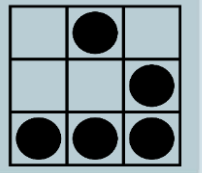
ВИКОРИСТАННЯ МОЖЛИВОСТЕЙ VBA ТАБЛИЧНОГО ПРОЦЕСОРА MS EXCEL ДЛЯ НАПИСАННЯ ГРИ

ЖИТТЯ

Кам'янський еколого-економічний ліцей
Кам'янської міської ради Черкаської області

Анотація



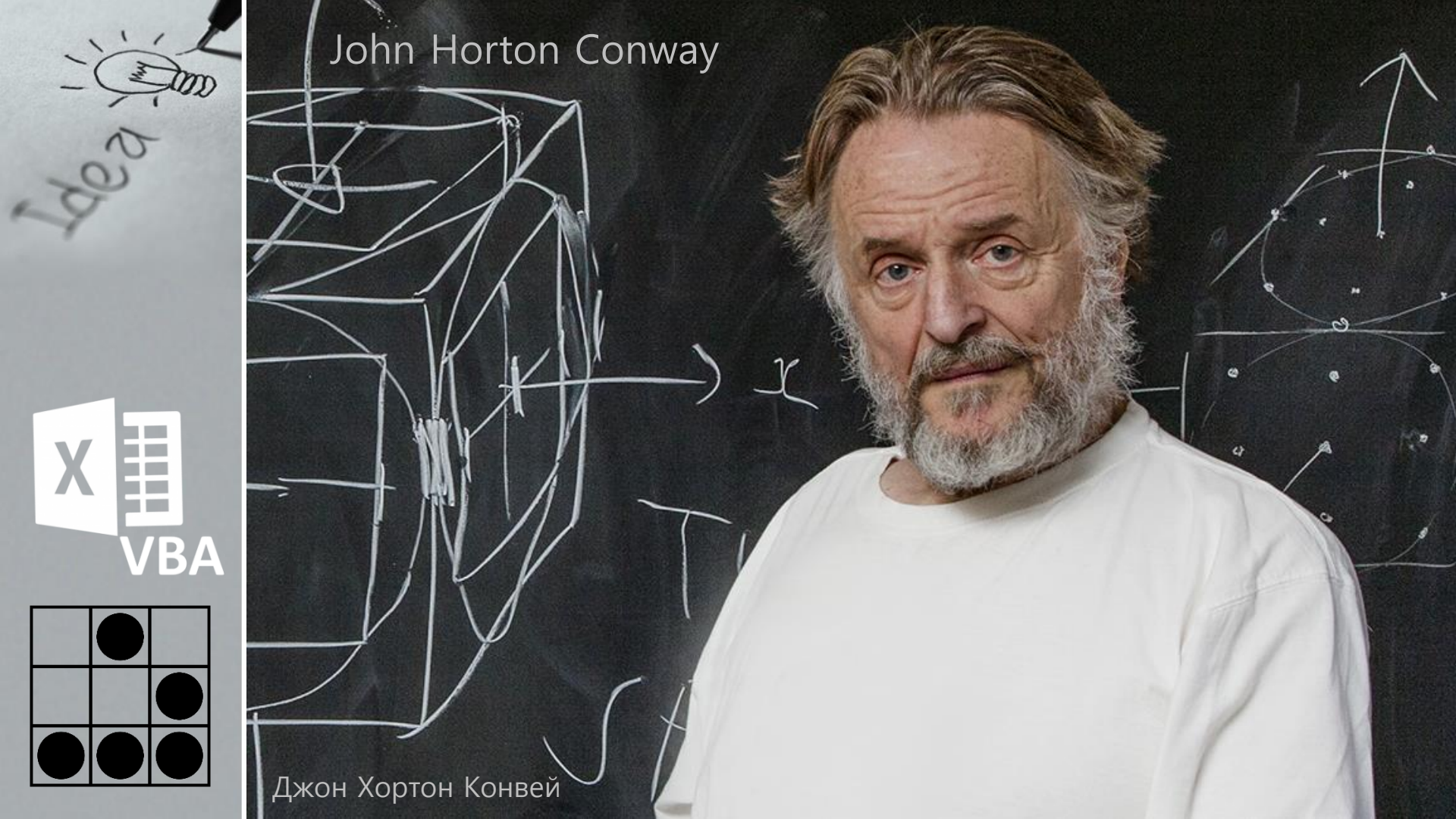
Visual Basic for Application

Microsoft

В даній роботі розглядається технологія VBA табличного процесора MS Excel на прикладі розробки ігрового додатку «Життя».

VBA – мова програмування (розшифровується як Visual Basic for Application) була розроблена компанією Microsoft. Дана мова не являється самостійною, а призначена лише для автоматизації процесів в пакеті MS Office. VBA широко використовується в Excel, я також в Access, Word та інших програмах пакету.

