

ТЕХНОЛОГІЯ СТВОРЕННЯ ЛЕКЦІЙНИХ ДЕМОНСТРАЦІЙ ДЛЯ ММС «ВИЩА МАТЕМАТИКА»

К. І. Словак¹, М. В. Попель²

¹ Україна, м. Кривий Ріг, Криворізький економічний інститут Київського національного економічного університету імені Вадима Гетьмана

² Україна, м. Кривий Ріг, Криворізький державний педагогічний університет

Slovak_kat@mail.ru

Провідним засобом наочності у ММС «Вища математика» є *лекційні демонстрації* – комп’ютерні моделі з графічним інтерфейсом і напівавтоматичним режимом управління, що ілюструють теоретичні поняття, теореми, методи тощо [1]. Використання і дослідження таких моделей надає можливість покращити розуміння математичної, фізичної чи економічної сутності методів та алгоритмів; глибше усвідомити новий матеріал та створити змістову основу для розв’язання прикладних задач, а також сприяє підвищенню пізнавальної активності через наочність [2].

Лекційні демонстрації передбачають багаторазове виконання обчислень для різних значень вхідних параметрів, тому при їх розробці доцільно використати візуальні елементи управління типу «поле для введення», «повзунок», «прапорець», «меню вибору» для створення яких використовують відповідні функції СКМ Sage [3].

Визначення кожного елемента управління здійснюється за допомогою вказівки `@interact`, після якої ключовим словом `def` оголошується сама функція та її ім’я `name`. Якщо `name` не використовується у подальших розрахунках, то використовують символ «`_`» – нижнє підкреслення. Потім, деякій змінній `a` присвоюємо результат виконання функції, що відповідає потрібному елементу управління. Для ілюстрації зовнішнього вигляду створених елементів управління (таблиця 1) використано функцію `show()`, що показує як сам елемент управління, так і поточне значення змінної `a`, що виводиться у полі графічних побудов.

Таблиця 1

Відомості щодо створення елементів управління в СКМ Sage

«Повзунок 1»

Функція:

`slider(vmin, vmax, step_size, default, label, display_value),`

`vmin` – основний параметр для задання мінімального значення;

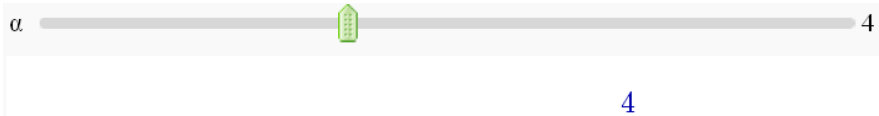
`vmax` – основний параметр для задання максимального значення;

`step_size` – додатковий параметр для задання кроку зміни числових значень;

default – додатковий параметр для задання значення за замовчуванням;
label – додатковий параметр для задання надпису ліворуч від елемента;
display_value – додатковий параметр логічного типу для регулювання виведенням на екран поточного значення.

Приклади:

```
@interact
def name(a = slider(1, 9, 1, default=4, label="α", )):
    show(a)
```



Значеннями «повзунка» можуть бути елементи списку різноманітної природи:

```
@interact
def _(a=slider([1..100], None, None, 3, 'α')):
    show(a)
```

```
@interact
def _(a=slider([1, 'x', 'abc', 2/3], 'x', 'α')):
    show(a)
```

«Повзунок 2»

Функція: range_slider(vmin, vmax, step_size, default, label),
vmin – основний параметр для задання мінімального значення;
vmax – основний параметр для задання максимального значення;
step_size – додатковий параметр для задання кроку зміни числових значень;
default – додатковий параметр для задання значень за замовчуванням у форматі (value_left, value_right);
label – додатковий параметр для задання надпису ліворуч від елемента.

Приклади:

```
@interact
def _(a=range_slider(1, 10, 1, default=(4, 5), label='Діапазон')):
    show(a)
```



«Прапорець»

Функція: checkbox(default, label),

default – основний параметр для задання стану «прапорця» за замовчуванням (приймає значення false або true);

label – додатковий параметр для задання надпису ліворуч від елемента.

