howest
hogeschool

# Object Oriented Architectures and Secure Development

## Custom Exceptions

*Matthias Blomme*
*Mattias De Wael*
*Frédéric Vlummens*

# What we already know: try-and-catch

```
try {

   //  the code that might go wrong …

} catch (SomeException ex) {

   //  the code to recover from the exceptional situation.

}
```

# What we already know: throw

```
public void withdraw(int amount) {
    if (amount < 0) {
        LOGGER.log(Level.WARNING, "Possible steal attempt");
        throw new IllegalArgumentException("Illegal amount");
    }

    if (this.balance < amount) {
        throw new IllegalStateException("Insufficient funds");
    }

    this.balance -= amount;

}
```



howest hogeschool

# Creating your own exceptions

```
public class InsufficientFundsException extends RuntimeException {

    public InsufficientFundsException() {
        super();
    }

    public InsufficientFundsException(String message) {
        super(message);
    }

    public InsufficientFundsException(String message, Throwable cause) {
        super(message, cause);
    }
}
```



howest hogeschool

# Throwing your own exceptions

```
public void withdraw(int amount) {
    if (amount < 0) {
        throw new IllegalArgumentException("Illegal amount");
    }

    if (this.balance < amount) {
        throw new InsufficientFundsException();
    }

    this.balance -= amount;
}
```

# Creating your own exceptions

```
public class InsufficientFundsException extends RuntimeException {

    private final int balance;
    private final int amount;

    public InsufficientFundsException(int balance, int amount) {
        super(String.format("Insufficient Funds: cannot withdraw %d when the balance is only %d.", amount, balance));
        this.balance = balance;
        this.amount = amount;
    }

    public int getBalance() {
        return balance;
    }

    public int getAmount() {
        return amount;
    }
}
```



howest hogeschool

# Throwing your own exceptions

```
public void withdraw(int amount) {
    if (amount < 0) {
        throw new IllegalArgumentException("Illegal amount");
    }

    if (this.balance < amount) {
        throw new InsufficientFundsException(this.balance, amount);
    }

    this.balance -= amount;
}
```