

Мій перший документ

МАН

2022-11-14

Оформлення шаблону книги

Мосійчук В.С.

Київський політехнічний інститут імені Ігоря Сікорського, м. Київ, Україна

E-mail: *mvs@ros.kpi.ua*

У роботі представлені вимоги щодо оформлення статей для подання у збірник наукових праць “Вісник Національного технічного університету України «Київський політехнічний інститут». Серія Радіотехніка. Радіоапаратобудування”. Показано, що дотримання встановлених правил дозволить покращити вашу статтю .

Ключові слова: правила оформлення, радіотехніка, радіоапаратобудування, L^AT_EX

1 Тест підтримки української мови

Obviously the statements title, date and author are not within the the document environment, so they will not directly show up in our document. The area before our main document is called preamble. In this specific example we use it to set up the values for the maketitle command for later use in our document. This command will automagically create a titlepage for us. The newpage command speaks for itself.

If we now compile again, we will see a nicely formatted title page, but we can spot a page number at the bottom of our title page. What if we decide, that actually, we don't want to have that page number showing up there. We can remove it, by telling LaTeX to hide the page number for our first page. This can be done by adding the pagenumbergobble command and then changing it back to on the next page numbers like so:

Формула кола: $y^2 + x^2 = 1$

$$y = \sqrt{x + 1}$$

$$y = \sqrt{x^2 + 1}$$

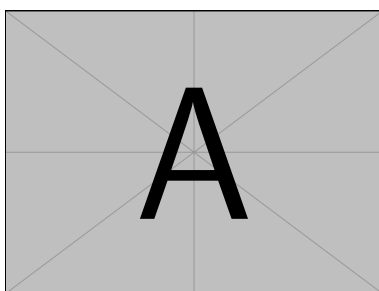
$$H(A) = - \sum_{i=1}^n p_i \log p(p_i) \quad (1)$$

$$H(A) = - \sum_{i=1}^n p_i \quad (2)$$

That's it. You've successfully created your first LaTeX document. The following lessons will cover how to structure your document and we will then proceed to make use of many features of LaTeX.

1.1 Begin subsection

Obviously the statements title, date and author are not within the the document environment, so they will not directly show up in our document. The area before our main document is called preamble. In this specific example we use it to set up the values for the maketitle command for later use in our document. This command will automagically create a titlepage for us. The newpage command speaks for itself.



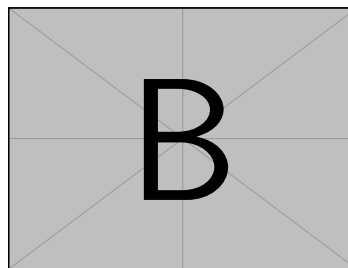
- Point 1 under the first description

Мал. 1: Підпис до рисунку.

If we now compile again, we will see a nicely formatted title page, but we can spot a page number at the bottom of our title page. What if we decide, that actually, we don't want to have that page number showing up there. We can remove it, by telling LaTeX to hide the page number for our first page. This can be done by adding the `pagenumberinggobble` command and then changing it back to `on` on the next page numbers like so:

If we now compile again, we will see a nicely formatted title page, but we can spot a page number at the bottom of our title page. What if we decide, that actually, we don't want to have that page number showing up there. We can remove it, by telling LaTeX to hide the page number for our first page. This can be done by adding the `pagenumberinggobble` command and then changing it back to `on` on the next page numbers like so:

If we now compile again, we will see a nicely formatted title page, but we can spot a page number at the bottom of our title page. What if we decide, that actually, we don't want to have that page number showing up there. We can remove it, by telling LaTeX to hide the page number for our first page. This can be done by adding the `pagenumberinggobble` command and then changing it back to `on` on the next page numbers like so:



Мал. 2: Підпис до рисунку.

2 Таблиці

Obviously the statements title, date and author are not within the the document environment, so they will not directly show up in our document. The area before our main document is called preamble. In this specific example we use it to set up the values for the maketitle command for later use in our document. This command will automagically create a titlepage for us. The newpage command speaks for itself.

Obviously the statements title, date and author are not within the the document environment, so they will not directly show up in our document. The area before our main document is called preamble. In this specific example we use it to set up the values for the maketitle command for later use in our document. This command will automagically create a titlepage for us. The newpage command speaks for itself.

Obviously the statements title, date and author are not within the the document environment, so they will not directly show up in our document. The area before our main document is called preamble. In this specific example we use it to set up the values for the maketitle command for later use in our document. This command will automagically create a titlepage for us. The newpage command speaks for itself.