Приклади:

```
@interact
```

```
def _ (a=checkbox(False, "Показувати відповідь")) :  
    show(a)
```

Показувати відповідь

False

```
@interact
```

```
def _ (a=checkbox(True, "Показувати відповідь")) :  
    show(a)
```

Показувати відповідь

True

«Меню вибору»

Функція: selector(values, label, default, nrows, ncols, width, buttons),

values – основний параметр для задання значення пунктів меню вибору, що можуть зазначатися переліком елементів – [val1, val2, val3, ...] або діапазоном елементів – [val_start..val_finish];

label – додатковий параметр для задання надпису ліворуч від елемента;

default – додатковий параметр для задання значень за замовчуванням;

nrows – додатковий параметр для задання кількості рядків у поданні пунктів меню вибору (при поданні пунктів меню вибору у вигляді кнопок);

ncols – додатковий параметр для задання кількості стовпчиків у поданні пунктів меню вибору (при поданні пунктів меню вибору у вигляді кнопок);

width – додатковий параметр для задання ширини кнопок (при поданні пунктів меню вибору у вигляді кнопок);

buttons – додатковий параметр логічного типу: при встановленому значенні true меню вибору подається у вигляді кнопок, при встановленому значенні false (за замовчуванням) – у вигляді списку, що розкривається.

Приклади:

```
@interact
```

```
def _ (a=selector([1..5], "Виберіть значення", default=2 ,  
buttons=false)) :  
    show(a)
```

Виберіть значення

2

```
@interact
def _(a=selector(['Приклад 1', 'Приклад 2', 'Приклад 3', 'Приклад 4',
                 'Приклад 5', 'Приклад 6'], label="", default='Приклад 6',
                 nrows=3, ncols=2, width=15)):
    show(a)
```

Приклад 1	Приклад 2
Приклад 3	Приклад 4
Приклад 5	Приклад 6

Приклад 6

«Поле для введення»

Функція: `input_box(default, label, type, width, kwargs)`,
`default` – основний параметр для задання значення, що повертається функцією за замовчуванням;
`label` – додатковий параметр для задання надпису ліворуч від елемента;
`type` – додатковий параметр для визначення типу даних, що вводяться;
`width` – додатковий параметр для задання ширини поля;
`kwargs` – додатковий параметр для підключення одного з існуючих словників.

Приклади:

```
@interact
def _(a=input_box("2+89", 'Введіть значення', width=10)):
    show(a)
```

Введіть значення

91

```
@interact
def _(a=input_box('Sage', label="", type=str)):
    show(a)
```

Sage

Sage

«Комірки для введення»

Функція:
`input_grid(nrows, ncols, default, label, to value=lambda`

x:x,width)
 nrows – основний параметр для задання кількості рядків;
 ncols – основний параметр для задання кількості стовпчиків;
 default – основний параметр для задання початкових значень у комірках;
 label – додатковий параметр для задання надпису ліворуч від елемента;
 to_value=lambda x:x – основний параметр для формування та виведення заданих даних у вигляді списку;
 width – додатковий параметр для задання ширини комірок;

Приклади:

```
@interact
def _(a=input_grid(3,4,default=[1,2,3,4,5,6,7,8,9, 10,11,12], label='Матриця A', to_value=lambda x:x, width=2)):
    show(a)
```

