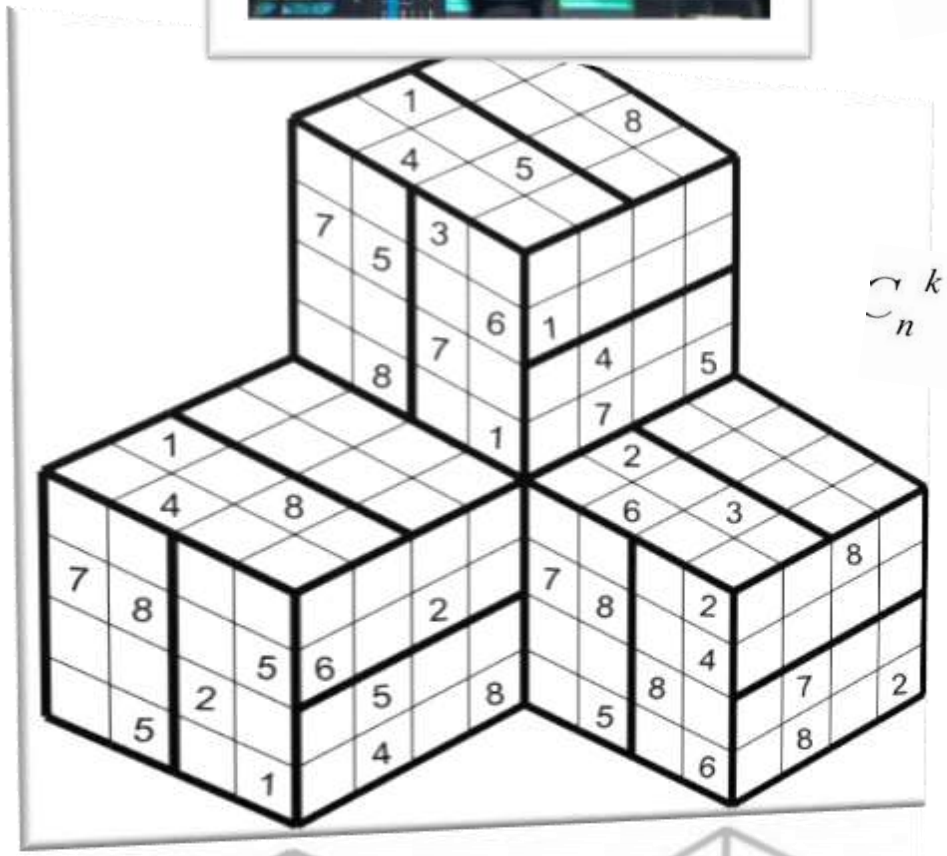
$$A_n^k = \frac{n!}{(n-k)!}$$



$$C_n^k = \frac{n!}{(n-k)!\cdot k!}$$

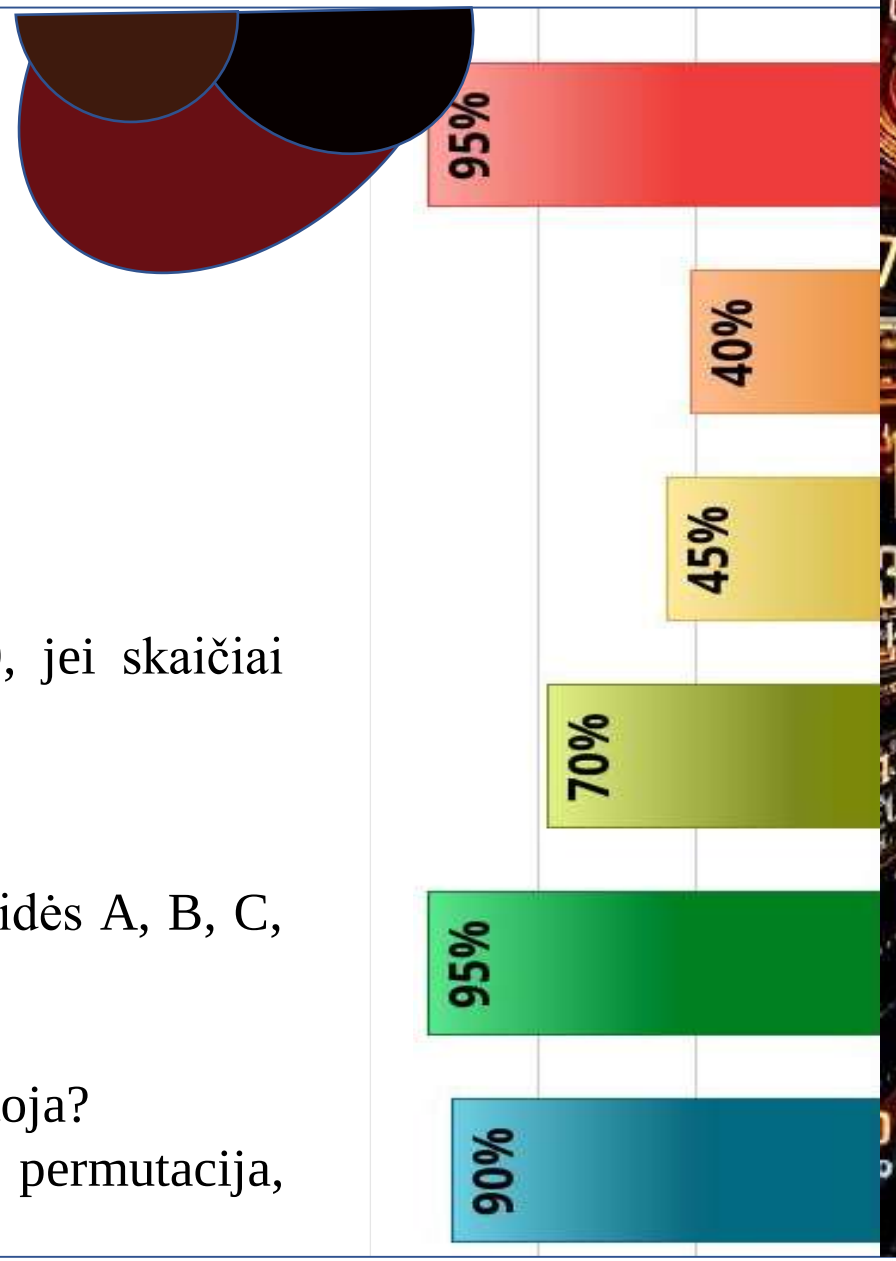
$$P_n = n!$$

„TVARKA CHAOSO VIDUJE: KOMBINATORIKA IR ALGORITMAI SUDOKU LAUKE“

Visagino technologijos ir verslo profesinio mokymo centras
Matematikos mokytoja: *Tetiana Yanchukovych*

Visaginas 2025

Ankstesnės medžiagos apžvalga - testas „Pakartokime kombinatoriką“



Wordwall Kahoot! Google Forms

10 klausimų:

1. Kiek yra 5 elementų permutacijų?
2. Kiek yra 4 x 2 aranžuočių be pakartojimų?
3. Kuo skiriasi permutacijos ir kombinacijos?
4. Kiek triženklių kodų galima sudaryti iš skaičių nuo 1 iki 9, jei skaičiai nesikartoja?
5. Kiek yra būdų, kaip iš 10 pasirinkti 3 studentus budėjimui?
6. Keliais eilėmis prie apvalaus stalo gali susodinti 6 žmones?
7. Kiek yra slaptažodžių, sudarytų iš 4 simbolių ir kuriuose yra raidės A, B, C, jei šiuos simbolius galima kartoti?
8. Kiek yra būdų sudaryti 8 žmonių eilę?
9. Kiek yra 4 skaitmenų PIN kodo variantų, jei skaitmenys nesikartoja?
10. Kokio tipo kombinatorika naudojama Sudoku sprendimui – permutacija, dėjimas ar derinys?

Kombinatorika realiame gyvenime

Slaptažodžių kūrimas:

6 skaitmenų skaitmeninis slaptažodis: $10^6 = 1\,000\,000$ parinktys.

Slaptažodis iš 6 lotyniškos abėcėlės raidžių be pasikartojimų:

$26 \times 25 \times 24 \times 23 \times 22 \times 21 = 165\,765\,600$ parinktys.