Obviously the statements title, date and author are not within the the document environment, so they will not directly show up in our document. The area before our main document is called preamble. In this specific example we use it to set up the values for the maketitle command for later use in our document. This command will automagically create a titlepage for us. The newpage command speaks for itself.

dummy text dummy text	cell2	cell3
cell4	cell5	cell6
cell7	cell8	dummy text

Col1	Col2	Col2	Col3
1	6	87837	787
2	7	78	5415
3	545	778	7507
4	545	18744	7560
5	88	788	6344

cell1 text text text	dummy dummy dummy	cell2	cell3
cell1 text text text	dummy dummy dummy	cell5	cell6
cell7		cell8	cell9

cell1 dummy text dummy text dummy text cell1 dummy text dummy text dummy text	item 12	item 13
item 21	item 22	item 23

2.1 Об'єднання комірок

Country List			
Country Name or Area Name	ISO ALPHA 2 Code	ISO ALPHA 3 Code	ISO numeric Code
Afghanistan	AF	AFG	004
Aland Islands	AX	ALA	248
Albania	AL	ALB	008
Algeria	DZ	DZA	012
American Samoa	AS	ASM	016
Andorra	AD	AND	020
Angola	AO	AGO	024

col1	col2	col3
Multiple row	cell2 cell5 cell8	cell3 cell6 cell9

3 Довгі таблиці

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat

ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

[illegible]

[illegible]

Табл. 1: Your caption here

Table 1 shows my first longtable.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

4 Висновки

A document has a preamble and document part The document environment must be defined

- Commands beginning with a backslash environments have a begin and end tag
- Useful settings for pagenumbering:
 - gobble – no numbers
 - arabic – arabic numbers
 - roman – roman numbers
 - roman – roman numbers2
 - roman – roman numbers3
 - roman – roman numbers4
 - roman – roman numbers5

5 Копище

Лозко Валерій Іванович

Копищенський ліцей, Олевська ТГ., Коростенський р-н., Житомирська обл.

E-mail: *v.i.lozko2gmail.com*

Засноване	1459 р.
Населення	983
Площа	2,83 км ²

5.1 Історія села Копище

Копище — село розташоване на березі річки Уборті (притока Прип'яті), за 55 км від районного центру та залізничної станції Олевськ.

Після **Андрусівського перемир'я** 1667 року село залишилося у складі Польщі. Боротьбу проти своїх поневолювачів селяни не припиняли. Вони активно боролися під час визвольного повстанського руху проти польсько-шляхетських загарбників за возз'єднання Правобережної України з Росією під проводом С. Палія у кінці XVII — на початку XVIII століття.

Страшних бідувань зазнали селяни Копища в роки першої світової війни. Відчувалася нестача робочих рук, тягла, скоротилися посівні площі, голод і хвороби косили людей.

Звістку про Лютневу буржуазно-демократичну революцію **1917** року в село принесли солдати, які поверталися з фронту. Всі чекали змін у житті, вимагали розподілу земель, але Тимчасовий уряд не поспішав поліпшувати становище трудового народу.

Активними борцями за нове життя були **копищанські жінки**. Так, у постійно діючих комісіях при сільраді їх працювало 4, у комнезамі — 3, у касі взаємодопомоги — 3, у шкільній раді — 3, у правлінні кредитного товариства — 3. На з'їзд жінок Олевського району 1925 року від Конища поїхало 16 жінок.

1927 року в селі Копищі було організовано перше колективне господарство — ТСОЗ, що охопив спершу 12 господарств, а на початку 1928 року — вже 35. Товариство мало всього лише 8 коней, 4 плуги, 4 борони, 2 культиватори та сівалку. В перші місяці його існування держава надала кредити в розмірі 7690 крб. на будівництво господарчих приміщень, придбання робочої худоби, на закладення саду.

1936 року в селі діяли медична амбулаторія, пологовий будинок на 8 ліжок, де, крім лікаря, працювали дві акушерки та дві санітарки. Під час жнив відкривалися дитячі ясла на 120 місць.

1936 року збудовано нове приміщення семирічної школи, де 8 учителів навчали 300 учнів. Добра слава йшла про роботу Копищанського клубу. Тут читалися лекції й доповіді. У залі на 200 місць ставилися концерти силами гуртківців, двічі на місяць демонструвалися кінокартини. Сільська бібліотека мала книжковий фонд 6 тис. примірників. Товарами народного споживання трудящих забезпечували два магазини.