	1	2	3	4
Матриця A	5	6	7	8
	9	10	11	12

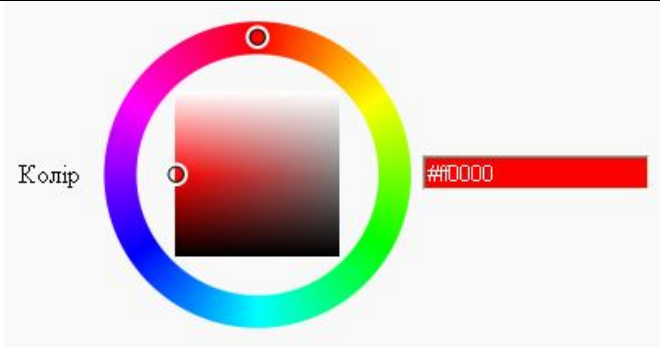
[[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]

«Поле вибору кольору»

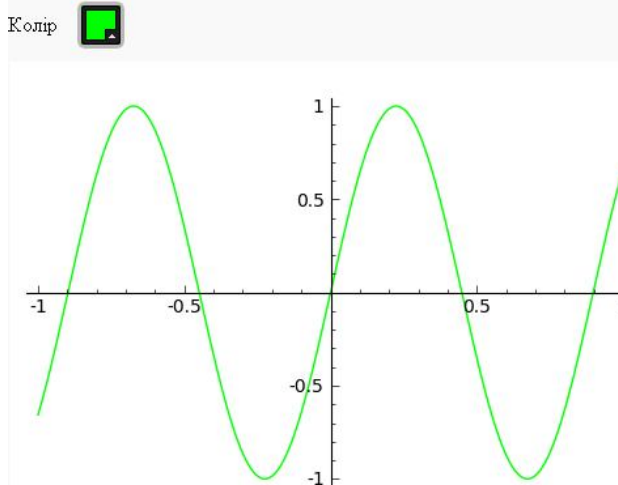
Функція: color_selector(default,label,widget,hide_box)
 default – основний параметр для задання кольору у палітрі (R,G,B);
 label – додатковий параметр для задання надпису ліворуч від елемента;
 widget – основний параметр для задання вигляду діалогового вікна, за замовчуванням присвоюється значення jpicker, також може набувати значень farbtastic або colorpicker.
 hide_box – основний параметр для відображення вікна вводу кольору у шістнадцятиричному форматі, за замовчуванням присвоюється значення False

Приклади:

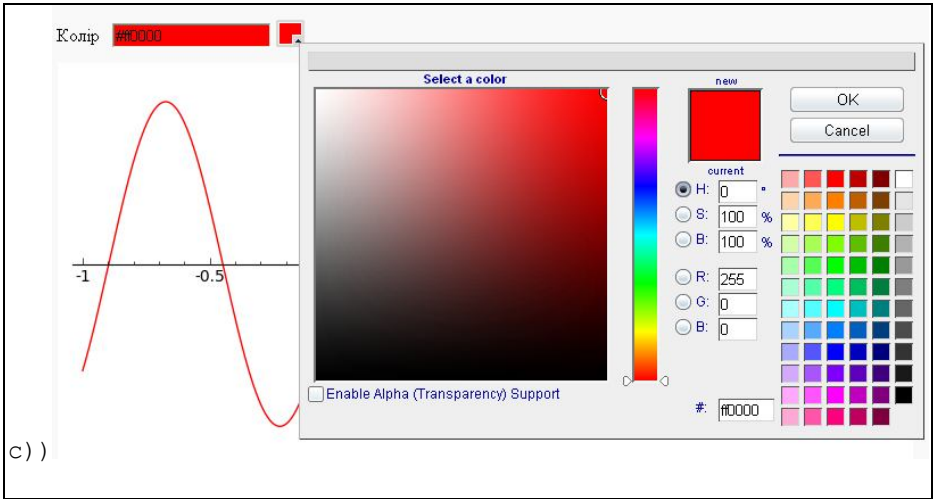
```
@interact
def _(c=color_selector((1,0,0),"Колір",widget='farbtastic',hide_box=False)):
    show(plot(sin(7*x), color = c))
```



```
@interact
def _(c=color_selector((0,1,0),"Колір",widget=
'colorpicker',hide_box=True)):
    show(plot(sin(7*x), color = c))
```



```
@interact
def _(c=color_selector((1,0,0),"Колір",widget=
'jpicker',hide_box=False)):
    show(plot(sin(7*x), color = c))
```



Зазвичай, створення лекційних демонстрацій потребує одночасного виведення на екран декількох елементів управління, раціональне та естетичне розташування яких, досягається за рахунок використання опції `layout` з відповідними ключами: `top`, `bottom`, `left`, `right` (вгору, вниз, ліворуч, праворуч відповідно). Наведемо декілька прикладів.

