Зміст

[Список ілюстрацій 4](#_Toc517169787)

[Постановка задачі 4](#_Toc517169794)

[Рис. 1 «Постановка задачі» 5](#_Toc517169795)

[Структура програми 5](#_Toc517169796)

[Рис. 2 «Структура програми» 6](#_Toc517169797)

[Опис методів та класів 6](#_Toc517169798)

[Інструкція для користувача 7](#_Toc517169799)

[Рис. 3 «Робота програми» 7](#_Toc517169800)

[Рис. 3.1 «Робота програми» 8](#_Toc517169801)

[Рис. 3.2 «Робота програми» 8](#_Toc517169802)

[Рис. 3.3 «Робота програми» 9](#_Toc517169803)

[Постановка задачі 9](#_Toc517169804)

[Структура програми 10](#_Toc517169805)

[Рис. 1 «Структура програми» 10](#_Toc517169806)

[Рис. 1.2 «Структура програми» 10](#_Toc517169807)

[Опис методів та класів 10](#_Toc517169808)

[Інструкція для користувача 13](#_Toc517169809)

[Рис. 3 «Робота програми» 13](#_Toc517169811)

[Рис. 3.1 «Робота програми» 13](#_Toc517169813)

[Рис. 3.2 «Робота програми» 14](#_Toc517169814)

[Рис. 3.3 «Робота програми» 14](#_Toc517169815)

[Висновок 1 15](#_Toc517169816)

[Висновок 3 15](#_Toc517169817)

# Список ілюстрацій

## Рис. 1 «Постановка задачі»……………………………………………………………………………………….……3

## Рис. 2 «Структура програми» …………………………………………………………………………………………4

## Рис. 3 «Робота програми» ……………………………………………………………………………………………..5

## Рис. 3.1 «Робота програми» …………………………………………………………………………………………..6

## Рис. 3.2 «Робота програми»……………………………………………………………………………………………6

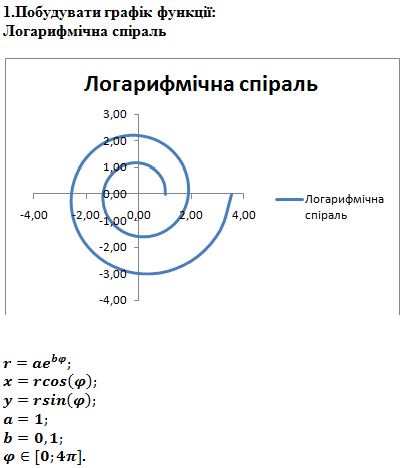
## Рис. 3.3 «Робота програми»……………………………………………………………………………………………7

# Постановка задачі

Написати програму яка будує графік заданої функції.

Обов'язкові функції:

1. Можливість задавати початкові дані, діапазон та крок.
2. Збереження графіку в файл.



## Рис. 1 «Постановка задачі»

# Структура програми

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

## Рис. 2 «Структура програми»

# Опис методів та класів

У програмі наявний тільки один клас My\_Function.

У цьому класі наявні такі методи:

* public static void main()

Метод, що запускає програму, а також:

* присвоює почтакові значення змінних, створює дві нових змінних для збереження максимальних значень х та у;
* створює набір данних XYSeries, де ми будемо зберігати наші х та у;
* цикл, що заповнює XYSeries й рахує максимальне значення х та у для того, щоб потім намалювати осі координат;
* створює «колекцію» для цих серії;
* створює сам графік на основі нашої «коллекції» з XYSeries;
* створює наше вікно програми за допомогою JFrame;
* створює дві панельки, одну для зображення графіка, другу – для кнопок;
* створює кнопки й спеціальну панель для графіку, що буде відображати наш графік у нашому вікні;
* додає actionListener до кнопок «зберегти» та «ввести»;
* додає все це на наше вікно, створює для нього певний дизайн.
* private static JFreeChart repaint(double step, double a, double b, double phiF, double phiT,)

Метод, що оновлює дані, на основі яких будується графік й перемальовує його, заново створюючи XYSeries, цикл, колекцію, сам графік, який і повертає в кінці.

* private static getVars()

Метод, що оновлює значення змінних після того як користувач змінює їх під час роботи програми за допомогою JSpinner

# 

# Інструкція для користувача

Як тільки Ви запустили програму, перед вами відкриється діалогове вікно, у якому буде намальований графік з даним за замовченням.

( параметри – step = 0.001, a = 1, b = 0.1, phiF = 0, phiT = 4 )Изображение выглядит как снимок экрана

Описание создано с высокой степенью достоверности

## Рис. 3 «Робота програми»

Якщо ви захочете змінити параметри для графіку, вам треба буде ввести інші параметри у потрібні поля і натиснути «Enter» після цього.