Мал. 3: Пам'ятник односельцям, загиблим в результаті трагедії 13 липня 1943 р.

13 липня 1941 року в село (вдерлися фашисти. Настали чорні дні неволі. Гітлерівці грабували населення, забирали хліб, худобу. 70 мирних жителів вивезли на каторжні роботи до Німеччини.

13 липня 1943 року в с.Копище сталася одна з найжорстокіших трагедій України: нацистські загарбники влаштували



Мал. 4: Відзначення 75-ї річниці Копищенської трагедії

каральну операцію “Фрау Хельга” (пані Ольга), метою якої було повне знищення жителів села. 4 За один день спалили живими 2887 мирних жителів, з них 1347 дітей віком до 12 років. (Мал. 3) Врятувались лише четверо дорослих та 46 дітей.

Незважаючи на величезні труднощі, копищанці всі сили віддавали на відродження рідного села. Стало легше працювати, коли після перемоги над гітлерівською Німеччиною до села почали повертатися демобілізовані чоловіки.

Поступово піднімалося з руїн Копище. Вже через рік після перемоги тут виросла ціла вулиця. Один з будинків пристосували під лікарняний пункт, інший під початкову школу.

Основні сільськогосподарські культури у колгоспі — льон, картопля, зернові. Важливу роль у його економіці відіграє тваринництво м'ясо-молочного напрямку. На 1 січня 1967 року колгосп мав 685 голів великої рогатої худоби, в т. ч. 275 корів та 199 голів овець. Для їх розміщення споруджено під шифером 5 добротних тваринницьких приміщень. На той час господарство було повністю електрифіковано, мало 9 тракторів, 6 комбайнів, 8 автомашин. Діяла пилорама.

На тваринницьких фермах обладнано автоматичне напування.

Копищанці справжні господарі своєї долі. 35 чоловік обрано депутатами сільської Ради, серед яких 15 жінок, 11 чоловік — членами правління колгоспу. В постійно діючих комісіях працює близько 100 чоловік. Бюджет Ради на 1973 рік становить 39 964 крб., переважна більшість цієї суми буде використана на поліпшення охорони здоров'я, на піднесення культурно-освітньої роботи.

В центрі Копища знаходиться Братська могила жертвам фатішизму (1961 р.), в якій поховано 2887 жителів села, у тому числі 1347 дітей (Мал. 3).

Поруч знаходиться музей, який розповідає людям про історію рідного села і про страшну трагедію, яку не можна забути. Директор музею — Мартиновець Марія Василівна.

У селі наразі є загальноосвітня школа I-III ступенів - сайт Копищенського ліцею. Дітей навчається 200. Також функціонує заклад дошкільної освіти з короткотривалим перебуванням дітей без харчування, який відвідує 22 дітей.

6 Вивчаємо фізику за допомогою Scratch

Мізюк Олександр Миколайович

Механіка. Вільне падіння і криволінійний рух під дією незмінної сили тяжіння

У цьому розділі представлений зразок розв'язування задачі на рух тіла кинутого під кутом до горизонту. Для моделювання процесу руху використовуйте застосунок, створений у середовищі програмування Scratch.

Ключові слова: механіка, криволінійний рух, Scratch, L^AT_EX

6.1 Рух тіла, кинутого під кутом до горизонту

Прочитавши про рекорди швидкості польоту спортивних снарядів, учениця Оленка вирішила з'ясувати, якої швидкості вона надає футбольному м'ячу. Для цього дівчинка вдарила по м'ячу, спрямувавши його під кутом 45° до горизонту (див. мал. 5).



Мал. 5: За напрямком і дальністю польоту м'яча ви можете визначити, якої швидкості ви надали м'ячу під час удару або кидка.

М'яч упав на землю на відстані **40 м** від учениці. Виконавши розрахунки, дівчинка вирішила, що вона надала м'ячу швидкості **20 м/с**, а м'яч піднявся на висоту **8 м**. Чи не помилилася учениця?

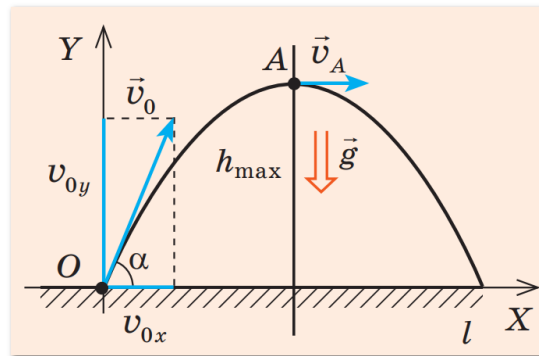
6.2 Розв'язуємо задачу¹

Футболістка вдарила по м'ячу, надавши йому швидкості v_0 , напрямленої під кутом α до горизонту. Визначте дальність польоту та найбільшу висоту підйому м'яча.

