



DEXAR

2020-21



EVERY PROGRAMMER IS AN
“AUTHOR”

DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING

VISION

- **To Provide Technical Education In The Field Of Computer Science And Engineering By Imparting Employability, Research Capabilities, Entrepreneurship With Human Values.**

MISSION

- **Adopting Innovative Teaching Learning Practices Through The State-Of-The-Infrastructure.**
- **Establishing A Conducive Environmentfor Innovations And Research Activities.**
- **Inculcating Leadership Skills,Moral And Social Values Through Extension Activities.**

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO 1

The Graduates Will Excel in Their chosen technical domains In Computer Science And Engineering.

PEO 2

The Graduates Will Solve Real World Problems In Computerscience And Engineering By Applying Logical And Coding Skills Considering Ethical Values.

PEO 3

The Graduates Will Work Enthusiastically As A Team And Engage In Life Long Learning.



Joel Johnson

A few years ago, Mark Reinhold, Chief Architect of the Java Platform Group at Oracle, wrote in a blog post titled "Moving Java ForwardFaster" that Java needed to advance more quickly in order to become competitive.

A Java framework is specific to the Java programming language and serves as a platform for creating Java applications and web applications. The aim of frameworks is to provide a common structure so that developers don't have to redo it from scratch and can reuse the code provided. It also allows designers and developers to focus on creating unique features for their web-based projects rather than re-inventing the wheel through coding. Spring is one of the most widely-used Java frameworks primarily for the development of web applications. Spring supports such things as application events and listeners, externalized configuration, YAML, and type-safe configuration. Micro service frameworks can be used for deploying Java. Spring Boot is probably the best Microservice in the Java framework that works on top languages for Inversion of Control, Aspect-Oriented Programming, etc

JavaScript (js) is a light-weight object-oriented programming language. JavaScript is so popular that it's the most used programming language in the world, used as a client-side programming language by 98.0% of all websites. Apache Parquet is an open source, column-oriented data file format designed for efficient data storage and retrieval. It has seen widespread adoption for fast analytical querying. The required skill for the upcoming years is to understand the basics on Java frameworks, JavaScript and microservices can enrich the developing knowledge of each and everyone.

Editorial Student Member : Joel Johnson.S



LIST OF CONTENTS

Java - based wheel of joy	01
Java prototypes	02
Updating Java	05
Do it Learnit	
Troop selection	07
Strange Counter	08
Star pattern	09
Indian Flag	10
Creating Parquet files in Java	11
The Spring framework	12
Circular progress indicator bar display	14
Project Glimpses	16
Do it learnit Solutions	18

Introduction

The goal of this project was to develop a Java class that creates a straightforward "wheel-of-joy" control type, which is just a wheel divided into n sections of equal size and has a tick in the middle of the right side (3 o'clock position) that indicates the current selection. The concept was that the mouse could be used to rotate the wheel, spin it with an initial speed, and decelerate the rotation if the mouse button was released mid-motion. Based on where the tick is pointing after the wheel stops spinning, it would be possible to read the selected string.

Java - Based Wheel of Joy



Looping Drawing

In separate loop iteration, each section of the wheel is drawn. The simplest method for sketching the wheel was to simply turn it around piece by piece while drawing the required components. As many times as there are strings in the ArrayList, the loop iterates (which is equal to the number of sections of the wheel).

Wheel Rotation:

The actual rotation is carried out by a unique class called SpinRunnable, which implements the Runnable interface.

Tracking the Speed of Rotation:

```
private class speed Timer Task extends Timer Task {
/* Timer Task class that monitors and refreshes the _spin Speed
The speed is calculated as a difference of two rotation angles over a
period of time. We add the 360 to the "now" angle and then MOD it by
360 to avoid miscalculation when passing the full circle.*/
    public void run() {
double prevAngle, nowAngle; long sleepTime = 100;
        while(true) {
prevAngle = getRotationAngle();
            try {
                Thread.sleep(sleepTime);
            } catch (InterruptedException e)
            { e.printStackTrace();
            }
            nowAngle = getRotationAngle();
            nowAngle = (nowAngle + Math.signum(nowAngle) * 360) % 360;
            spinSpeed = Math.abs(nowAngle - prevAngle) * (1000 / sleepTime);
        }
    }
}
```



Tracking the Speed of Rotation :

There is also a different class that extends the Timer Task class that measures rotational speed.

