Universidade do Minho
Escola de Engenharia

Ferramentas e técnicas de *Compiler warnings*

GRUPO 1:

RICARDO PEREIRA

TIAGO RAMIRES

GRUPO 3:

ADRIANA MEIRELES

CARLA CRUZ

Introdução

- Existem outras formas de prevenir falhas de segurança no *software*?

- Medidas que se podem tomar durante a produção do código.

- ***Segmentation faults***, *compiler warnings* e **vulnerabilidades**.

Java

- Baseada em C/C++ e foi criada por James Gosling. Formalmente anunciada em 1995, o Java é uma linguagem de programação orientada a objetos.
- Esta linguagem é compilada para um *bytecode* que é interpretado por uma máquina virtual (Java Virtual Machine - JVM).
- O Java é rápido, seguro e confiável.

Funcionamento do Interpretador e Compilador

- Em Java, a compilação é instantânea traduzindo bytecodes para código máquina, sendo este executado posteriormente. Isto melhora o tempo de execução do programa.
- A JVM é um interpretador Java que carrega e executa as aplicações Java que estão em bytecodes, convertendo esses bytecodes em código executável de máquina.
- O equilíbrio entre interpretação e compilação evoluiu bastante com o passar do tempo, de modo que os programas Java executados com frequência sofrem pouquíssimo trabalho extra da interpretação.

Problemas Mais Comuns nas Outras Linguagens

- Input Validation Error
- BoundaryConditionError
- BufferOverflow
- Access Validation Error
- ExceptionalConditionError
- CrossSiteScripting
- SQLInjection

Exemplo noutra Linguagem

- BufferOverflow

  Nesta situação são ultrapassados os limites de um array para executar um código malicioso.

- FormatString

  Pretende-se:

  - Conseguir informações sobre os conteúdos dos endereços de memória que normalmente não são acessíveis
  - Conseguir uma “shell” de um super-utilizador a partir de uma exceção provocada pela entrada de uma string mal intensionada

Vulnerabilidades em Java

- Exposição dos Bytecodes

- Objetos String e Garbage Collection

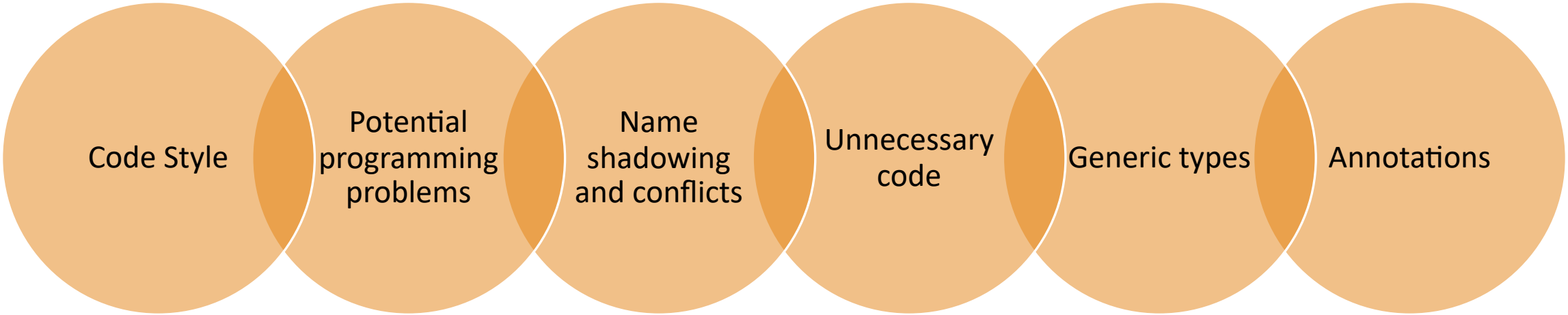
- Literais de String

Conclusões:

As questões de segurança em Java devem ser vistas de forma diferente das outras linguagens