¹Ознайомтеся з розв'язанням задачі. Скориставшись отриманими формулами, оцініть розрахунки дівчинки, а після уроків проведіть подібний експеримент та оцініть швидкість, якої ви надаєте м'ячу.

Дано	Розв'язання
v_0	
α	
g	
$L-?$	
$h_{max}-?$	

Виконаємо пояснювальний рисунок (див. мал. 6): початок координат пов'яжемо з точкою на поверхні Землі, де м'яч відірвався від бутси футболістки; вісь **OY** спрямуємо вертикально вгору; вісь **OX** — горизонтально.



Мал. 6: Пояснювальний рисунок.

В обраній системі відліку:
рух уздовж осі **OX** — рівномірний:

$$v_x = v_{0x}, x = x_0 + v_{0x}t, \quad (3)$$

де

$$x_0 = 0, v_{0x} = v_0 \cos \alpha$$

рух уздовж осі **OY** — рівноприскорений:

$$v_y = v_{0y} + g_y t, y = y_0 + v_{0y}t + \frac{g_y t^2}{2}, \quad (4)$$

де

$$y_0 = 0, v_{0y} = v_0 \sin \alpha, g_y = -g,$$

тому рівняння (3) і (4) набувають вигляду:

$$v_x = v_0 \cos \alpha, x = v_0 \cos \alpha \cdot t \quad (5)$$

і

$$v_y = v_0 \sin \alpha - gt, y = v_0 \sin \alpha \cdot t - \frac{gt^2}{2} \quad (6)$$

відповідно. Час t_1 руху м'яча до верхньої точки траєкторії (точки **A**) знайдемо з умови: $v_y(t_1) = 0$:

$$v_0 \sin \alpha - gt_1 = 0 \Rightarrow t_1 = \frac{v_0 \sin \alpha}{g}. \quad (7)$$

Координата y м'яча в точці **A** — це максимальна висота підйому м'яча:

$$h_{max} = y_A = v_0 \sin \alpha \cdot t_1 - \frac{gt_1^2}{2}. \quad (8)$$

Після підстановки t_1 отримуємо формули для визначення максимальної висоти підйому та загального часу руху м'яча:

$$h_{max} = \frac{v_0^2 \sin^2 \alpha}{2g}; \quad (9)$$

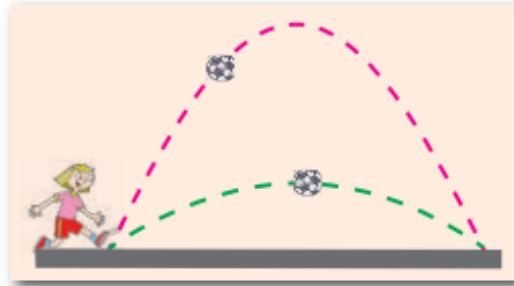
$$t = 2t_1 = \frac{2v_0 \sin \alpha}{g}. \quad (10)$$

Дальність **L** польоту дорівнює координаті **x** тіла наприкінці руху (**x** = **L**):

$$x = v_0 \cos \alpha \cdot t = v_0 \cos \alpha \cdot \frac{2v_0 \sin \alpha}{g}. \quad (11)$$

Отже, дальність польоту:

$$2 \cos \alpha \cdot \sin \alpha \Rightarrow L = \frac{v_0^2 \sin 2\alpha}{g}. \quad (12)$$