Гра «Життя» – одна з найпопулярніших математичних ігор. Цю гру винайшов англійський математик Джон Хортон Конвей в 1970 році. Про захопливість гри свідчать результати безлічі цікавих досліджень і численних комп'ютерних реалізацій. Гра Конвея "Життя" – типовий приклад клітинного автомата, як математичного об'єкта, що представляє собою дискретну динамічну систему. По суті клітинні автомати є синтетичними світами, поведінка яких здебільшого визначається простими локально діючими правилами.

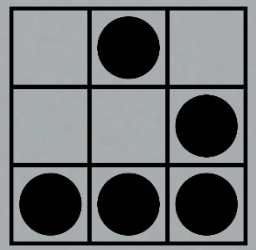

John Horton Conway
Idea
VBA
Джон Хортон Конвей

Актуальність теми

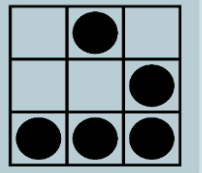
Застосування клітинних автоматів

Клітинні автомати широко застосовуються при моделюванні систем, для яких важлива просторова взаємодія між елементами системи. Існує багато прикладів таких моделей в біології, інформатиці (включаючи системи телекомунікацій). У фізиці клітинні автомати застосовують для аналізу явищ перенесення (теплопровідності, дифузії і в'язкості), моделювання твердого тіла.

Актуальність теми: враховуючи сучасні процеси інформатизації суспільства і пов’язані з ним тенденції в освіті, постає актуальна проблема створення і використання в освітньому процесі навчальних комп’ютерних програм, спрямованих на гармонійний розвиток особистості. Застосовуючи гру «Життя», можна із задоволенням розвивати логічне та аналітичне мислення, інтуїцію, здатність концентрувати увагу і загалом покращувати свій рівень математичних здібностей.



Дослідження



Об'єкт і предмет дослідження

Об'єкт дослідження – процес створення програмного продукту за допомогою технологій VBA табличного процесора MS Excel.

Предмет дослідження – створення гри «Життя» мовою VBA табличного процесора MS Excel.

Методологія програмування проекту заснована на уявленні програми у вигляді сукупності об'єктів (ігрове поле, клітина), кожен з яких є реалізацією певного типу, що використовує механізм пересилання повідомлень.

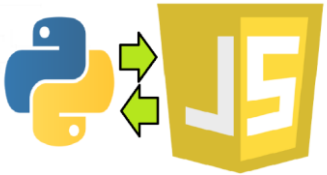
Застосування VBA табличного процесора MS Excel при розробці даного проекту включає в себе об'єктно–орієнтований аналіз предметної області задачі та об'єктно–орієнтоване програмування.

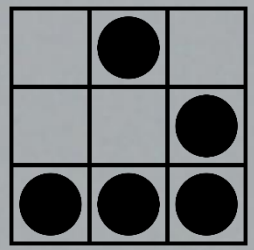
Дослідження

Методи, практичне значення,

приклади реалізації в інших середовищах

Calc, JavaScript, Python





Методи дослідження:

• **теоретичні**: аналіз науково–технічної і спеціалізованої літератури з проблеми дослідження, наявних ПЗ, аналіз інформаційних матеріалів глобальної мережі Інтернет; систематизація та узагальнення матеріалу;

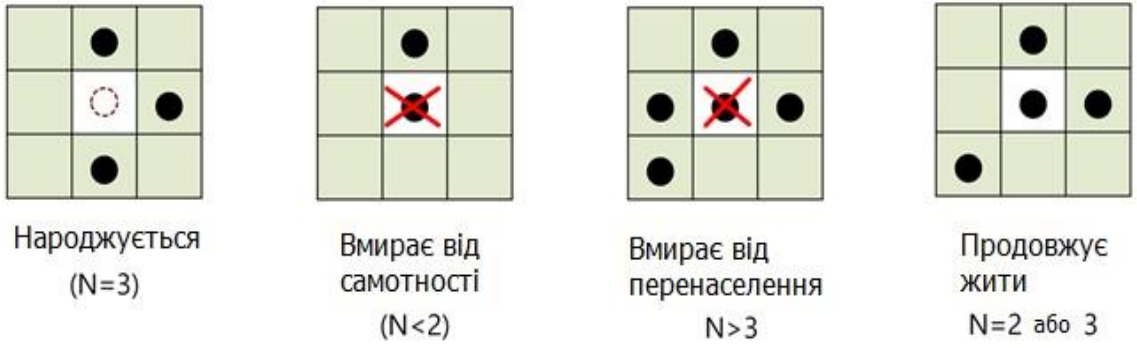
• **емпіричні**: спостереження й аналіз проведених експериментів і процесів.

Практичне значення дослідження: процес дослідження мови VBA табличного процесора MS Excel для створення програми, проектування архітектури гри, аналіз вимог до неї, реалізації та тестування програми.