O programador é o maior responsável pelas vulnerabilidades da aplicação

Compiler Warnings em Java



```
$ javac *.java
Note: Some input files use unchecked or unsafe operations.
Note: Recompile with -Xlint:unchecked for details.
```

```
$ javac -Xlint:unchecked *.java
LegacyDragons.java:9: warning: [unchecked] unchecked call to add(E) as a member
of the raw type List
    unicorns.add(new Unicorn());
                ^
  where E is a type-variable:
    E extends Object declared in interface List
LegacyDragons.java:11: warning: [unchecked] unchecked method invocation: method
printDragons in class LegacyDragons is applied to given types
    printDragons(unicorns);
                ^
  required: List<Dragon>
  found: List
LegacyDragons.java:11: warning: [unchecked] unchecked conversion
    printDragons(unicorns);
                 ^
  required: List<Dragon>
  found:    List
3 warnings
```

Warnings na Prática

Python

- É uma linguagem de programação orientada aos objetos;
- Alto nível;
- Suporta módulos e pacotes;
- Incentiva à reutilização de código.

1. Funcionamento do Compilador

- É uma linguagem de programação interpretada e compilada;
- O compilador traduz a linguagem Python para bytecode;
- O interpretador executa este bytecode numa Máquina Virtual;
- Caso existam erros, o compilador gera um relatório de erros e o interpretador interrompe a tradução quando encontra o primeiro erro.

2. Warnings

- São, geralmente, emitidos quando o encerramento do programa não é garantido;
- Os Warnings são emitidos recorrendo-se à função **warn()**.
- Existem dois estados no controlo de warnings:
  - Primeiro, é feita uma determinação se uma mensagem deve ou não ser emitida que é controlada por um filtro, recorrendo-se à função **filterwarnings()**.
  - O segundo estado é que caso uma mensagem seja emitida, é impressa recorrendo à função **showwarning ()** e para a sua formatação é chamada a função **formatwarning ()**

2. Warnings

- Existem várias categorias de warnings:
  - Warning;
  - UserWarning;
  - DeprecationWarning;
  - SyntaxWarning;
  - RuntimeWarning;
  - FutureWarning;
  - PendingDeprecationWarning;
  - ImportWarning;
  - UnicodeWarning;
  - BytesWarning;
  - ResourceWarning.

2. Warnings

- Existem os filtros de warnings que decidem se os warnings são ignorados, exibidos ou transformados em erros:
  - Ação;
  - Mensagem;
  - Categoria;
  - Módulo;
  - Lineno.

3. Warnings na Prática

- **Gerar Warnings:**

```
import warnings

print ('Before the warning')
warnings.warn('This is a warning message')
print ('After the warning')
```

```
thug80:TP2$ python -W "error::UserWarning::0" warnings_warn.py
Before the warning
Traceback (most recent call last):
  File "warnings_warn.py", line 4, in <module>
    warnings.warn('This is a warning message')
UserWarning: This is a warning message
```

3. Warnings na Prática

- **Filtrar com padrões:**

```
import warnings

warnings.filterwarnings('ignore', '.*do not.*',)

warnings.warn('Show this message')
warnings.warn('Do not show this message')
```

```
thug80:TP2$ python warnings_filterwarnings_message.py
warnings_filterwarnings_message.py:5: UserWarning: Show this message
  warnings.warn('Show this message')
```

3. Warnings na Prática

- **Warnings repetidos:**

```
import warnings

warnings.simplefilter('once', UserWarning)

warnings.warn('This is a warning!')
warnings.warn('This is a warning!')
warnings.warn('This is a warning!')
```

```
thug80:TP2$ python warnings_once.py
warnings_once.py:5: UserWarning: This is a warning!
  warnings.warn('This is a warning!')
```

Conclusão

- Devem ser tidas em conta boas práticas para a construção de *software;*

- Estas ferramentas são essenciais para para garantir *softwares* fidedignos e funcionais;

- **A aplicação destas ferramentas não tem qualquer custo** !

- A redução de bugs é garantida;

- As boas práticas devem ser ensinadas desde logo **no período de formação** dos programadores.

Universidade do Minho
Escola de Engenharia

Ferramentas e técnicas de *Compiler warnings*

GRUPO 1:

RICARDO PEREIRA

TIAGO RAMIRES

GRUPO 3:

ADRIANA MEIRELES

CARLA CRUZ