Приклад 1.

```
@interact(layout={'top':[['a','b']], 'left':[['c']]])
def _(a=input_box("2+2", 'Вираз', width=5),
      b=slider(2, 5, 3/17, 3, 'R'),
      c=checkbox(False, "Відобразити")):
    show(a)
```



Приклад 2.

```
@interact(layout={'right':[['a','b']], 'left':[['c']]])
def _(a=input_box("2+2", 'Вираз', width=5),
      b=slider(2, 5, 3/17, 3, 'R'),
      c=checkbox(True, "Відобразити")):
    show(a)
```

Відображати Вираз R

52/17

Приклад 3.

```
@interact(layout=[[ 'a', 'b'], [ 'c', 'd'], [ 'e' ]])
def _(a=input_box("2+2", 'Вираз'),
      b=selector([1,2,7], default=2),
      c=checkbox(False, "Відображати"),
      d=checkbox(True),
      e=slider([1..10], None, None, 3, 'N')):
    show(a)
```

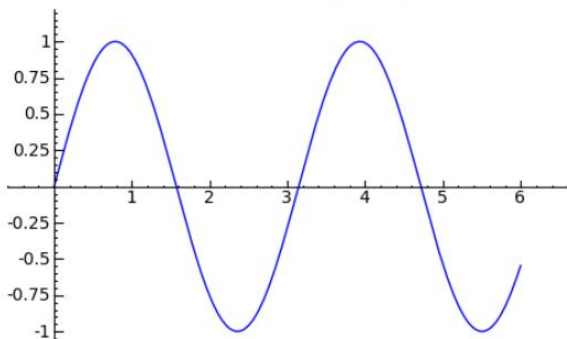
Крім елементів управління та основного програмного коду, невід'ємною частиною лекційних демонстрацій є різноманітні підписи, що надають можливість більш детально ілюструвати та коментувати різні математичні вирази. Для додавання підписів у полі графічних побудов у вигляді таблиць, кольорових графіків, текстів різного формату та надписів – використовують мову HTML, а для подання математичних виразів та формул у природному записі – команди мови LaTeX.

Приклад 4.

```
@interact
def _(a = slider(1, 4, default=2, label="<font size=4 color='blue'><b>a:</b></font>"),
      b = slider(0, 10, default=0, label="<font size=4 color='blue'><b>b:</b></font>")):
    html("<font size=5 color=blue> <center> <b> Півняння виду "+"<font size=3 color=red> <b> "+"$y=\sin(a\cdot x+b)$"+" </font></b>")
    show(plot(sin(a*x+b), (x,0,6)), figsize=5)
```


