek	27
3.1. Szimbolikus deriválási szabályok	35
3.12. Egyszerűsítési szabályok	36