Conclusion :

The wheel could be used to choose at random from a small pool of strings. There is a limit because the wheel can Only visually fit so many because it is divided into a many pieces as the size of the string array. This project was in a sense designed to experiment a little bit with java animation using Java AWT as I am relatively new to Java Program.



Introduction :

As a fun project, I decided to make some prototypes. They were mostly based on various open source OpenGL wrappers with a built-in window creation and management system. At the same time, I was trying to improve my C# skills, so I used platform-dependent modules in prototypes. The platform was usually Windows. Each prototype was made to solve certain problems and to test platform capabilities. When the prototypes grew enough, I began to analyze the future project, establish strict rules and develop goals. The prototypes were in C#, so I decided to include the .NET Core platform to make the project cross-platform (besides, at that time, it was impossible to create a UI app for .NET Core). My teammate used Java, so we decided to add the JVM platform to the project.

Together, this should also Support for the following programming languages perspectives:

- .NET platform: C#, Visual Basic, C++CLI
- JVM platform: Java, Scala, Kotlin

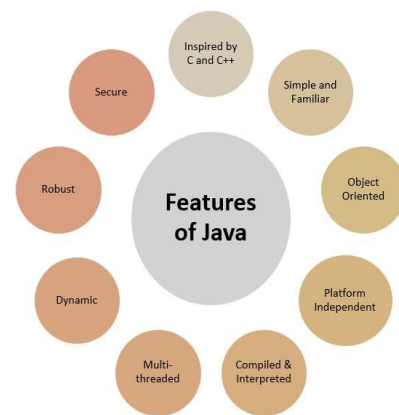
Using this approach, you can create your own library of items, because the main feature of the framework is not the elements, but the kitchen with the rules and recipes. Using the rules and recipes, you can create items of any complexity and for any purpose.

ITEM SYSTEM:

The primary data packet type in the framework is Item . The type is the fundamental building block of the framework and it is compatible with all systems.

THE MAIN RULE:

This indicates that there are two states that each object experiences: creation and initialization. It's crucial to remember that an item's fundamental functionality is only accessible after it enters the initialization state.



OTHER SKILLS

- ❖ The ability to analyze complex technical information
- ❖ We can analyze, design and implement database structures
- ❖ Excellent problemsolver

THEMES FOR APPLICATION STYLE:

- Initializing Common-based framework components. Common Service. at the beginning of the application, `initSpaceVILComponents()` (Main method). At this point, the OS, library accessibility, and OS dependencies are all examined. The system's fundamentals are set up, along with the overarching concept for all of the framework's components.
- Using `ActiveWindow` as the foundation, the window class is constructed and initialised (`InitWindow()` method). Typically, this process involves placing products and customising the window. The program entry point creates a window instance (Main method) Use a window manager (`WindowManager`) or the global window storage to call the `Show()` method of the window (`WindowsBox`).

THE STATES SYSTEM OF ELEMENTS AND USER INTERACTION WITH THE ELEMENTS:

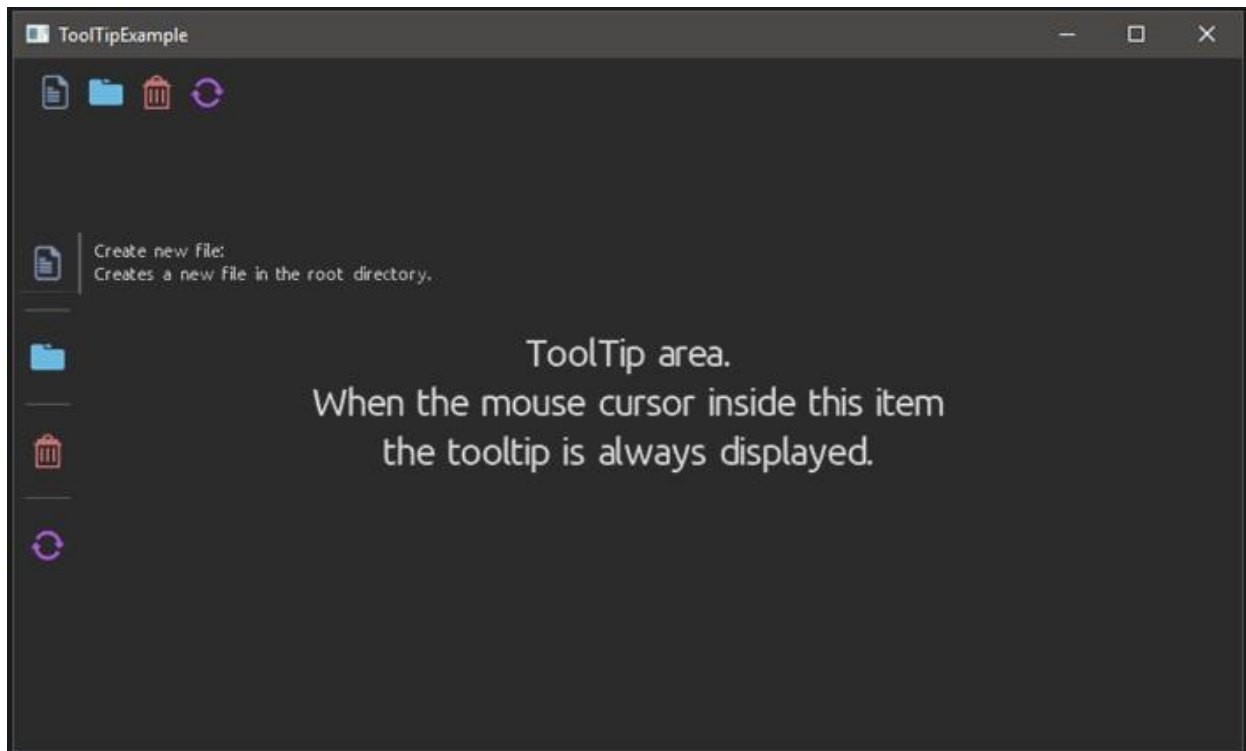
There are typically only six different ways for the user to engage with interactive elements:

- `ItemStateType.Base` (Basic static idlestate)
- `ItemStateType.Hovered` (Hoverstate)
- `ItemStateType.Pressed` (Pressedstate)
- `ItemStateType.Toggled` (Toggled state(on/off))
- `ItemStateType.Focused` (Focused state when an element receives events from the keyboard)
 - `ItemStateType.Disabled` (Disabled state, when an element ignores all events)

PROGRAM :

```
public static Style GetButtonStyle()
{
    Style style = Style.GetButtonCoreStyle();
    style.Background = Color.FromArgb(255, 13, 176, 255); //background color
    style.Foreground=Color.Black; // text color
    style.BorderRadius =newCornerRadius(6); //corner borderradius
    // take the default font and set it a new style and size
    style.Font = DefaultsService.GetDefaultFont(FontStyle.Regular, 18);
    // a button will occupy all available space
    style.SetSizePolicy(SizePolicy.Expand, SizePolicy.Expand);
    // set button alignment to center
    style.SetAlignment(ItemAlignment.HCenter,ItemAlignment.VCenter);
    // set text alignment in the button to center
    style.SetTextAlignment(ItemAlignment.HCenter,ItemAlignment.VCenter);
    // change the state of a button on hover style.ItemStates.Add
    (ItemStateType.Hovered, new ItemState(Color.FromArgb(60, 255, 255, 255)));
    return style;
}
```


Apps Screenshots on SpaceVIL

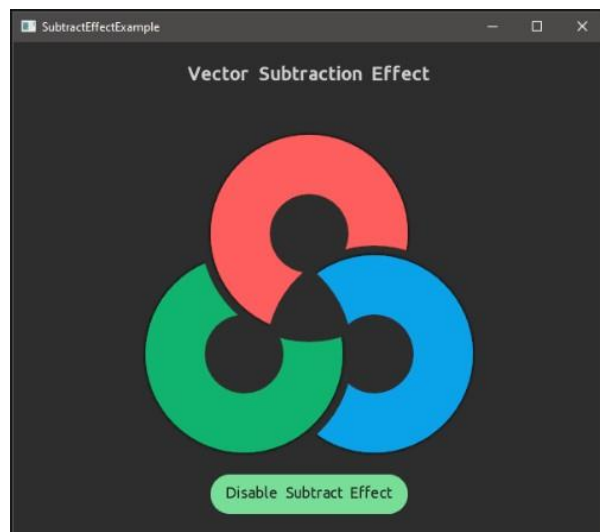


CONCLUSION:

SpaceVIL is a powerful, flexible and easy to use UI framework that can cover up to 80-90% of all types of desktop programs. To make the system flexible and capable of easy porting and development, its modules were strictly separated so that they could be easily replaced if necessary without harming the entire system. These separate modules became the core of the system. This is the essence of the SpaceVIL framework. That is all the system needs, and from this moment, it shows its true flexibility.

We will be glad if you try our framework in action and tell us your opinion about it.

DISABLE SUBTRACT EFFECT





A significant portion of our industry has always criticized Java to a significant extent. The majority of this criticism has been directed at Java's verbosity and the quantity of boilerplate code it frequently produces without necessity. I've always enjoyed Java, but I couldn't tell that these claims are untrue. It is true that Java can frequently become highly unpleasant due to its verbosity and amount of clutter. However, the majority of the time, we have to acknowledge that our world is not flawless and that, generally speaking, we must choose the lesser of two evils. We all agree that Java wasn't ideal, but the key point is why these issues weren't addressed earlier.

Personally, I believe that Java's lack of sufficient competition and the fact that things were fine as they were the sole reasons why changes took so long to implement. The Java language dominated the market most likely as a result of the absence of significant rivals and the significant efforts made by Sun initially and Oracle subsequently.

Java has become a very popular language for large projects because of the robust type safety it offers and some of its other qualities as a well-structured language. It's often more harder for situations to spiral out of control while using Java. Additionally, Java's key feature of being a multi-platform language that operates on its own virtual machine made it the ideal choice for many businesses. You have a very good set of reasons to adopt Java if you combine them with its innate ability to execute automated performance optimizations through its renowned JIT compiler, which minimizes the effects of poorly written code in many circumstances.

But then what happened? What happened is that new languages able to run in the same JVM as Java were introduced into the market, languages that were eliminating some of the biggest annoyances in Java and offering a nicer environment to developers with quite a flat learning curve, in some cases.

Before moving on, let's review and take a little tour of the JVM languages' past.

HISTORY OF JVM LANGUAGES

One thing I'd like to clarify before we start is that I've skipped some of the existing JVM languages mainly because they never got enough traction to be considered candidates for wide use in our industry. Having said that, let's kick off our quick tour through the history of JVM languages.

Java has been around for 24 years already; not bad, eh? Java was first introduced in January 1996. Java started off as an exclusively imperative language with a wholly object-oriented approach. It was also a tightly typed language. Although Java's syntax shares certain similarities with the C++ and C languages, it is thought to be an improvement because it is considerably simpler to write code in Java than in C or C++. On the other hand, its verbosity is cited as the main criticism by those who disagree with it.

Groovy was the second JVM language to be made available. Although its first formal and standardized version 1.0 didn't appear until January 2007, it has existed since 2003. The fact that Groovy could also be utilized as a scripting language was a plus. One of the reasons why some developers dislike Groovy is because it is a dynamically typed language, meaning type checks happen at runtime.



```
import java.io.*;
import java.util.Date;

public class SaveData {

    public static void main(String args[]) throws Exception {
        FileOutputStream fos = new FileOutputStream("test.txt");
        ObjectOutputStream oos = new ObjectOutputStream(fos);
        Date date = new Date();
        oos.writeObject(date);
        oos.flush();
        oos.close();
        fos.close();
    }
}
```



Java documents

Many of us have been asking for Java records as a feature for a very long time. When you regretfully have to implement function toString() { [native code] }, hashCode, equals, as well as getters for each existing field, I assume you've seen this scenario numerous times. (I'm assuming you're not using setters anymore, and you really shouldn't.)