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

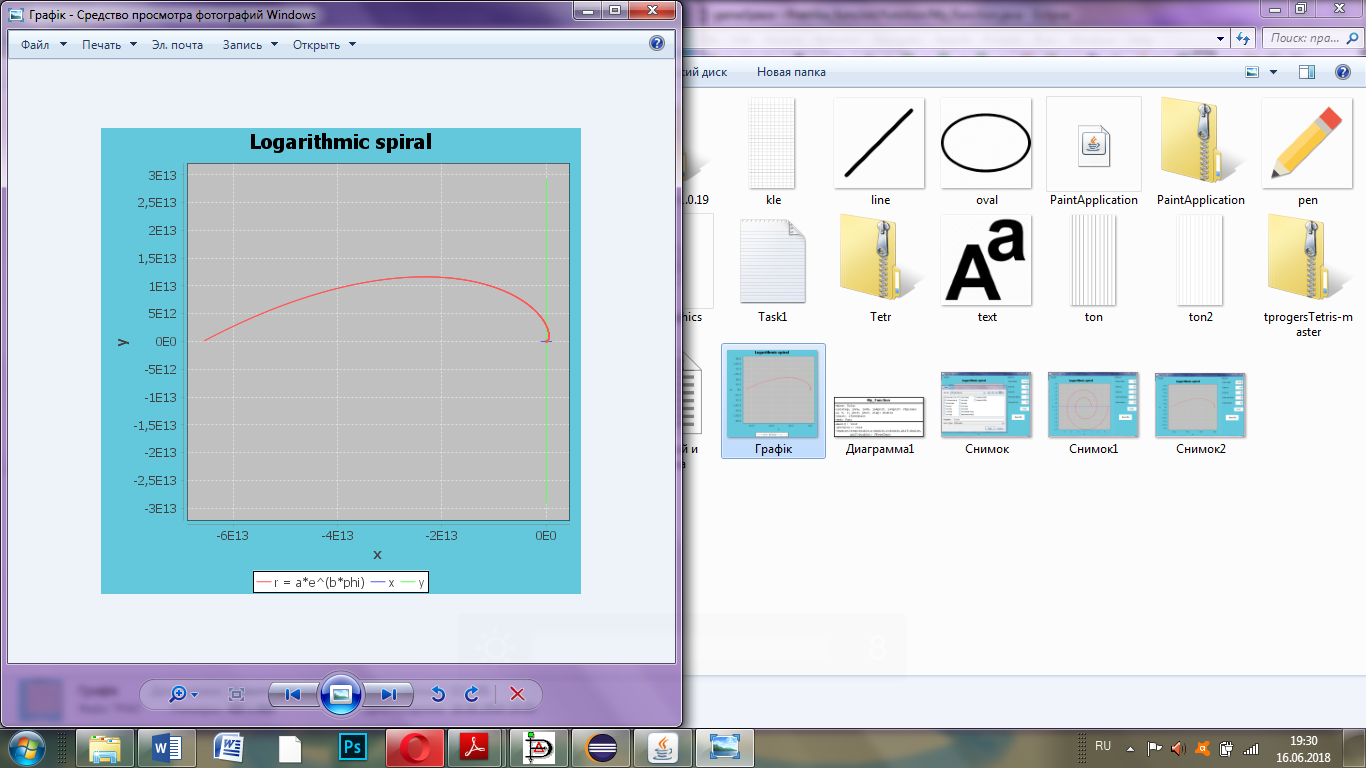
## Рис. 3.1 «Робота програми»

Якщо ви захочете зберегти графік як зображення, потрібно просто натиснути «Save» і після цього обрати папку, у яку ви хочете зберігти файл. Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

## Рис. 3.2 «Робота програми»

Потім ваше зображення можно буде знайти там де ви його зберігли.



## Рис. 3.3 «Робота програми»

# Постановка задачі

Створити гру «Тетріс». Написати для цього функціональну програму, написати графічну програму. Тобто має бути реалізований GUI (графічний інтерфейс користувача, дизайн). Ми розділили написання цієї програми з Ніколаюк Дарією таким чином: я писала клас Board та більшу частину класу Tetris, який ми писали разом.