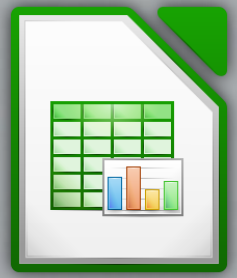
Робота над ресурсом проводилась у декілька етапів: вибір теми й опис проблеми; постановка задачі; аналіз об'єкту; розробка алгоритму й синтез моделі; форма подання інформації й вибір програмних продуктів; синтез комп'ютерної моделі об'єкта.

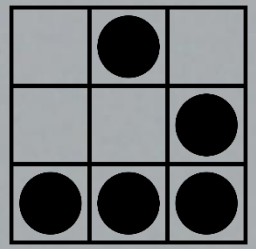
Правила гри

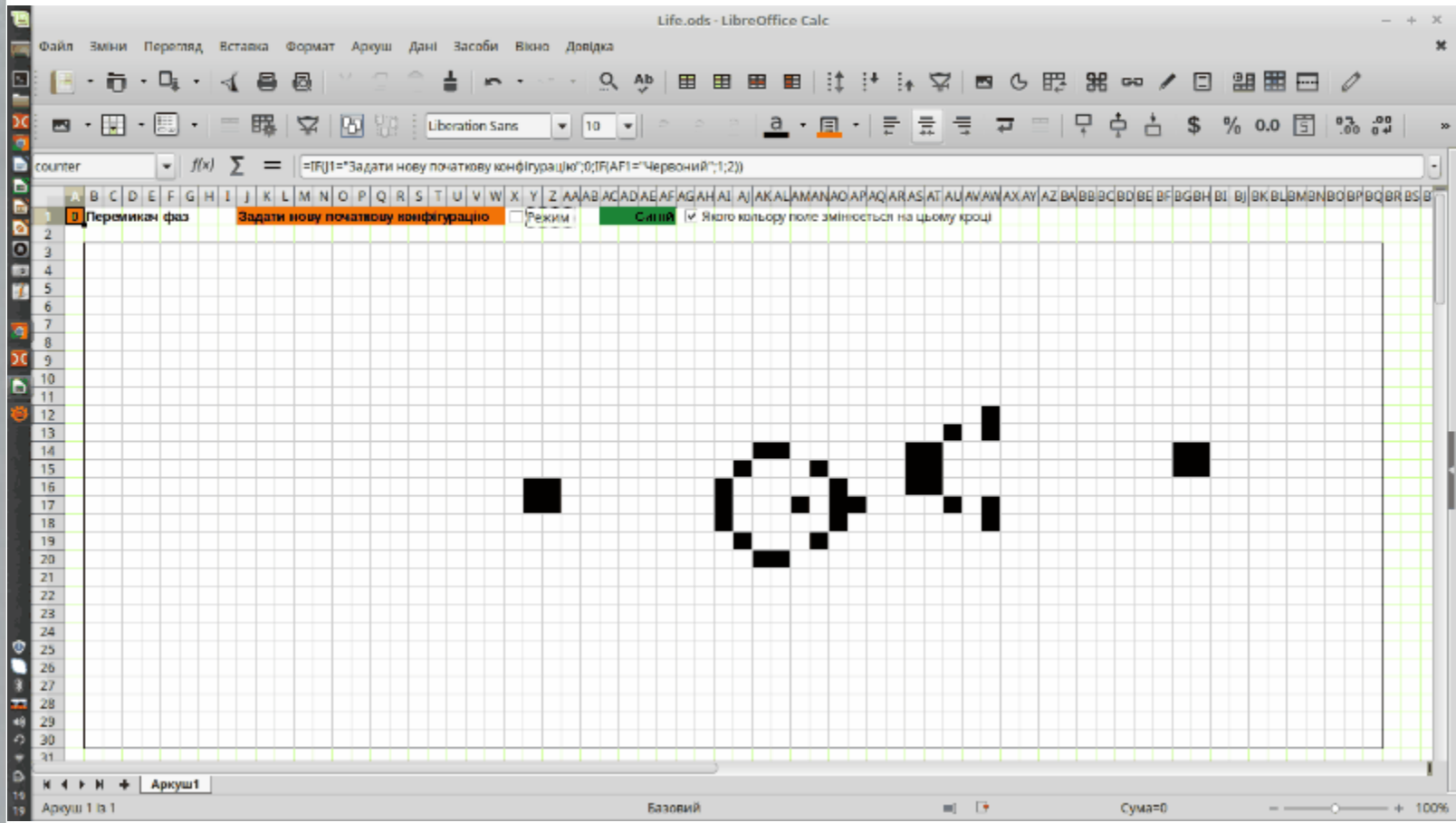
- якщо у живої клітини два чи три сусіди – то вона лишається жити;
- якщо у живої клітини один чи немає сусідів – то вона помирає від «самотності»;
- якщо у живої клітини чотири та більше сусідів – вона помирає від «перенаселення»;
- якщо у мертвої клітини рівно три сусіди – то вона оживає.