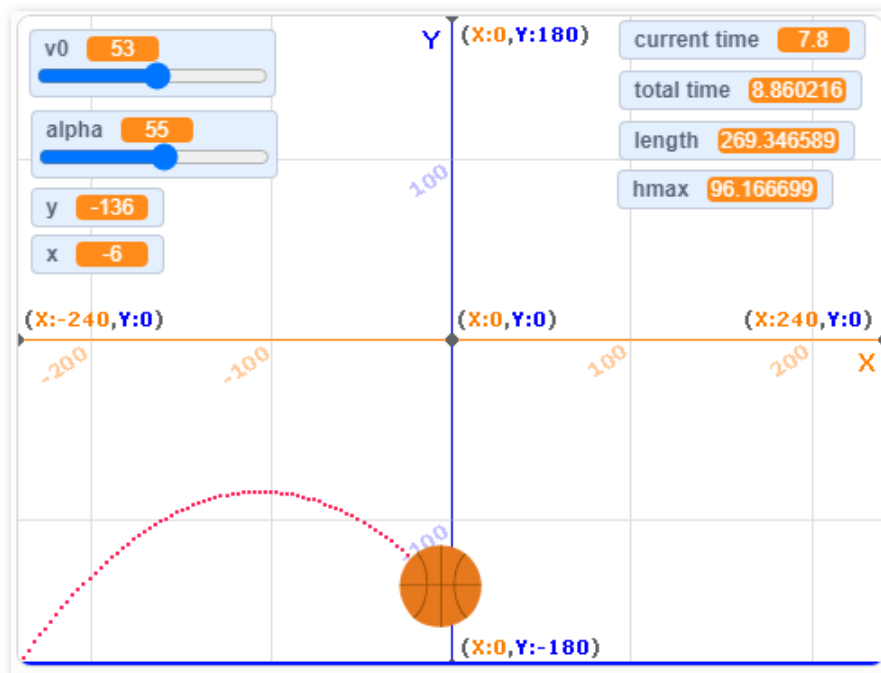


Мал. 7: Однакова дальність польоту при різних траєкторіях.

Зверніть увагу! З останньої формули випливає:

- якщо кинути тіло під кутом α , а потім під кутом $90^\circ - \alpha$, то дальність польоту не зміниться, тобто тіло потрапить у ту саму точку, рухаючись різними траєкторіями (див. мал. 7)
- максимальної дальності польоту тіло сягає, якщо $\alpha = 45^\circ$ ($\sin 2\alpha = 1$).

6.3 Моделювання руху у Scratch



Мал. 8: Знімок вікна проєкту.

Щоб розглянути код проєкту, перейдіть за покликанням.

6.4 Підбиваємо підсумки

Траєкторія руху тіла, кинутого під кутом до горизонту, - параболічна. Такі рухи розглядають як результат додавання двох простих рухів: горизонтального - рівномірного уздовж осі **OX** і вертикального - рівноприскореного (з прискоренням **g**) уздовж осі **OY**.

Рівняння залежностей проекції швидкості та координати від часу у цьому разі мають вигляд:

$$v_x = v_{0x},$$

$$x = x_0 + v_{0x}t,$$

$$v_y = v_{0y} + g_y t,$$

$$y = y_0 + v_{0y}t + \frac{g_y t^2}{2}.$$

6.5 Контрольні запитання

- Який вигляд має рівняння руху, якщо тіло кинуто під кутом до горизонту?
- Якою є траєкторія руху тіла, кинутого під кутом до горизонту? Наведіть приклади.
- Як визначити модуль і напрямок швидкості руху тіла в будь-якій точці траєкторії?

6.6 Вправа для самостійного розв’язування

Опором повітря нехтуйте. Вважайте, що $g = 10^2$ м/с.

Струмінь води, напрямлений під кутом 60° до горизонту, сягнув висоти 15 м.

1. Використайте розроблений Scratch-проект для дослідження руху струменя води.
2. Знайдіть:
 - (а) швидкість витікання води;
 - (б) час польоту частинок струменя;
 - (в) дальність польоту частинок струменя.
3. Якою буде дальність струменя, якщо спрямувати його під кутом 30° до горизонту?
4. Чому струмінь води розширюється?

7 Розкладання рядків (Регулярні вирази)

7.1 Мета:

ознайомитися з поняттям “регулярних виразів”, синтаксисом їхнього запису та можливостями їх реалізації засобами Java; навчитись практично використовувати “регулярні вирази” для пошуку і перевірки рядків на відповідність певному шаблону.

7.2 Теоретичний матеріал

1. Що таке “Регулярні вирази”?
2. Засади синтаксису.
3. Символьні класи (набір символів).
4. Діапазони.
5. Квантифікатори (вказівка кількості повторень).
6. Групування.

7.3 Проект «Телефонний довідник»

Програма “Телефонний довідник” імітує роботу телефонної книги та містить інформацію про абонента, номер телефону та його електронну адресу.

Завдання:

- Створення графічного інтерфейсу для введення та виведення інформації (Рис. 9).
- Створення класу Phone для визначення абонентів в телефонній книзі.
- Накладання умов на правильність занесення даних до книги.