# Структура програмиИзображение выглядит как снимок экрана, текст Описание создано с очень высокой степенью достоверности

## Рис. 1 «Структура програми»

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

## Рис. 1.2 «Структура програми»

# Опис методів та класів

У програмі наявні три класи: Tetris( головний клас, звідки ініціалізацується вікно й запускається програма), Board( клас для самої гри, «дошка», на якій розміщуються фігурки, що падатимуть під час гри), Piece( клас де реалізовані методи для фігурок) та enum TetroShapes( тут розташована інформація про колір й вигляд та координати самих фігурок).

У класі Tetris наявні такі методи:

* public static void main()

Метод, що запускає програму, створюючи об’єкт цього класу, тим самим створюючи новий JFrame.

* Конструктор класу Tetris

Цей конструктор:

* інціалізує верхню стрічку для балів та надписів GameOver/Pause;
* інціалізує панельку для виводу підказок щодо користування грою;
* додає картинку на цю панельку;
* запускає саму гру, створюючи об’єкт класу Board.

У класі Board наявні такі методи:

* конструктор Board

Цей конструктор:

* додає фонове зображення;
* створює нову падаючу фігурку та запускає таймер;
* створює масив фігурок;
* очищує всю дошку перед початком гри й додає «слухач» кнопок.
* int squareWidth()/squareHeight()/imgWidth()/imgHeght()

Ці методи просто повертають розміри для шматочків фігури\фонового зображення.

* TetroShapes shapeAt()

Повертає значення з нашого масив «дошка» за координатами, отриманими з параметрів методу;

* cleanAll()

Очищує всю «дошку», присвоюючи кожній її клітинці порожню фігурку з enum;

* pieceDropped()

Пересуває падаючу фігурку донизу, видаляє повні рядки, перевіряє чи закінчила падати попередня фігурка;

* newPiece()

Створює нову рандомну фігурку, ініціалізує її наверху дошки, перевіряє чи є взагалі можливість «скинути» цю нову фігурку, якщо ні – завершує гру через переповнення рядків дошки;

* oneLineDown()

Перевіряє чи є можливість посунити фігурку вниз, якщо є, переміщує її на один рядок униз;

* actionPerformed()

Перевіряє чи закінчила падати фігурка, якщо так – створює нову, якщо ні – пересуває поточну на один рядок донизу;

* drawSquare()

Промальовує самі фігурки за допомогою квадратиків, створює для них дизайн;

* paint()

Переписаний метод, до нього додають промалювання цих квадратиків за певних умов;

* start()

Метод, що запускає гру. Якщо гра на паузі, то нічого не робить. Якщо можна посунити фігурку, починає гру, очищує дошку, запускає таймер;

* pause()

Ставить гру на паузу, зупиняє таймер. Якщо гра вже на паузі, навпаки запускає її;

* tyMove()

Метод, що на основі параметрів перевіряє чи є можливість посунути фігурку, чи не «вдариться» вона в стінку й т.д.;

* deleteFullRows()

Метод, що «прибирає» повні рядки, додає за це бали, та посуває старі рядки на місце нових;

* dropDown()

Роняє фігурку на самий низ за один хід.

# Інструкція для користувача

## На початку гри користувача відкривається вікно, і одразу починає падати перша рандомна фігурка. Щоб рухати її або повертати, користувач повинен натискати стрілочки на клавіатурі комп’ютера.Изображение выглядит как снимок экрана Описание создано с очень высокой степенью достоверности

## Рис. 3 «Робота програми»

## За кожен знищений рядок користувач отримує бали.

## Изображение выглядит как снимок экрана Описание создано с высокой степенью достоверности Рис. 3.1 «Робота програми»

Якщо всі рядки заповнені, й нова фігура не має місця куди випасти, то гра завершується.

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

## Рис. 3.2 «Робота програми»

Якщо натиснути на клавішу p то гра зупиниться на паузу, продовжити гру можна так само натиснувши на клавішу p.

Изображение выглядит как снимок экрана

Описание создано с очень высокой степенью достоверности

## Рис. 3.3 «Робота програми»

# Висновок 1

Під час роботи над цим проектом я навчилася користуватися різним бібліотеками Java для досягнення своїх цілей. Вперше використала бібліотеки Swing, навчилася працювати з ними, створюючи застосунки для комп’ютера за допомогою Swing та малюючи графіку за допомогою awt.

# Висновок 3

Під час роботи над цим проектом я навчилася користуватися різним бібліотеками Java для досягнення своїх цілей. Вперше використала бібліотеки Swing та JFeeChart, навчилася працювати з ними, створюючи застосунки для комп’ютера за допомогою Swing та малюючи графіки за допомогою JFeeChart. Також я вдосконалила свої навички роботи з бібліотекою awt, навчившись обробляти різні види дій (з бібліотеки java.awt.event.\*).