Loterija „6 iš 49“:

Derinių skaičius: $C_{49}^{6} = \frac{49!}{6!(49-6)!} = 13\,983\,816$

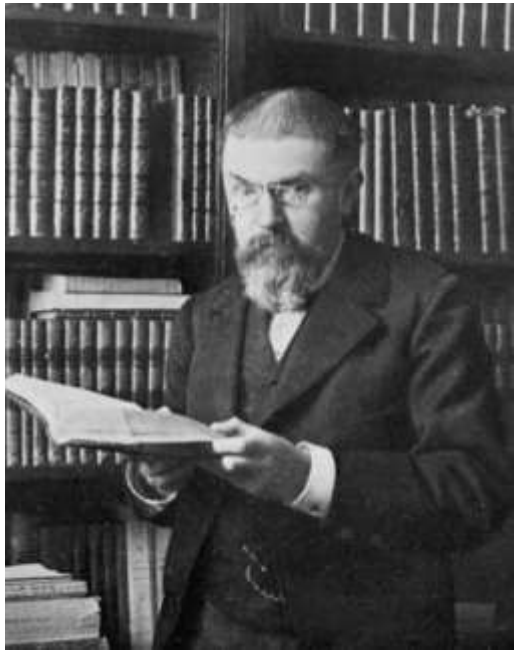
Tvarkaraščio planavimas:

Kiek būdų galima išdėstyti 5 skirtingas pamokas 5 laiko intervalais: 5! = 120

Porų kodavimas programuojant:

Kiek porų galima sudaryti iš 8 dalyvių: $C_8^2 = 28$

„Matematika
ne tik skaitmeniniai
skaičiavimai, tai ir
chaoso įveikimo
menas"



(Henri Poincaré)

9 realaus gyvenimo Sudoku galvosūkiai, kurių kiekvienas naudoja kombinatorikos formules: permutacijas, išdėstymus, derinius ir apribojimų principus

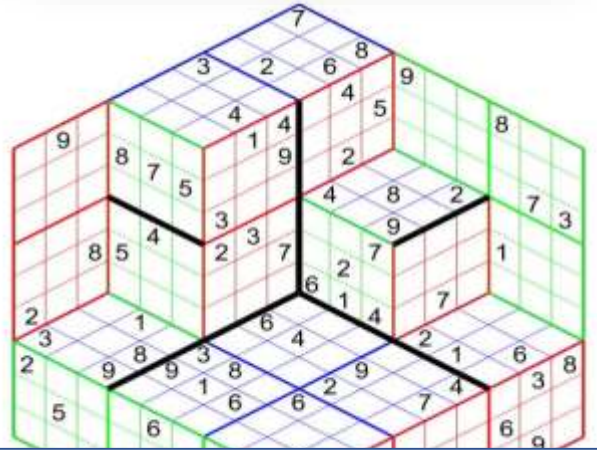
1 uždavinys. Kiek būdų yra užpildyti vieną Sudoku eilutę?

Sąlyga: Sudoku eilutėje yra 9 langeliai, reikia sudėti skaičius nuo 1 iki 9 be pasikartojimų.

2 uždavinys. Kiek yra būdų, kaip pasirinkti 3 skaičius iš 9 ir išdėstyti juos 3 eilutės langeliuose?

Sąlyga: 3 eilutėje yra tuščių langelių, reikia pasirinkti ir išdėstyti skaičius be pasikartojimų.

3 uždavinys. Kiek būdų galima išdėstyti du vienodus skaičius (pavyzdžiui, 5 ir 5) dviejuose skirtinguose 3x3 blokuose nepažeidžiant eilučių ir stulpelių taisyklių?



4 uždavinys. Kiek būdų yra pasirinkti 17 langelių iš 81 ir užpildyti juos užuominomis?

Sąlyga: Kiek būdų yra pasirinkti 17 pradinių užuominų (nepriklausomai nuo jų verčių)?

5 uždavinys. Kiek būdų yra įrašyti 2 skaičius į 81 langelį, jei skaičiai eilutėje nesikartoja?