a: 1.9979959919839505
 b: 0.0

Рівняння виду $y = \sin(a \cdot x + b)$



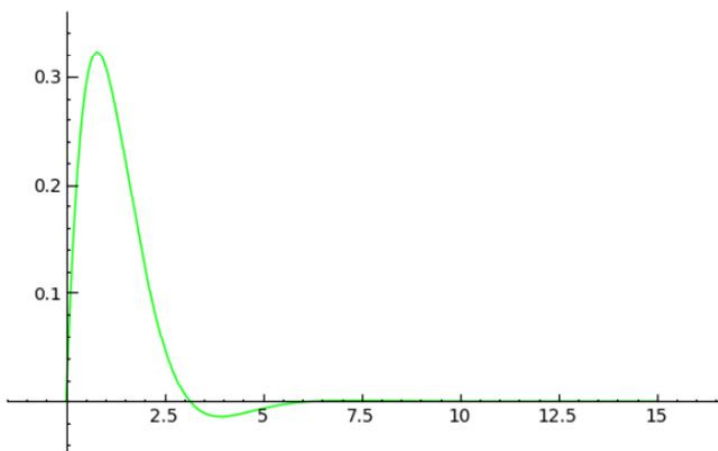
Так, у прикладі 4, під час оголошення функцій, що створюють елементи управління типу «Повзунок» параметру `label`, присвоюється значення записане за допомогою команд HTML, що передбачають збільшення розміру та кольору тексту на екрані. Для виведення підпису «Рівняння виду ...» у звичному вигляді необхідно записати службове слово `html`, після якого в дужках за допомогою відповідних тегів вказати колір, тип шрифту, розмір та розташування тексту, що виводиться. Виведення функції $y = \sin(ax + b)$ здійснюється з використання команд мови LaTeX, на що вказують одинарні лапки та знаки « $\$$ ». Кожна команда розпочинається символом « \backslash » (backslash – обернений слеш), після якого зазначається власне ім'я команди: команда « $\backslash\sin$ » відображає функцію `sin`, команда « $\backslash\cdot$ » – операцію множення.

Приклад 5.

```
@interact
def _(f=input_box(label="Введіть функцію:", default=sin(x)*e^(-x))):
    var('x')
    show(plot(f, 0, 15, rgbcolor='green', thickness=1))
    html("<b><font color=blue size=4>Зображено графік функції:  

    "+ '$f(x)=%s$' % latex(f) + " "+ "</font></b>")
```

Введіть функцію:



Зображено графік функції: $f(x) = e^{-x} \sin(x)$

На відміну від прикладу 4, приклад 5 передбачає разом із зміною параметру у полі «Введіть функцію», зміну функції, що виводиться разом з підписом «Зображено графік функції:». Це досягається за рахунок використання наступної комбінації: `+'$f(x)=%s$'\%latex(f)+`, де $f(x)$ – назва функції (постійне значення), `%s` – оператор для виведення строкової змінної, `%latex` – оператор для виведення математичних виразів у природній формі.

Отже, для створення лекційних демонстрацій у СКМ Sage необхідно:

- відповідно до моделі, що розробляється створити необхідні елементи управління;
- описати програмний код моделі, що створюється за допомогою комбінації необхідних команд (побудова графіків, фігур, обчислення виразів тощо);
- додати підписи різного формату у вигляді тексту або математичних виразів за допомогою команд мови HTML і LaTeX.

Література

1. Попель М. В. Засоби наочності ММС «Вища математика: мобільний курс» / М. В. Попель, К. І. Словак // Інноваційні інформаційно-комунікаційні технології навчання математики, фізики, інформатики у середніх та вищих навчальних закладах : зб. наук. праць за матеріалами Всеукраїнської науково-методичної конференції молодих науковців, 17–

18 лютого 2011 р. – Кривий Ріг : Криворізький державний педагогічний ун-т, 2011. – С. 54–58.

2. Словак К. І. Особливості застосування ММС Sage під час вивчення курсу вищої математики / К. І. Словак, М. В. Попель // Теорія та методика навчання математики, фізики, інформатики : збірник наукових праць. Випуск VIII : в 3-х томах. – Кривий Ріг : Видавничий відділ НМетАУ, 2010. – Т. 1 : Теорія та методика навчання математики. – С. 125–130.

3. Шокалюк С. В. Методичні засади комп'ютеризації самостійної роботи старшокласників у процесі вивчення програмного забезпечення математичного призначення : дис. ... канд. пед. наук : 13.00.02 – теорія та методика навчання (інформатики) / Шокалюк Світлана Вікторівна ; Національний педагогічний ун-т ім. М. П. Драгоманова. – К., 2009. – 261 с.