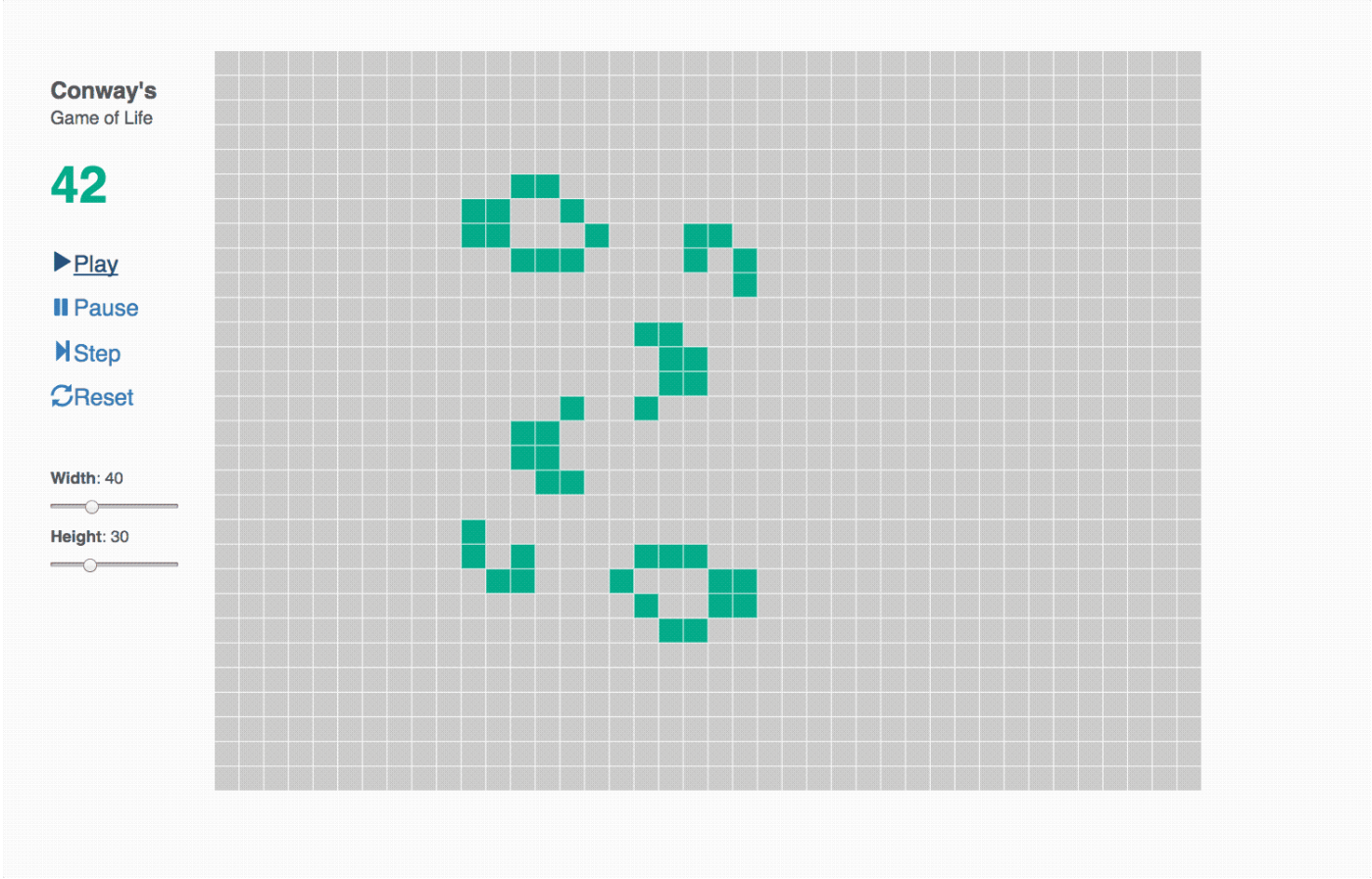
Game of Life Calc LibreOffice

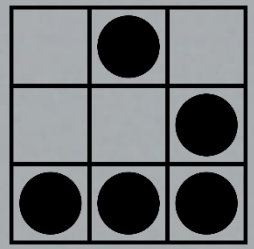






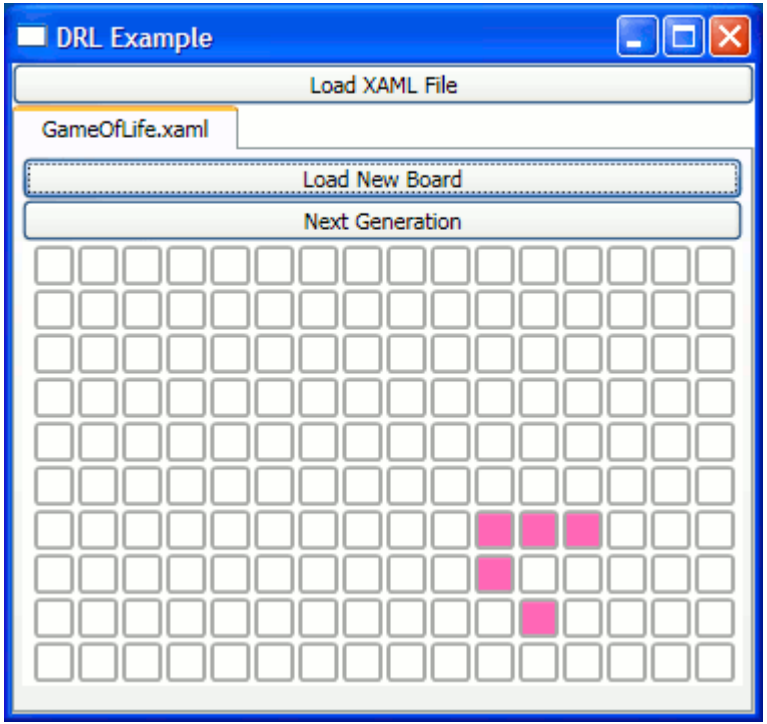
Game of Life JavaScript

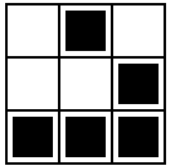




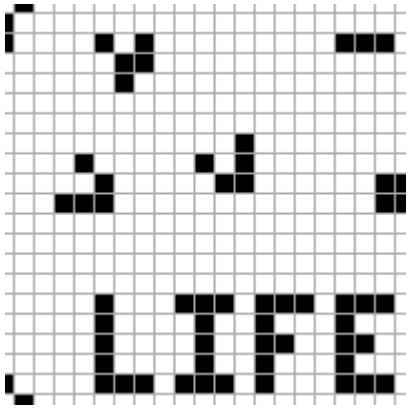
Game of Life Python

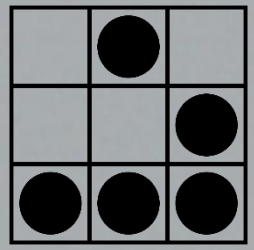
John Conway's Game of Life in XAML/WPF using embedded Python