Sąlyga: Pasirinkite 2 skirtingus skaičius ir išdėliokite juos taip, kad jie kiekvienoje eilutėje pasirodytų ne daugiau kaip vieną kartą.

6 uždavinys. Kiek yra būdų, kaip pasirinkti 3 nesikertančias eilutes ir jose įrašyti tą patį skaitmenį (pavyzdžiui, 7), nepažeidžiant kartojimo stulpeliais taisyklės?

7 uždavinys. Kiek yra būdų pasirinkti 3 skirtingus skaičius ir sudėti juos į vieną 3x3 bloką be pasikartojimų eilutėse ir stulpeliuose?

8. uždavinys. Kiek yra galimų Sudoku užpildymo būdų, atsižvelgiant į visus apribojimus?

9 uždavinys. Kiek yra būdų pasirinkti 3 skirtingas eilutes ir kiekvienoje iš jų tame pačiame stulpelyje įrašyti tą patį skaičių (pavyzdžiui, 4)?

„Gera problema yra ta, kurią norima išspręsti."

(Georgas Kantoras)

Kombinatorika gyva ne tik vadovėliuose. Laikas parodyti ją veiksmais – programavimo kodu!

Informatikoje kombinatorikos principai sudaro pagrindą:

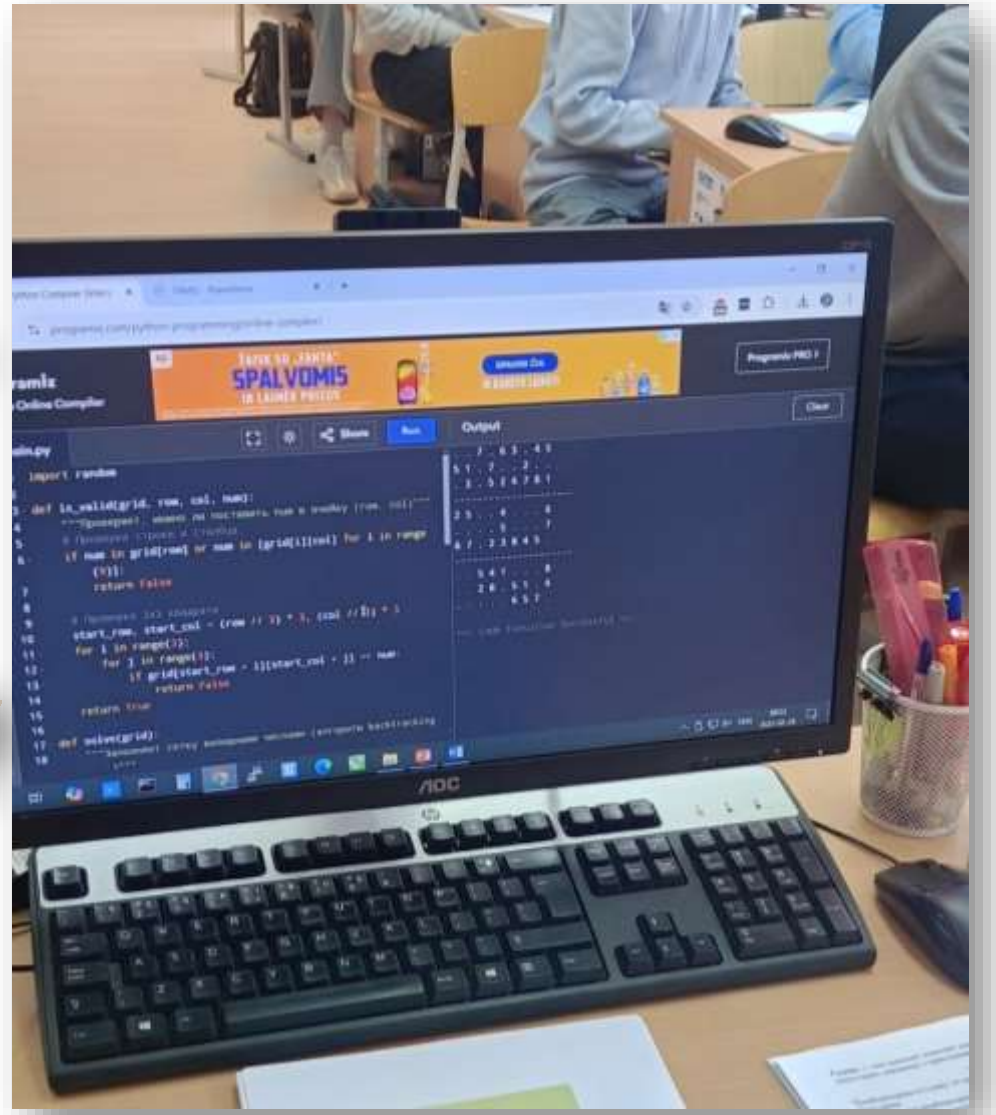
- ❑ išsamios paieškos (brute force) algoritmams;
- ❑ kombinacijų generavimui;
- ❑ sprendimų paieškai optimizavimo uždaviniuose ir net dirbtinio intelekto sistemų kūrimui.

