

Міністерство освіти і науки України
Національний університет «Запорізька Політехніка»

Кафедра програмних засобів

ЗВІТ

з розрахунково-графічного завдання № 1
з дисципліни «Soft skills, групова динаміка та комунікації» на тему:
«СТВОРЕННЯ ПРЕЗЕНТАЦІЙ»

Виконав:

студент групи КНТ-122

О. А. Онищенко

Прийняли:

доцент:

В. М. Льовкін

2023

1 СТВОРЕННЯ ПРЕЗЕНТАЦІЙ	3
Мета роботи	3
Завдання до роботи	3
Короткі теоретичні відомості	4
Текст доповіді.....	5
Презентація	8
ВИСНОВКИ	14
Контрольні запитання.....	15
1.5.1 Поясніть призначення пунктів головного меню програми для створення презентацій.	15
1.5.2 Як змінити шаблон макета слайдів?	15
1.5.3 Як змінити метод розмітки для слайда?	16
1.5.4 Як збільшити/зменшити шрифт тексту? Які типи вирівнювання тексту доступні?	17
1.5.5 Які відмінності між PowerPoint 2003 і 2010?	18

1 СТВОРЕННЯ ПРЕЗЕНТАЦІЙ

Мета роботи

1.1.1 Вивчити основні можливості сучасних програм для підготовки презентацій.

1.1.2 Навчитися створювати презентації за допомогою засобів програм для підготовки презентацій.

Завдання до роботи

1.3.1 Ознайомитися з основними теоретичними відомостями за темою роботи, використовуючи дані методичні вказівки, лекційні матеріали та рекомендовану літературу.

1.3.2 Вивчити можливості однієї з сучасних програм підготовки презентацій, зокрема дослідити пункти головного меню та навчитися:

- створювати та зберігати нові презентації;
- використовувати шаблони слайдів та самостійно змінювати оформлення;
- працювати з місцезаповнювачами;
- встановлювати ефекти анімації елементів слайду;
- встановлювати ефекти переходу при зміні слайдів;
- створювати елементи навігації за презентацією;
- вставляти в презентацію таблиці, графіки, малюнки, кліпи та звукові файли, створені за допомогою інших програм.

1.3.3 Підготувати доповідь за темою, що відповідає індивідуальному завданню, обраному з таблиці А.1 за номером варіанту та узгодженому з викладачем.

1.3.4 Створити презентацію за підготовленою доповіддю, що складається не менше ніж з 10 слайдів.

При створенні презентації обов'язково використовувати такі засоби програм підготовки презентацій:

- зміна кольорової схеми презентації;
- ефекти анімації;
- ефекти переходу;
- розгалуження;
- нотатки.

1.3.5 Оформити звіт з роботи.

1.3.6 Відповісти на контрольні запитання.

Короткі теоретичні відомості

Програми підготовки презентацій дозволяють розв'язати безліч проблем, пов'язаних з підготовкою різних доповідей, звітів, розрахунків та аналізу даних. Такі програми розробляються для того, щоб полегшити користувачеві роботу з різними даними, систематизувати їх і зробити більш наочними.

Наявність програми підготовки презентацій у складі офісного пакету стала стандартом на даний момент, наприклад: PowerPoint – у Microsoft Office, Keynote – у iWork, Impress – у LibreOffice та Apache OpenOffice.

Текст доповіді

Вступ:

Swift - це універсальна, мультипарадигмальна, компільована мова програмування, розроблена компанією Apple Inc. у 2014 році для macOS, iOS, watchOS та tvOS. Swift призначена для роботи з фреймворками Apple Cocoa та Cocoa Touch, а також з великою кількістю існуючого коду на Objective-C, написаного для продуктів Apple. Це потужна та інтуїтивно зрозуміла мова, яку легко вивчати та використовувати.

Особливості мови програмування Swift:

1. Безпека та швидкість:

Swift розроблена, щоб бути безпечною, швидкою та ефективною. Вона побудована з потужною типізацією, яка запобігає поширеним помилкам програмування, таким як винятки з нульовим покажчиком та неініціалізовані змінні. Swift також має вбудовану систему керування пам'яттю, яка автоматично керує життєвим циклом об'єктів, звільняючи пам'ять, коли вона більше не потрібна. Завдяки цьому програми на Swift працюють швидше та ефективніше, ніж програми іншими мовами програмування.

2. Легкість у вивченні:

Swift легко вивчити навіть початківцям. Вона має чіткий синтаксис, який легко читати та розуміти. Swift також надає функцію ігрового майданчика, яка дозволяє розробникам експериментувати з кодом і бачити результати в режимі реального часу. Це робить вивчення Swift цікавим та інтерактивним.