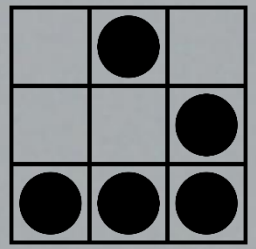


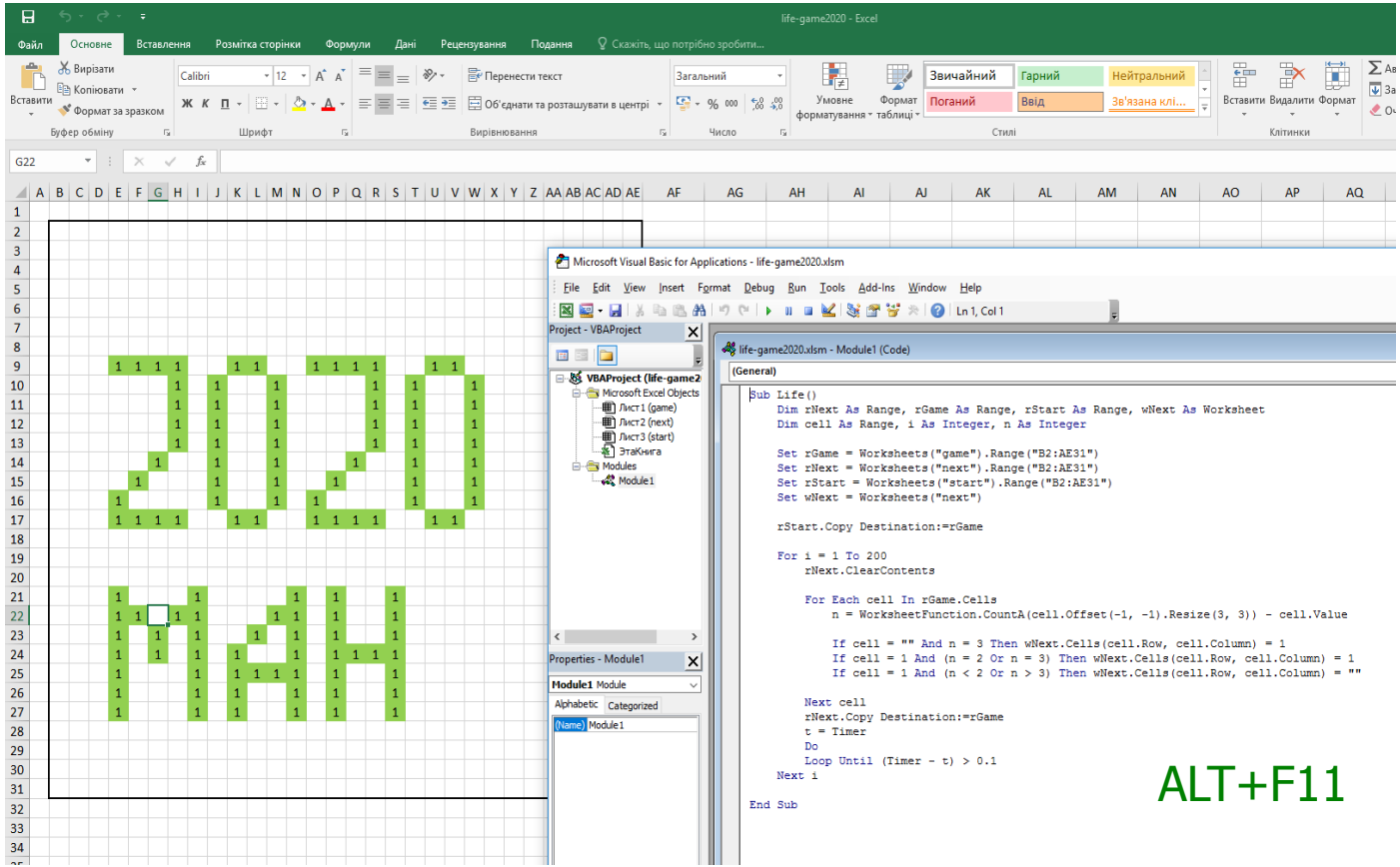
THE **GAME OF LIFE**



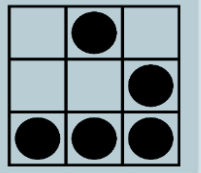


ПИШЕМО ГРУ "ЖИТТЯ" НА VBA В EXCEL

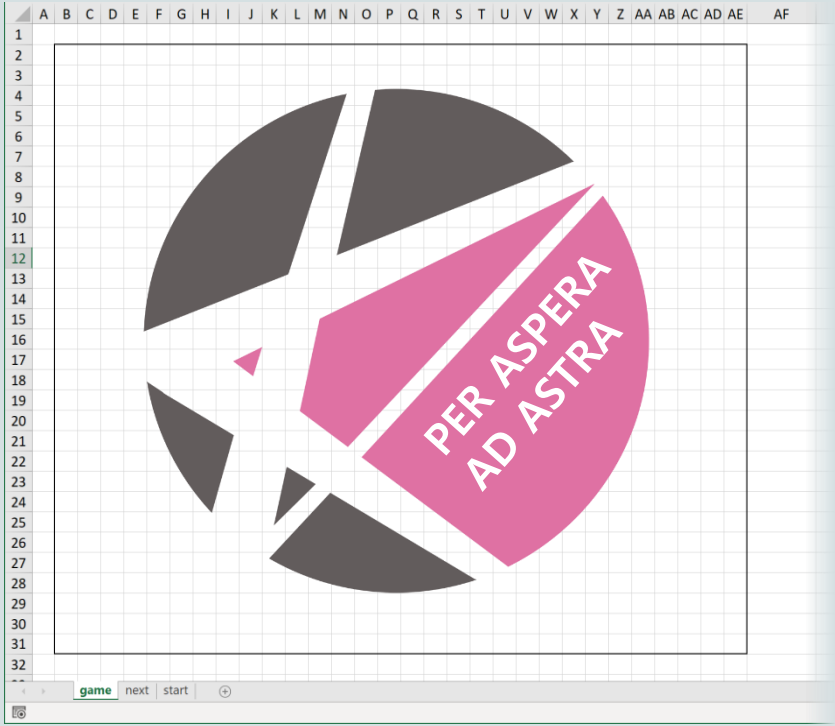




ПИШЕМО ГРУ "ЖИТТЯ"



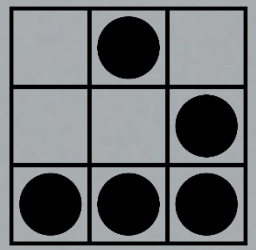
Готуємо ігровий простір



Створимо в новій книзі три аркуші:

- **game** - це буде основний аркуш гри, де ми будемо спостерігати за розвитком нашої "колонії"
- **next** - цей аркуш буде формувати наступне покоління, яке потім прийде на зміну поточного
- **start** - на цьому аркуші ми будемо задавати початкову конфігурацію, тобто перше покоління в нашій грі

.



У вікні редактора **Visual Basic** створимо новий модуль за допомогою команди меню **Insert** - **Module**, а потім напишемо туди код макросу:

```
Sub Life()
    Dim cell As Range, n As Integer, i As Integer
    Set rGame = Worksheets("Game").Range("B2:AE31")
    Set rStart = Worksheets("Start").Range("B2:AE31")
    Set rNext = Worksheets("Next").Range("B2:AE31")
    Set wNext = Worksheets("Next")
    rStart.Copy Destination:=rGame
 For i = 1 To 50
  rNext.ClearContents
  For Each cell In rGame.Cells
  n = WorksheetFunction.CountA(cell.Offset(-1, -1).Resize(3, 3)) - cell.value
  If cell = "" And n = 3 Then wNext.Cells(cell.Row, cell.Column) = 1
  If cell = 1 And (n = 2 Or n = 3) Then wNext.Cells(cell.Row, cell.Column) = 1
  If cell = 1 And (n < 2 Or n > 3) Then wNext.Cells(cell.Row, cell.Column) = ""
  Next cell
  rNext.Copy Destination:=rGame
  Next i
End Sub
```