Sudoku generavimas – ryškus pavyzdys, kur taikomi:

- ❖ paieškos algoritmai;
- ❖ rekursija;
- ❖ atsitiktinis pasirinkimas – visa tai padeda modeliuoti žmogaus mąstymą: greitai, tiksliai ir logiškai.

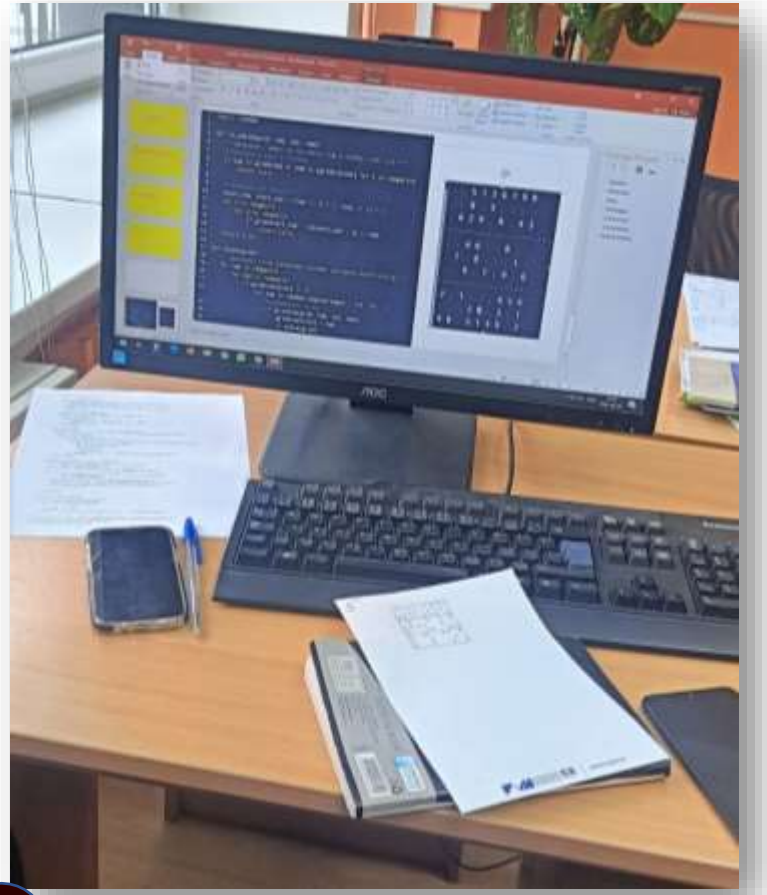
FINALINĖ UŽDUOTIS: „SUDOKU MŪŠIS“

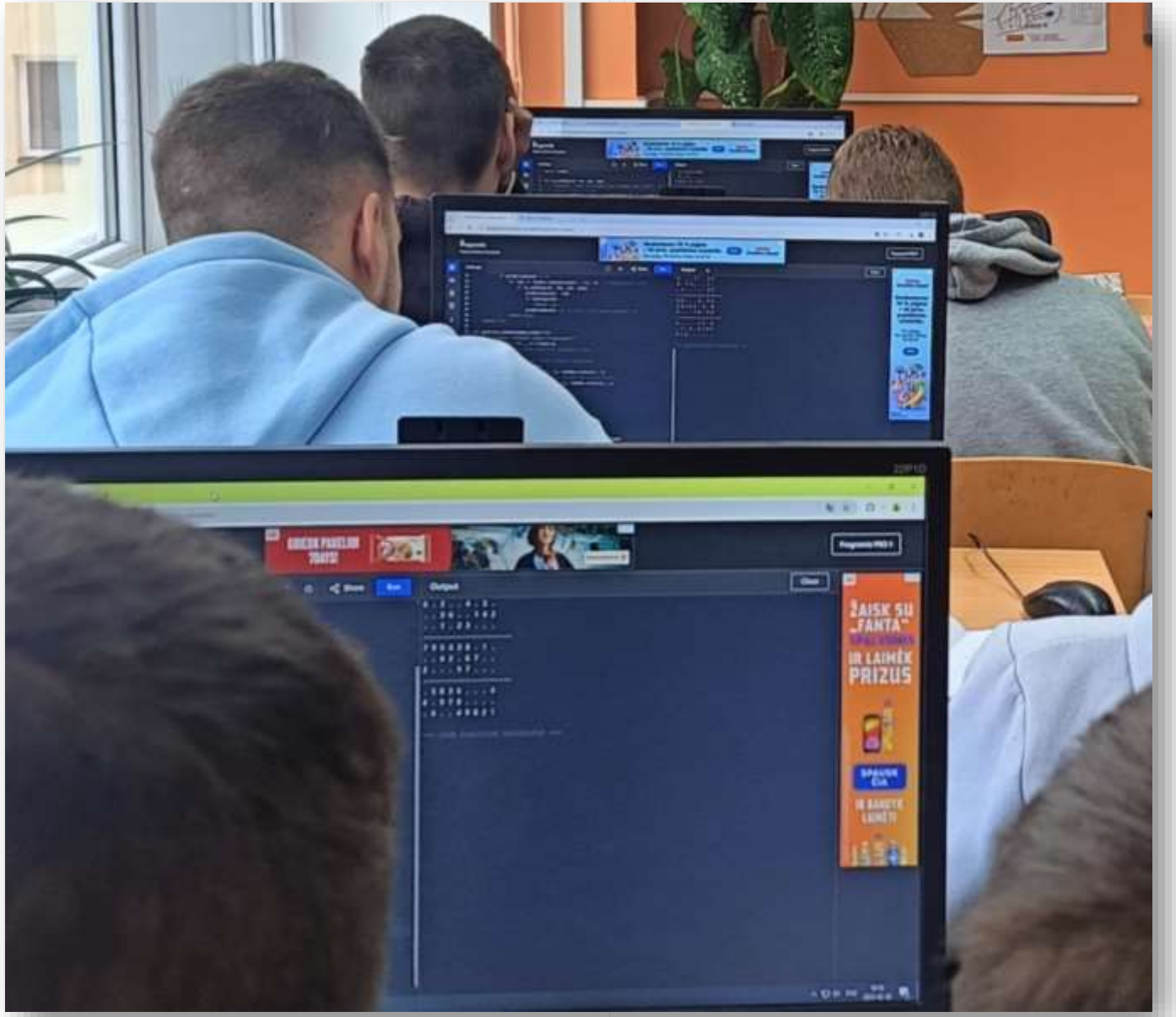
- Kiekviena dviejų mokinių komanda gauna neužpildytą Sudoku lentelę.
- **Užduotis:** užpildyti lentelę, naudojant kombinatorikos metodus ir parašytą kodą, skirtą kitos komandos sprendimui tikrinti.
- **Laimi ta komanda, kuri pirmoji teisingai užpildo lentelę.**



```
import random

def is_valid(grid, row, col, num):
    """Проверяет, можно ли поставить num в ячейку (row, col)"""
    # Проверка строки и столбца
    if num in grid[row] or num in [grid[i][col] for i in range(9)]:
        return False

    # Проверка 3x3 квадрата
    start_row, start_col = (row // 3) * 3, (col // 3) * 3
    for i in range(3):
        for j in range(3):
            if grid[start_row + i][start_col + j] == num:
                return False
    return True

def solve(grid):
    """Заполняет сетку валидными числами (алгоритм backtracking)"""
    for row in range(9):
        for col in range(9):
            if grid[row][col] == 0:
                for num in random.sample(range(1, 10), 9):  # Перемешиваем числа
                    if is_valid(grid, row, col, num):
                        grid[row][col] = num
                        if solve(grid):
                            return True
                        grid[row][col] = 0  # Откат, если дальше решения нет
                return False
    return True

def generate_sudoku(empty_cells=40):
    """Генерирует валидную судоку-головоломку"""
    grid = [[0] * 9 for _ in range(9)]
    solve(grid)  # Заполняем полностью правильное поле

    # Убираем случайные числа, чтобы создать головоломку
    for _ in range(empty_cells):
        row, col = random.randint(0, 8), random.randint(0, 8)
        while grid[row][col] == 0:  # Не убираем уже пустые клетки
            row, col = random.randint(0, 8), random.randint(0, 8)
        grid[row][col] = 0

    return grid

def print_sudoku(grid):
    """Выводит судоку красиво"""
    for i, row in enumerate(grid):
        if i % 3 == 0 and i != 0:
            print("-" * 21)
        print(" ".join(str(num) if num != 0 else "." for num in row))

# Генерация судоку с 40 пустыми клетками (средняя сложность)
sudoku_puzzle = generate_sudoku(40)
print_sudoku(sudoku_puzzle)
```