3. Об'єктно-орієнтованість:

Swift є об'єктно-орієнтованою мовою програмування, що означає, що вона базується на концепції об'єктів. Об'єкти - це екземпляри класів, які є шаблонами, що визначають властивості та методи кожного об'єкта.

Swift також підтримує інкапсуляцію, успадкування та поліморфізм, які є ключовими особливостями об'єктно-орієнтованого програмування.

4. Функціональне програмування:

Swift також підтримує функціональне програмування, яке є парадигмою програмування, що наголошує на використанні функцій для створення багаторазового коду. Swift забезпечує першокласну підтримку функцій, закриття та функцій вищого порядку, що є основними характеристиками функціонального програмування.

5. Опціонали:

Опціонали - це потужна функція Swift, яка дозволяє розробникам безпечно працювати з нульовими величинами. У Swift кожна змінна та властивість за замовчуванням є опціональною, що означає, що вона може мати значення або бути нульовою. Це усуває необхідність у виняткових ситуаціях з нульовим показником і робить код більш надійним та безпечним.

6. Приведення типів:

Swift забезпечує виведення типів, що означає, що компілятор може автоматично визначити тип змінної на основі її початкового значення. Це робить код більш стислим і читабельним, оскільки розробникам не потрібно явно оголошувати тип кожної змінної.

7. Узагальнення:

Узагальнення - це потужна функція Swift, яка дозволяє розробникам писати узагальнений код, що може працювати з будь-яким типом. Це робить код більш придатним для багаторазового використання та зменшує дублювання коду. Swift надає вбудовану підтримку узагальнень, що полегшує написання універсального коду.

8. Закриття:

Закриття - це потужна функція Swift, яка дозволяє розробникам писати автономні блоки коду, які можна передавати як значення. Закриття

схожі на функції, але вони можуть перехоплювати і зберігати посилання на будь-які константи і змінні з навколишнього контексту. Це робить закриття потужним інструментом для написання асинхронного та керованого подіями коду.

9. Інтєроперабельність:

Swift розроблено для безперешкодної роботи з кодом на Objective-C та C. Це означає, що розробники можуть використовувати існуючі бібліотеки та фреймворки, написані на Objective-C та C, у своїх проєктах на Swift. Swift також надає прості у використанні API для роботи з кодом на C та Objective-C.

10. Ігровий майданчик:

Ігровий майданчик - це унікальна функція Swift, яка дозволяє розробникам експериментувати з кодом і бачити результати в реальному часі. Ігрові майданчики забезпечують інтерактивне середовище для написання та тестування коду, що робить навчання та експерименти зі Swift приємними та цікавими.

Висновок:

Swift - це потужна та інтуїтивно зрозуміла мова програмування, яку легко вивчати та використовувати. Вона розроблена задля забезпечення надійності, швидкості та ефективності, має потужну систему типів та вбудовані засоби керування пам'яттю. Swift підтримує як об'єктно-орієнтоване, так і функціональне програмування, з такими корисними можливостями, як опції, узагальнення та закриття. Swift також забезпечує безперешкодну роботу з кодом на Objective-C та C, що дозволяє розробникам використовувати існуючі бібліотеки та фреймворки у своїх проєктах на Swift. Завдяки зрозумілому синтаксису, потужним функціям та інтерактивному середовищу, Swift є чудовим вибором для розробки програмного забезпечення для платформ Apple.

Можливості мови програмування Swift

Онищенко Олег

План презентації

- | | |
|---------------------------------------|-------------------------------|
| I. <u>Вступ</u> | VII. <u>Приведення типів</u> |
| II. <u>Безпека та швидкість</u> | VIII. <u>Узагальнення</u> |
| III. <u>Легкість у вивченні</u> | IX. <u>Закриття</u> |
| IV. <u>Об'єктно-орієнтованість</u> | X. <u>Інтероперабельність</u> |
| V. <u>Функціональне програмування</u> | XI. <u>Ігровий майданчик</u> |
| VI. <u>Опціонали</u> | XII. <u>Висновки</u> |