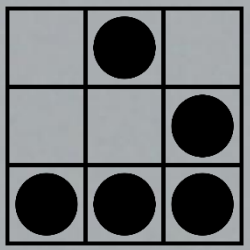
Розберемо код

Оскільки в коді нам доведеться кілька разів посилатися і багато раз працювати з діапазонами ігрового простору (B2: AE31) на кожному з трьох аркушів книги, то має сенс відразу оформити їх як змінні. Це робиться в блоці:

```
Set rGame = Worksheets ( "Game"). Range ( "B2: AE31")
Set rStart = Worksheets ( "Start"). Range ( "B2: AE31")
Set rNext = Worksheets ( "Next"). Range ( "B2: AE31")
```

Заодно ми створюємо ще й змінну **wNext**, яка посилається на весь аркуш **next** цілком - це нам теж стане в нагоді в майбутньому:

```
Set wNext = Worksheets ( "Next")
```



Розберемо код

Потім, перед початком гри, ми повинні перенести перше покоління з аркушу **start** на аркуш **game**. Це виконуємо командою прямого копіювання з використанням вже створених змінних:

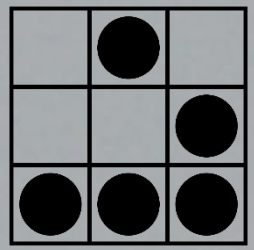
```
rStart.Copy Destination: = rGame
```

Оскільки ми хочемо прокрутити в нашій грі не одне, а кілька (наприклад, 50 для початку) поколінь, то подальші дії укладені в цикл:

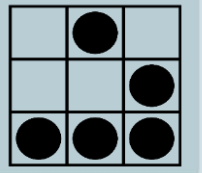
```
For i = 1 to 50
...
Next i
```

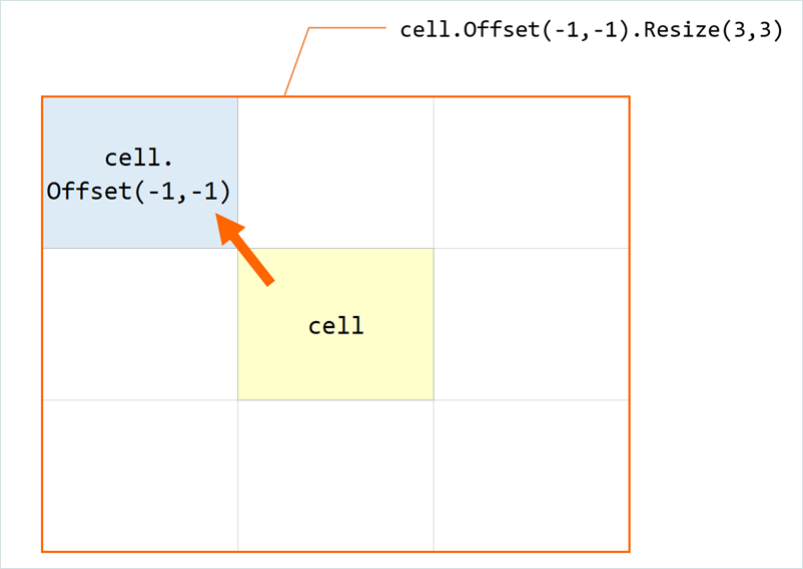
А всередині цього циклу ми, по-перше, спочатку очищаємо робочий простір на аркуші **next** для формування наступного покоління:

```
rNext.ClearContents
```



Розберемо код





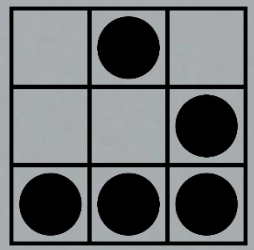
А, по-друге, запускаємо вкладений цикл проходу по всім осередкам ігровий всесвіту на аркуші **game**, щоб перевірити кожну з них - це реалізовано циклом проходу по колекції:

```
For Each cell in rGame.Cells
...
Next cell
```

Посилання на чергову перевіряється осередок буде зберігатися у змінній cell. Для цього осередку нам потрібно спочатку побудувати околиця 3x3 з нею в середині. Це виконується за допомогою конструкції:

```
cell.Offset (-1, -1) .Resize (3, 3)
```

Тут метод Offset (-1, -1) віртуально зрушує поточну перевіряється осередок на один рядок вгору і на один стовпець вліво, а потім метод Resize (3,3) знову ж віртуально розтягує цю одну клітинку до нових розмірів 3 на 3:



Розберемо код

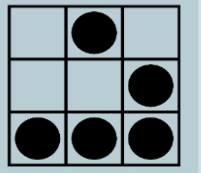
Щоб порахувати кількість заповнених осередків в отриманої околиці застосовується функція робочого аркушу СЧЁТЗ (COUNTA), яку в VBA можна викликати за допомогою об'єкта WorksheetFunction. Таким чином кількість живих сусідів в навколишньому поточну комірку області 3 на 3 ми отримуємо виразом (не забувши відняти з отриманої кількості поточну комірку):

```
n = WorksheetFunction.CountA (cell.Offset (-1, -1) .Resize (3, 3)) - WorksheetFunction.CountA (cell)
```