Хід роботи:

1. Використовуючи Scene Builder, розробляємо інтерфейс вікна програми (Рис. 9). Ієрархія елементів вікна представлена на Рис. 10.
2. Для створення розмітки використовуємо контейнери VBox і HBox. Для представлення даних у вигляді таблиці будемо використовувати TableView.
3. Для елементів головного вікна програми, до яких необхідно отримати доступ з Java-коду встановлюємо fx:id:
 - phoneList – таблиця для виведення інформації
 - nameCol – поле для виведення імені
 - phoneCol – поле для виведення номеру телефону
 - addressCol – поле для виведення електронної адреси
 - nameInput – поле для введення імені
 - phoneInput – поле для введення номеру телефону
 - addressInput – поле для введення електронної адреси

- addPhone – On Action для додавання абонента
- filterInput – поле для введення ключових значень для пошуку
- filterPhones – On Action для пошуку по ключовим значенням

4. Створюємо клас Phone, який визначає модель даних і надає методи та поля для подальшої роботи з таблицею.

```
public class Phone {
    private final SimpleStringProperty name;
    private final SimpleStringProperty phone;
    private final SimpleStringProperty address;

    Phone(String name, String phone, String address) {
        this.name = new SimpleStringProperty(name);
        this.phone = new SimpleStringProperty(phone);
        this.address = new SimpleStringProperty(address);
    }

    public String getName() {
        return name.get();
    }

    public void setName(String name) {
        this.name.set(name);
    }

    public String getPhone() {
        return phone.get();
    }

    public void setPhone(String phone) {
        this.phone.set(phone);
    }

    public String getAddress() {
        return address.get();
    }

    public void setAddress(String address) {
        this.address.set(address);
    }
}
```

Об'єкти name, phone, address створені для ввімкнення посилання на певний елемент даних таблиці. Крім того, для кожного елемента даних передбачені get та set методи .

5. В класі PhoneBookController створюємо масив ObservableList і заносимо дані про абонентів. Ці особи будуть зразу відображатись у нашій телефонній книзі. Інших абонентів можна буде додати пізніше за допомогою відповідних полів та кнопок.

```
public class PhoneBookController {
    private final ObservableList<Phone> data = FXCollections.observableArra
```

```
new Phone("Іванов І.І.", "094794721", "ivan.ivanov@ukr.net"),
new Phone("Петрова М.Н.", "0954847216", "isabella@example.com"),
new Phone("Сидоров Ю.А.", "0673259833", "ethan.man@gmail.com"),
new Phone("Васильків П.Р.", "0504877732", "freeman@yahoo.com"),
new Phone("Герасименко Р.В.", "0963334532", "michael@mail.com")
);
```

6. Наступним кроком є зв'язування даних зі стовпцями таблиці:

```
@FXML public void initialize() {
    nameCol.setCellValueFactory(
        new PropertyValueFactory<Phone,String>("name")
    );
    nameCol.setCellFactory(TextFieldTableCell.forTableColumn());

    phoneCol.setCellValueFactory(
        new PropertyValueFactory<Phone,String>("phone")
    );
    phoneCol.setCellFactory(TextFieldTableCell.forTableColumn());

    addressCol.setCellValueFactory(
        new PropertyValueFactory<Phone,String>("address")
    );
    addressCol.setCellFactory(TextFieldTableCell.forTableColumn());
    nameCol.setOnEditCommit(
        new EventHandler<CellEditEvent<Phone, String>>() {
            @Override
            public void handle(CellEditEvent<Phone, String> t) {
                Pattern ptNameColl = Pattern.compile("[A-ZА-ЯІ]{1}[a-za-яі]+\s");
                Matcher n = ptNameColl.matcher(t.getNewValue());
                boolean fpn = n.matches();
                if (fpn) {
                    ((Phone) t.getTableView().getItems().get(
                        t.getTablePosition().getRow()
                    )).setName(t.getNewValue());
                }
                phoneList.refresh();
            }
        }
    );
    phoneCol.setOnEditCommit(
        new EventHandler<CellEditEvent<Phone, String>>() {
            @Override
            public void handle(CellEditEvent<Phone, String> t) {
                Pattern ptPhoneColl = Pattern.compile("[0-9]+");
```

```

        Matcher p = ptPhoneColl.matcher(t.getNewValue());
        boolean fpp = p.matches();
        if (fpp) {
            ((Phone) t.getTableView().getItems().get(
                t.getTablePosition().getRow()
            )).setPhone(t.getNewValue());
        }
        phoneList.refresh();
    }
}

);
addressCol.setOnEditCommit(
    new EventHandler<CellEditEvent<Phone, String>>() {
        @Override
        public void handle(CellEditEvent<Phone, String> t) {
            Pattern ptAddressColl = Pattern.compile("[a-z0-9]{1,}@{1}([a-z0-9]{1,})");
            Matcher a = ptAddressColl.matcher(t.getNewValue());
            boolean fpa = a.matches();
            if (fpa) {
                ((Phone) t.getTableView().getItems().get(
                    t.getTablePosition().getRow()
                )).setAddress(t.getNewValue());
            }
            phoneList.refresh();
        }
    }
);

phoneList.setItems(filteredData);
}