Вступ



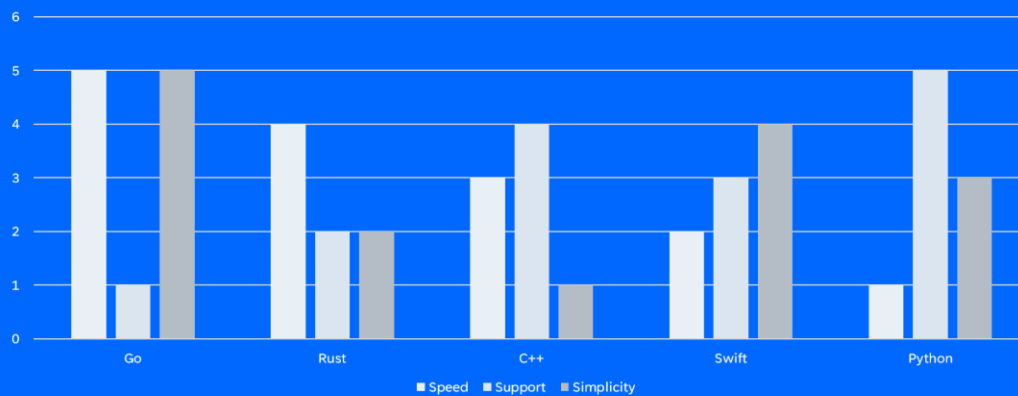
2014



20.04.2023

3

Безпека та швидкість



20.04.2023

4

Легкість у вивченні

```
1 let myArray = ["apple", "banana", "orange"]
2 for fruit in myArray {
3   print(fruit)
4 }
```

```
1 if myVariable == 5 {
2   print("The variable is equal to 5")
3 } else {
4   print("The variable is not equal to 5")
5 }
```

```
1 func addNumbers(a: Int, b: Int) → Int {
2   return a + b
3 }
```

20.04.2023

5

Об'єктно-орієнтованість



Class



Object

20.04.2023

6

Функціональне програмування

```
1 let numbers = [1, 2, 3, 4, 5]
2
3 // Map function to double each number in the array
4 let doubledNumbers = numbers.map { \${0} * 2 }
5
6 // Filter function to get only even numbers
7 let evenNumbers = numbers.filter { \${0} % 2 == 0 }
8
9 // Reduce function to get the sum of all numbers in the array
10 let sum = numbers.reduce(0, +)
```

20.04.2023

7

Опціонали

```
1 var optionalString: String? = "Hello, world!"
2
3 // Unwrap the optional using optional binding
4 if let unwrappedString = optionalString {
5     print(unwrappedString)
6 } else {
7     print("The optional string is nil.")
8 }
9
10 // Force unwrap the optional (not recommended)
11 print(optionalString!)
12
13 // Use optional chaining
14 let count = optionalString?.count ?? 0
15 print("The optional string has \${count} characters.")
```

20.04.2023

8

Узагальнення

```
1 func findLarger<T: Comparable>(first: T, second: T) → T {  
2   if first > second {  
3     return first  
4   } else {  
5     return second  
6   }  
7 }
```

```
1 let intResult = findLarger(first: 5, second: 10)  
2 print(intResult) // Output: 10  
3  
4 let doubleResult = findLarger(first: 3.14, second: 2.71)  
5 print(doubleResult) // Output: 3.14  
6  
7 let stringResult = findLarger(first: "hello", second: "world")  
8 print(stringResult) // Output: world
```

Закриття

```
1 // Define a closure that takes two Int parameters and returns an Int  
2 let addClosure: (Int, Int) → Int = { (a: Int, b: Int) → Int in  
3   return a + b  
4 }  
5  
6 // Call the closure with arguments and print the result  
7 let result = addClosure(2, 3)  
8 print(result) // Output: 5
```

Інтероперабельність

```
1 // C++ code
2 #include <iostream>
3
4 extern "C" {
5     int add(int x, int y) {
6         return x + y;
7     }
8 }
9
10 // Swift code
11 let result = add(3, 4)
12 print("The result is \(result)")
```

20.04.2023

12

Ігровий майданчик

```
1 import UIKit
2 import PlaygroundSupport
3
4 let view = UIView(frame: CGRect(x: 0, y: 0, width: 200, height: 200))
5 view.backgroundColor = .red
6
7 let label = UILabel(frame: CGRect(x: 50, y: 50, width: 100, height: 50))
8 label.text = "Hello, Playgrounds!"
9 label.textColor = .white
10 label.textAlignment = .center
11
12 view.addSubview(label)
13
14 PlaygroundPage.current.liveView = view
```

20.04.2023

13



ВИСНОВКИ



Дякую за увагу

ВИСНОВКИ

Таким чином, ми вивчили основні можливості сучасних програм для підготовки презентацій та навчилися створювати презентації за допомогою засобів програм для підготовки презентацій.

Контрольні запитання

1.5.1 Поясніть призначення пунктів головного меню програми для створення презентацій.