Далі нам потрібно перевірити отриману кількість сусідів і позначити на аркуші наступного покоління поточну комірку як живу або мертву, згідно з правилами гри. Це виконує блок з трьох перевірок:

```
If cell = "" And n = 3 Then wNext.Cells (cell.Row, cell.Column) = 1
If cell = 1 And (n = 2 Or n = 3) Then wNext.Cells (cell.Row, cell.Column) = 1
If cell = 1 And (n <2 Or n> 3) Then wNext.Cells (cell.Row, cell.Column) = ""
```

Game of Life

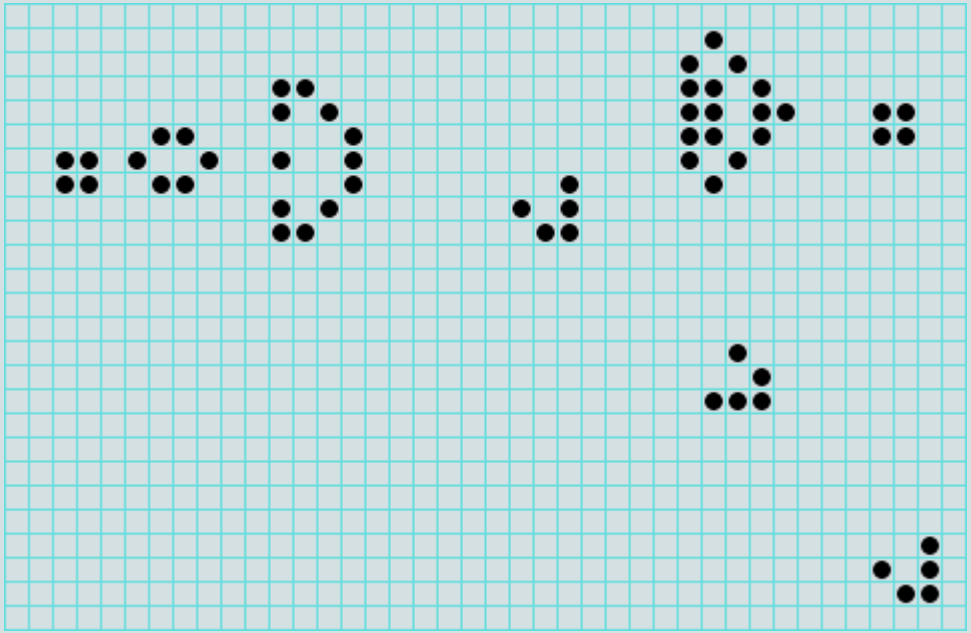


Коли цикл проходу по осередках буде завершено, то сформоване наступне покоління з аркушу **next** потрібно скопіювати на місце поточного на аркуші **game** - робимо це вже знайомої конструкцією:

```
rNext.Copy Destination: = rGame
```

Ось, власне, і вся логіка.

Залишилося повернутися в Excel на аркуш **game**, запустити нашу гру через вкладку Розробник - Макроси (Developer - Macro) і насолодитися процесом розвитку нашої колонії…



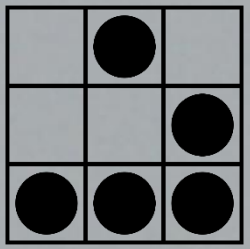
ВИСНОВКИ

В даній роботі розглянуто створення гри «Життя» мовою VBA табличного процесора MS Excel.

У першому розділі було надане поняття про клітинні автомати та розглянуті їх способи класифікації. Також, були приведені приклади застосування клітинних автоматів на практиці.

У другому розділі були розглянуті правила гри, підготовка ігрового простору, написання макросу.

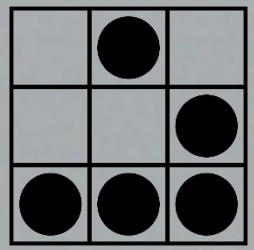
Створений програмний продукт було апробовано для дослідження фігур клітинного автомату "Гра життя"

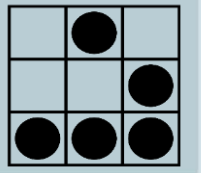


ВИСНОВКИ

Симулятори життя можуть фокусуватися на біологічному чи соціальному аспекті життя. Перші дозволяють гравцеві експериментувати с генетикою живих істот, моделюють екологічні системи, особливе місце серед них займають симулятори еволюції. Другі грунтуються на соціальній взаємодії між живими істотами – їх спілкуванні, роботі.

Застосовуючи гру «Життя», можна розвивати логічне та аналітичне мислення, інтуїцію, здатність концентрувати увагу і загалом покращувати свій рівень математичних здібностей.





ДЯКУЮ ЗА УВАГУ

VBA