```

7. Наступна частина коду дозволяє перевіряти на валідність введення даних до телефонної книги, яку ми записуємо в класі Phone:

```

public boolean isValidName() {
    Pattern ptN = Pattern.compile("[A-ZА-ЯІ]{1}[a-za-яі]+\\s[A-ZА-ЯІ]{1}\\s[A-ZА-ЯІ]{1}");
    Matcher n = ptN.matcher(getName());
    boolean fn = n.matches();
    return fn;
}

public boolean isValidPhone() {
    Pattern ptP = Pattern.compile("[0-9]+");
    Matcher p = ptP.matcher(getPhone());
    boolean fp = p.matches();
}

```

```

        return fp;
    }
    public boolean isValidAddress() {
        Pattern ptA = Pattern.compile("[a-z0-9]{1,}@{1}([a-z0-9]{1,}\\.|\\.)+[a-z]{1,3}");
        Matcher a = ptA.matcher(getAddress());
        boolean fa = a.matches();
        return fa;
    }

```

Можна побачити, що для перевірки правильності введення даних ми використовуємо регулярні вирази (пишемо символічні класи, вказуємо діапазони та кількість повторень) для імені, номеру телефону та електронної адреси.

8. Для додавання нових абонентів в PhoneBookController створюємо обробник події для кнопки addPhone:

```

@FXML public void addPhone() {
    Phone p = new Phone(nameInput.getText(), phoneInput.getText(), addressInput.getText());
    if ((p.isValidName()) && (p.isValidPhone()) && (p.isValidAddress())) {
        data.add(p);
        nameInput.clear();
        phoneInput.clear();
        addressInput.clear();
    }
    if (p.isValidName() == false) {
        nameInput.setStyle("-fx-text-fill: red");
    } else {
        nameInput.setStyle("-fx-text-fill: black");
    }
    if (p.isValidPhone() == false) {
        phoneInput.setStyle("-fx-text-fill: red");
    } else {
        phoneInput.setStyle("-fx-text-fill: black");
    }
    if (p.isValidAddress() == false) {
        addressInput.setStyle("-fx-text-fill: red");
    } else {
        addressInput.setStyle("-fx-text-fill: black");
    }
}

```

Тут передбачено забарвлення тексту в червоний колір, якщо дані, які вводяться, неправильні або не відповідають шаблону введення (прикладом може бути букви в номері телефону чи неправильний e-mail).

9. Для пошуку даних в класі PhoneBookController створюємо обробник події для кнопки filterPhones:

```

@FXML public void filterPhones() {
    filteredData.setPredicate(e -> e.getName().contains(filterInput.getText()));
}
private final FilteredList<Phone> filteredData = new FilteredList<>(data, p ->
}

```

10. Для редагування елементів в таблиці використовуємо метод setOnEditCommit, який зазначений в п.6. При редагуванні ми також перевіряємо дані на правильність введення, використовуючи регулярні вирази.

```

nameCol.setOnEditCommit(
new EventHandler<CellEditEvent<Phone, String>>() {
    @Override
    public void handle(CellEditEvent<Phone, String> t) {
        Pattern ptNameColl = Pattern.compile("[A-ZА-ЯІ]{1}[a-za-яі]+\\s[A-ZА-ЯІ]{1}");
        Matcher n = ptNameColl.matcher(t.getNewValue());
        boolean fpn = n.matches();
        if (fpn) {
            ((Phone) t.getTableView().getItems().get(
                t.getTablePosition().getRow()
            )).setName(t.getNewValue());
        }
        phoneList.refresh();
    }
});

phoneCol.setOnEditCommit(
new EventHandler<CellEditEvent<Phone, String>>() {
    @Override
    public void handle(CellEditEvent<Phone, String> t) {
        Pattern ptPhoneColl = Pattern.compile("[0-9]+");
        Matcher p = ptPhoneColl.matcher(t.getNewValue());
        boolean fpp = p.matches();
        if (fpp) {
            ((Phone) t.getTableView().getItems().get(
                t.getTablePosition().getRow()
            )).setPhone(t.getNewValue());
        }
        phoneList.refresh();
    }
});

