## Kotlin Exam

\*\*Instructions:\*\*

- This exam consists of both theory and practice questions.

- The exam duration is 2 hours.

- Answer all the questions.

- For theory questions, provide your answers in the space provided.

- For practice questions, write the code in the space provided.

\*\*Theory Questions:\*\*

1. What is Kotlin? How is it different from Java?

- Your Answer:

Kotlin là một ngôn ngữ lập trình, tạo ra để them các tính năng để cải thiện hơn so với Java.

Điểm khác biệt:

* Null Safety: giúp tránh lỗi NullPointerException
* Extension function: cho phép mở rộng một class mà không cần thay đổi mã nguồn ban đầu.
* Data class: lớp chỉ chứa dữ liệu, được cung cấp sẵn một vài phương thức( toString, …)
* Coroutine: một cách an toàn để xử lí bất đồng bộ
* Default Parameter và Name Argument: cho phép định nghĩa giá trị theo tên để dễ đọc

2. What are the key features of Kotlin?

- Your Answer:

Một số điểm đặc biệt của Kotlin:

* Null safety
* Exxtension function
* Cú pháp ngắn gọn
* Có thể tương tác với Java
* Coroutine, Data class, …

3. Explain the following terms in Kotlin:

- a. Null Safety:

Giúp tránh lỗi NullPointerException.

Phải xác định rõ xem biến có null hay không:

+ Biến có thể null: để xác định biến có thể null dùng ?

+ Biến không thể null: để xác định biến không thể null dùng !!

- b. Data Classes: class dùng để lưu và truyền dữ liệu

+ Tự động tạo các phương thức equals(), toString(),…

+ Các thuộc tính có thể được khai báo là val hoặc var

- c. Coroutines:

+ Một thứ để xử lí bất đồng bộ một cách hiệu quả.

+ Giúp mã dễ đọc , dễ hiểu, dễ bảo trì so với thread, callback,..

Một vài đặc điểm:

+ Suspend function

+ Coroutine scope: một phạm vi để tạo coroutine và thực thi

+ Coroutine context: môi trường thực thi coroutine

+ Dispatcher: Main, IO, Defaault, Unconfined xác định rõ coroutine hoạt động ở đâu

4. What are the primary differences between `val` and `var` in Kotlin?

- Your Answer:

Val: không thể gán lại giá trị mới

Var: có thể thay đổi, gán giá trị mới

5. Explain the difference between functions and higher-order functions in Kotlin.

- Your Answer:

Điểm khác nhau:

* Hàm: là một khổi thực hiện một công việc cụ thể.
* Higher-order function: hàm có thể nhận một hoặc nhiều hàm khác làm tham số hoặc giá trị trả về.

\*\*Practice Questions:\*\*

1. Write a Kotlin class named `Student` that has the following properties: `name` (String), `age` (Int), `roomNumber` (Int), and `gradeLevel` (Int). Provide a function named `displayInfo()` that prints the student's information.

```kotlin

// Your code here

```

2. Write a Kotlin class named `Room` that has a property named `number` (Int) and a property named `capacity` (Int). Provide a function named `isFull()` that returns true if the room is full (i.e., the number of students in the room is equal to the capacity).

```kotlin

// Your code here

```

3. Write a Kotlin program that creates an instance of the `Room` class and checks if the room is full.

```kotlin

// Your code here

```

4. Create a superclass named `Resident` with a function named `checkIn()` that prints "Resident checked in." Create a subclass named `Student` that inherits from `Resident` and overrides the `checkIn()` function to print "Student checked in." Create an instance of the `Student` class and call the `checkIn()` function.

```kotlin

// Your code here

```

5. Write a Kotlin interface named `Paymentable` with a function named `makePayment()` that calculates and returns the total payment for a dormitory resident. Create a class named `Student` that implements the `Paymentabcle` interface and provides its own implementation of the `makePayment()` function. Create an instance of the `Student` class and call the `makePayment()` function.

```kotlin

// Your code here

```

6. Write a Kotlin class named `Dormitory` that has a property named `residents` (a list of `Student` objects). Provide a function named `getTotalOccupancy()` that calculates and returns the total number of students in the dormitory.

```kotlin

// Your code here

```

7. Create an instance of the `Dormitory` class and demonstrate the usage of the `getTotalOccupancy()` function.

```kotlin

// Your code here

```

8. Explain the concepts of inheritance and polymorphism in Kotlin.

- Your Answer:

Kế thừa:

* Tạo ra một lớp mới từ một lớp đã tồn tại gọi là lớp cha. Lớp vừa kế thừa là lớp con.
* Lớp con kế thừa tất cả các thuộc tính và phương thức của lớp cha và có thể them các thuộc tính hoặc phương thức riêng

Đa hình:

* Một việc có thể thực hiện theo nhiều cách khác nhau.
* Đa hình thực hiện thông qua kế thừa và override lại các phương thức

9. What is an abstract class in Kotlin? How does it differ from an interface?

- Your Answer:

Abstract class:

* Là một lớp không thể tạo ra một đối tượng cụ thể mà như là một bản thiết kế cho lớp con.
* Một abstract class có thể chứa nhiều phương thức trừu tượng.
* Không thể sử dụng tự khoá new để tạo trực tiếp

So sánh với interface:

* Một lớp con chỉ có thể kế thừa từ một abstract class và có thể kế thừa nhiều interface
* Abstract class được dùng như một bản thiết kế và triển khai phương thức cho lớp con. Interface sử dụng khi cần chỉ rõ đối tượng có hành vi nào.
* Abstract class có phương thức triển khai, interface chỉ định các lớp implement tự triển khai các phương thực

10. Explain the concept of encapsulation in OOP and how it is implemented in Kotlin.

- Your Answer:

Tính đóng gói cho phép che dấu thông tin.

* Private: chỉ có thể truy cập khi trong cùng một lớp
* Public: có thể truy cập ở khắp mọi nơi
* Protected: tương tự private nhưng có subclass
* Internal: có thể truy cập khi trong cùng 1 module

11. Describe the purpose and usage of the `super` keyword in Kotlin.

- Your Answer:

* Dùng để gọi phương thức cha.
* Truy cập thuộc tính của lớp cha
* Gọi constructor của lớp cha

---