

Informatique 1

# 11a. Flux de données

## *Les exceptions*

# Quels problèmes avec les fichiers ?



# A quoi servent les exceptions ?

1. Indiquer qu'un bout de code **peut poser** éventuellement problème.
2. **Lorsqu'un** problème se pose, le résoudre à un endroit donné.
3. Si le problème ne peut être résolu, **déléguer** plus loin.
4. Faire quelque chose **après** que le problème se soit passé.

Les exceptions

# Nouveaux mots-clés

1. Indiquer qu'un bout de code **peut poser** éventuellement problème.
2. **Lorsqu'un** problème se pose, le résoudre à un endroit donné.
3. Si le problème ne peut être résolu, **déléguer** plus loin.
4. Faire quelque chose **après** que le problème se soit passé.

**try**

**catch**

**(throw)**

**(finally)**

Les exceptions

# La structure throw-try-catch

- Implémentation :
  - ▶ Là où le bloc peut poser problème → **try**
  - ▶ Lorsque l'erreur arrive → **throw**
  - ▶ Pour gérer erreur → **catch**

```
int main() {  
    ...  
    try{  
        ...  
        if(...) throw Exception  
        ...  
    }  
    catch (Exception) {  
        ...  
    }  
}
```

# Exemple simple

- Pas de `throw`, pas de `finally`
- Pour nous, la plupart des cas

```
int a = Input.readInt();

try{
    int b = 35 / a;
    ...
}
catch (Exception e) {
    e.printStackTrace();
}
```

# Exemple d'exceptions