FINALINĖ UŽDUOTIS: „SUDOKU MŪŠIS“



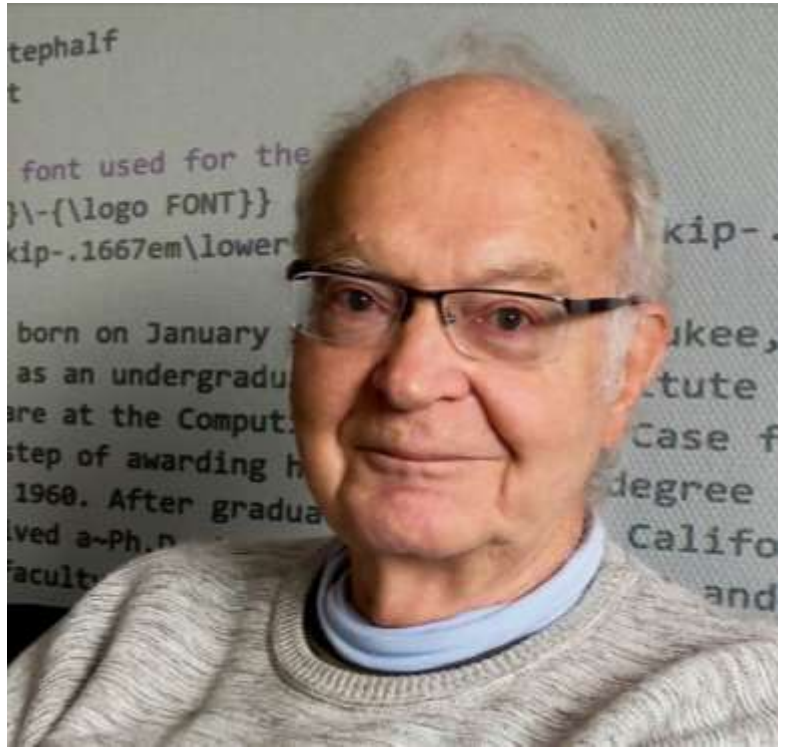


Apibendrinkime

Šios pamokos parodė, kad:

- Matematikos formulės gali atgyti per programavimą ir realias užduotis.
- Algoritminis mąstymas padeda rasti sprendimus ten, kur reikia logikos, struktūros ir strategijos.
- Mokymasis gali būti žaidimas ir iššūkis – tai motyvuoja ir įkvepia.
- STEAM metodas – tai ne tik mokslo integracija, bet ir kūrybinis požiūris į žinias.
- Tai buvo žingsnis nuo teorijos prie kūrybiško pritaikymo!

„Kiekviename algoritme slypi estetikos elementas.“



(Donaldas Knutas)

Ačiū už dėmesį!