Пункти головного меню в програмному забезпеченні для створення презентацій призначені для того, щоб допомогти користувачам створювати та редагувати презентації. Нижче наведено деякі з найпоширеніших пунктів головного меню та їхнє призначення:

- **Файл:** це меню дозволяє створювати, відкривати, зберігати та друкувати презентації.
- **Основне:** це меню містить опції для редагування вмісту презентації, зокрема копіювання, вставки та скасування.
- **Конструктор:** Це меню дає змогу налаштувати спосіб відображення презентації, наприклад, збільшити або зменшити масштаб, показати або приховати лінійку та перемикатися між різними поданнями.
- **Вставлення:** Це меню містить параметри для додавання нового вмісту до презентації, зокрема текстових полів, зображень, відео та діаграм.
- **Форматування:** Це меню дає змогу формувати вміст презентації, наприклад, змінити шрифт, колір або тло слайдів.
- **Показ слайдів:** Це меню містить параметри для показу слайдів, наприклад, запуск слайд-шоу, налаштування переходів між слайдами та репетиції синхронізації.
- **Довідка:** Це меню надає доступ до довідкових ресурсів, таких як посібник користувача або онлайн-підтримка.

1.5.2 Як змінити шаблон макета слайдів?

Щоб змінити шаблон макета слайдів у більшості програм для створення презентацій, виконайте такі дії:

- Відкрийте презентацію, яку потрібно відредагувати.
- Перейдіть до слайда, для якого потрібно змінити макет.
- Знайдіть пункт "Макет" у головному меню або на панелі інструментів.
- Натисніть на "Макет", щоб побачити список доступних шаблонів макетів.
- Виберіть шаблон, який ви хочете використати для свого слайда.
- Слайд буде оновлено відповідно до нового шаблону.

1.5.3 Як змінити метод розмітки для слайда?

Спосіб розмітки слайда в програмі для створення презентацій - це спосіб додавання анотацій і коментарів до слайда. Щоб змінити спосіб розмітки, виконайте такі дії:

- Відкрийте презентацію, яку потрібно відредагувати.
- Перейдіть до слайда, до якого потрібно додати анотації або коментарі.
- Знайдіть пункт "Рецензування" в головному меню або на панелі інструментів.
- Натисніть на "Рецензування", щоб побачити список доступних інструментів розмітки.
- Виберіть інструмент розмітки, який ви хочете використовувати, наприклад, ручку або маркер.
- Почніть додавати анотації або коментарі до слайда за допомогою вибраного інструменту розмітки.

1.5.4 Як збільшити/зменшити шрифт тексту? Які типи вирівнювання тексту доступні?

Щоб збільшити або зменшити шрифт тексту в більшості програм для створення презентацій, виконайте такі дії:

- Виділіть текст, для якого ви хочете змінити розмір шрифту.
- Знайдіть пункт "Шрифт" у головному меню або на панелі інструментів.
- Натисніть на "Шрифт", щоб побачити список доступних розмірів шрифтів.
- Виберіть розмір шрифту, який ви хочете використовувати для виділеного тексту.

Типи вирівнювання тексту, доступні в більшості програм для створення презентацій, включають

- Вирівнювання по лівому краю: Вирівнює текст за лівим краєм слайда.
- Вирівнювання по центру: Вирівнює текст по горизонталі на слайді.
- Вирівнювання по правому краю: Вирівнює текст за правим краєм слайда.
- Вирівнювання по ширині: Вирівнює текст за лівим і правим краєм слайда, створюючи чисті краї з обох боків.

Щоб змінити вирівнювання тексту в більшості програм для створення презентацій, виконайте такі дії:

- Виділіть текст, для якого потрібно змінити вирівнювання.
- Знайдіть пункт "Вирівнювання" в головному меню або на панелі інструментів.
- Натисніть "Вирівнювання", щоб побачити список доступних варіантів вирівнювання.

- Виберіть варіант вирівнювання, який ви хочете використовувати для виділеного тексту.

1.5.5 Які відмінності між PowerPoint 2003 і 2010?

Існує кілька відмінностей між PowerPoint 2003 і 2010, зокрема

- Інтерфейс користувача: Інтерфейс користувача PowerPoint 2010 було оновлено за допомогою стрічки, яка забезпечує більш впорядкований і контекстний спосіб доступу до команд і функцій.
- Формати файлів: PowerPoint 2010 за замовчуванням використовує формат файлів Office Open XML (.pptx), тоді як PowerPoint 2003 використовує старіший двійковий формат файлів (.ppt).
- Спільна робота: PowerPoint 2010 містить більше можливостей для спільної роботи, таких як співавторство, коментування та керування версіями, які були недоступні в PowerPoint 2003.
- Мультимедіа: PowerPoint 2010 містить більше мультимедійних функцій, таких як редагування та вбудовування відео, які були недоступні в PowerPoint 2003.
- Безпека: PowerPoint 2010 містить більше функцій безпеки, зокрема захищене подання, які допомагають захиститися від шкідливого програмного забезпечення та інших загроз.