D0:

Rigorous Methods for Software Engineering (F21RS)

(2023-2024)

Specification of Coursework 1

A SPARK High Integrity Software Development Exercise

HWU ID: H00317921

Username: jc2154

Name: James Cockfield

Campus: Edinburgh

Contents

[D1: Assumptions. 3](#_Toc148182413)

[D2: Design. 3](#_Toc148182414)

[D3: Source Code. 3](#_Toc148182415)

[D4: gnatprove summary using flow. 3](#_Toc148182416)

[D5: gnatprove summary using prove. 3](#_Toc148182417)

[D6: ADLS log file. 3](#_Toc148182418)

[D7: Describe how you protected against run-time exceptions. 3](#_Toc148182419)

[D8: Compare and contrast your chosen language with SPARK. 3](#_Toc148182420)

[Java 3](#_Toc148182421)

[Language Features of Java 3](#_Toc148182422)

[Strengths of Java 4](#_Toc148182423)

[Limitations of Java 4](#_Toc148182424)

[SPARK 4](#_Toc148182425)

[Language Features of SPARK 4](#_Toc148182426)

[Strengths of SPARK 4](#_Toc148182427)

[Limitations of SPARK 4](#_Toc148182428)

[Comparison of Java and SPARK 4](#_Toc148182429)

[Conclusion 5](#_Toc148182430)

# D1: Assumptions.

* I assume that there is a system that looks after the sensors on the wheels where if the three sensors have different values it will tell the driver there is a fault. The impact of this was small as no implementation was needed to catch this edge case.

# D2: Design.

# D3: Source Code.

# D4: gnatprove summary using flow.

# D5: gnatprove summary using prove.

# D6: ADLS log file.

# D7: Describe how you protected against run-time exceptions.

# D8: Compare and contrast your chosen language with SPARK.

Java and SPARK are two programming languages with distinct features, designed for different purposes. In this comparison, we will explore their language features through code examples, and highlight two strengths and two limitations of each language.

## Java

### Language Features of Java

Java is a widely used, general-purpose object-oriented language. It promotes the use of classes and objects to structure code.

1. class Person {

2. private String name;

3.

4. public Person(String name) {

5. this.name = name;

6. }

7.

8. public String getName() {

9. return name;

10. }

11. }

12.

Java code is compiled into an intermediate form (bytecode) that can run on any platform with a Java Virtual Machine (JVM), providing platform independence.

### Strengths of Java

* Java boasts a vast ecosystem of libraries and frameworks, making it suitable for a wide range of applications, from web development (Spring Boot) to mobile apps (Android) and enterprise systems (Java EE).
* Java has a large and active community, ensuring constant updates, support and a wealth of resources for developers.

### Limitations of Java

* The need to run the JVM introduces the performance overhead compared to natively compiled languages.
* Java can be verbose, requiring more lines of code for certain tasks compared to other modern languages.

## SPARK

### Language Features of SPARK

SPARK is known for its emphasis on formal verification, using mathematical proof to ensure program correctness. It includes contracts for specifying program behaviour.

1. package Math is

2. X : Integer;

3. with Pre => X > 0, Post => X \* X > 0;

4. end Math;

5.

SPARK is designed for safety-critical and high-integrity systems, making it suitable for industries like aerospace and healthcare.

### Strengths of SPARK

* SPARK’s formal verification and strong typing help create highly reliable, safety-critical systems where errors can have catastrophic consequences.
* SPARK’s strict typing and verification processes lead to predictable program behaviour, reducing unexpected errors in critical applications.

### Limitations of SPARK

* SPARK’s focus on formal verification and strict typing can be challenging for developers unfamiliar with these concepts, potentially leading to slower development.
* SPARK has a smaller community and less extensive library support compared to Java. It may lack readily available libraries for common, non-safety-critical tasks.

## Comparison of Java and SPARK

Java is object-oriented, emphasizing code reusability and maintainability through classes and objects. On the other hand, SPARK focuses on formal verification, aiming to guarantee program correctness for safety-critical systems, often in single-threaded or lightly concurrent contexts.

SPARK is more platform-independent in theory but is often used in environments where predictability and control take precedence over platform flexibility. Whereas Java’s bytecode allows it to run on various platforms with JVM support.

## Conclusion

In conclusion, Java and SPARK are distinct programming languages, each tailored to specific needs. Java is versatile and popular, with a vast ecosystem, while SPARK excels in safety-critical applications with its strong emphasis on reliability and formal verification. The choice between them should consider the project’s requirements, safety considerations, and the developer’s familiarity with the language and its ecosystem.