```

```

    }

    );

    addressCol.setOnEditCommit(
    new EventHandler<CellEditEvent<Phone, String>>() {
        @Override
        public void handle(CellEditEvent<Phone, String> t) {
            Pattern ptAddressColl = Pattern.compile("[a-z0-9]{1,}@{1}([a-z0-9]{1,})");
            Matcher a = ptAddressColl.matcher(t.getNewValue());
            boolean fpa = a.matches();
            if (fpa) {
                ((Phone) t.getTableView().getItems().get(
                    t.getTablePosition().getRow()
                )).setAddress(t.getNewValue());
            }
            phoneList.refresh();
        }
    }
    );

```

11. Клас Main:

```

public class Main extends Application {
    @Override
    public void start(Stage primaryStage) {
        try {
            VBox root = (VBox)FXMLLoader.load(getClass().getResource("Phone.fxml"));
            Scene scene = new Scene(root,400,400);
            scene.getStylesheets().add(getClass().getResource("application.css").toExternalForm());
            primaryStage.setTitle("Телефонний довідник");
            primaryStage.setScene(scene);
            primaryStage.show();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public static void main(String[] args) {
        launch(args);
    }
}

```

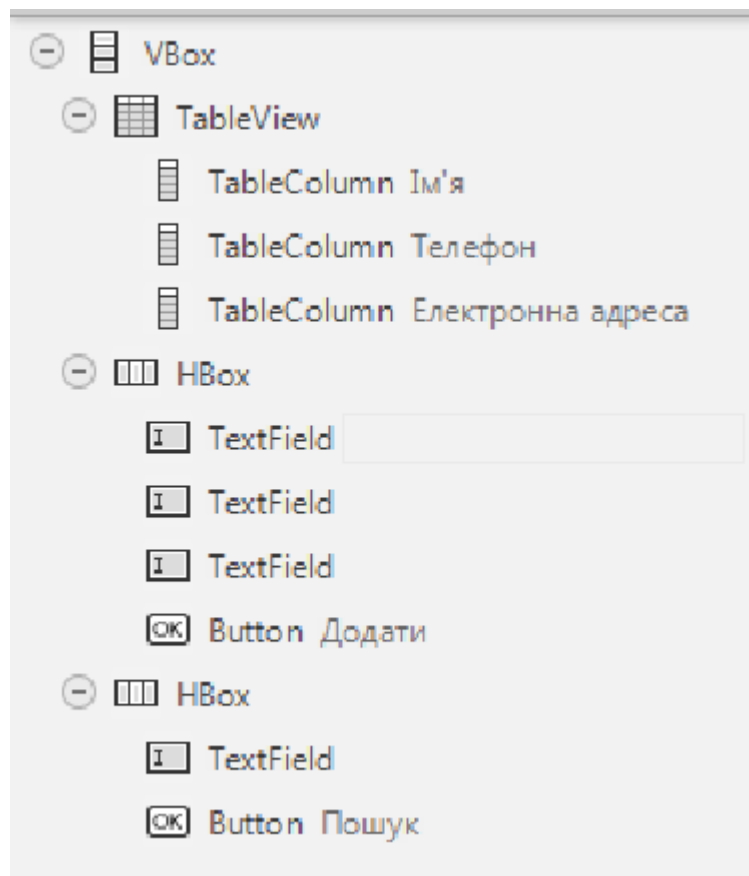
7.4 Задачі для самостійного виконання

В даному проекті створіть можливість передбачення введення телефонного номера у форматі (xxx)-xxx-xx-xx.

Ім'я	Телефон	Електронна адреса
Іванов І.І.	094794721	ivan.ivanov@ukr.net
Петрова М.Н.	0954847216	isabella@example.com
Сидоров Ю.А.	0673259833	ethan.man@gmail.com
Васильків П.Р.	0504877732	freeman@yahoo.com
Герасименко Р.В.	0963334532	michael@mail.com

Ім'я	Телефон	Адреса	Додати
------	---------	--------	--------

Мал. 9



Мал. 10