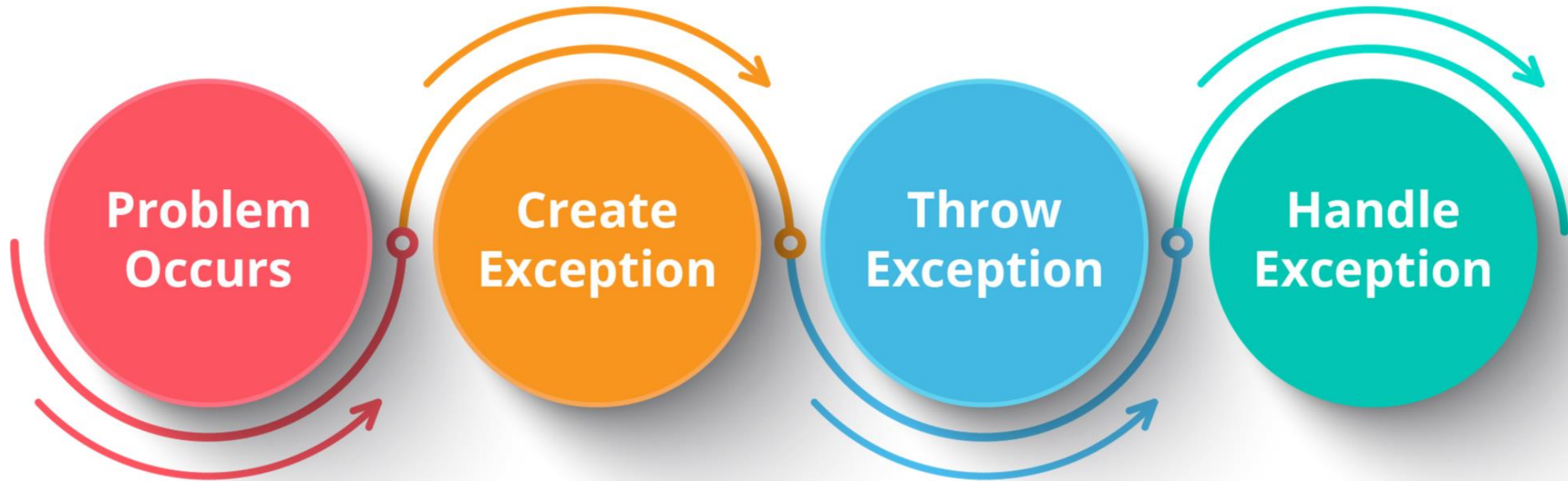


# Exception



# Khái niệm

- Là lỗi xảy ra trong quá trình biên dịch (compile) hoặc trong quá trình chạy (runtime)

# So sánh checked & unchecked

## Checked

- Là những lỗi có thể thấy tại thời điểm compile
- Ví dụ: IOException, SQLException, ...

## Unchecked

- Các ngoại lệ không được kiểm tra tại thời điểm biên dịch, nhưng được kiểm tra trong thời gian chạy.
- Ví dụ: ArithmeticException, NullPointerException, ...

# Một số từ khóa cần lưu ý

Keyword	Meaning
try	Sử dụng để chứa đoạn lệnh <b>có thể</b> gây ra ngoại lệ, nó phải được theo sau bởi khối 'catch' hoặc 'finally'
catch	Chứa các dòng code để xử lý ngoại lệ
finally	Thực thi đoạn mã cần thiết của chương trình (luôn được thực thi)
throw	Sử dụng để đưa ra ngoại lệ
throws	Sử dụng để khai báo ngoại lệ

# Xử lý ngoại lệ (Handle exception)

## 1. Sử dụng try-catch

```
try {  
    //Khởi lệnh có thể ném ra ngoại lệ  
} catch(<Exception_class_name> e) {  
    //Code xử lý ngoại lệ  
}
```

# Xử lý ngoại lệ (Handle exception)

## 2. Sử dụng try-finally

```
try {  
    //Khối lệnh có thể ném ra ngoại lệ  
} finally {  
    //Khối lệnh trong này luôn được thực thi  
}
```

# Xử lý ngoại lệ (Handle exception)

## 3. Sử dụng try-catch-finally

```
try {  
    //Khởi lệnh có thể ném ra ngoại lệ  
} catch(<Exception_class_name_1> e) {  
    //Code xử lý ngoại lệ 1  
} catch(<Exception_class_name_2> e) {  
    //Code xử lý ngoại lệ 2  
} finally {  
    //Khởi lệnh trong đây luôn được thực thi  
}
```

# Flowchart Diagram

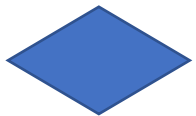
- Là sơ đồ thiết kế các bước xử lý nghiệp vụ cho từng quy trình
- Chú thích:



Các bước thực hiện



Bắt đầu hoặc kết thúc một quy trình



Kiểm tra điều kiện để đưa ra quyết định



# Flowchart Diagram

## QUY TRÌNH ONBOARDING NHÂN VIÊN MỚI