To address this issue, Kotlin offers data classes, while Java plans to do likewise with the release of the record classes, as does Scala with its case classes.

PROGRAM:

```
package com.theboreddev.java14;
import java.util.Objects;
public class Employee {
    private final String firstName;
    private final String surname;
    private final int age;
    private final Address address;
    private final double salary;

    public Employee(String firstName, String surname, int age, Address address, double salary) {
        this.firstName = firstName;
        this.surname = surname;
        this.age = age;
        this.address = address;
        this.salary = salary;
    }

    public String getFirstName() {
        return firstName;
    }

    public String getSurname() {
        return surname;
    }

    public int getAge() {
        return age;
    }

    public Address getAddress() {
        return address;
    }

    public double getSalary() {
        return salary;
    }

    public boolean equals(Object o) {
        if (this == o) return true;
        if (o == null || getClass() != o.getClass())
            return false;
        Employee employee = (Employee) o;
        return age == employee.age &&
            Double.compare(employee.salary, salary) == 0 &&
            Objects.equals(firstName, employee.firstName) &&
            Objects.equals(surname, employee.surname) &&
            Objects.equals(address, employee.address);
    }

    public int hashCode() {
        return Objects.hash(firstName, surname, age, address, salary);
    }

    public String toString() {
        return "Employee{" +
            "firstName=" + firstName + "\n" +
            ", surname=" + surname + "\n" +
            ", age=" + age +
            ", address=" + address +
            ", salary=" + salary + 65. '}'
    }
}
```

JAVA'S "KOTLINISATION"

The majority of these features are feature previews, so please keep that in mind. This implies that you won't be able to utilise JDK 14 or JDK 15 out of the box if you install them when they are published.

New features that are included in a release of Java but are deactivated by default are called feature previews. They are still vulnerable to modification because they are just included in the release in order to get feedback from the development community. It is not advised to use them in production code because of this.

You would need to take the following actions to activate them at compilation time: 14-year-old Javac —enable-preview

You would need to carry out the following in order to enable them at runtime

CONCLUSION:

The Java language has benefited greatly from competition in my opinion. Otherwise, I feel like Java would have sat back and enjoyed its success. Additionally, Java's rivals have demonstrated that it was possible to programme differently, illuminating the path forward rather than continuing with an antiquated and rusty approach.

“DO IT LEARN IT” ACTIVITY

Troop Selection

Problem Description:

- Commander-in-chief of Indian Army Wing-I wants to select jawans for west punjab troop. To ensure reliable security in west punjab Commander-in-chief needs P level of strength in westpunjab.
- Every Jawan is represented by a Decimal code di and his strength level is number of set bits(1) in the decimal codedi.
- Now Commander-in-chief wants to know the strength level of his troop consists of N Jawans in West punjab.

Sample Input

Sample Input:

10 3

5 6 7

Sample Output:

week

Explanation

Sample Input:

Strength of decimal code 5,6 and 7 are 2, 2 and 3 respectively. Total Strength is 7, required is 10.

So the output is "week".

For Solution Refer Page No:18

Strange Counter

Problem Description:

- There is a strange counter. At the first second, it displays the number 3. Each second, the number displayed by decrements by 1 until it reaches 1. In next second, the timer resets to 2 X the initial number for the prior cycle and continues counting down. The diagram below shows the counter values for each time t in the first three cycles:

time	value	time	value	time	value
1	3	4	6	10	12
2	2	5	5	11	11
3	1	6	4	12	10
		7	3	13	9
		8	2	14	8
		9	1	15	7
			
				21	1

Find and print the value displayed by the counter at time?

Sample Input :

4

Sample Output :

6

Explanation :

Time t = 4 marks the beginning of the second cycle. It is double the number displayed at the beginning of the first cycle: $2 \times 3 = 6$. This is shown in the diagram in the problem statement.

For Solution Refer Page No:19

Star Pattern

Problem Description:

- Commander-in-chief of Indian Army wants to give star shield for soldiers for displaying extreme act of bravery during war time. The star size is based upon the rank level of the soldier.

Sample Input:

4

Sample Output:

```
*
* * * *
* * *
* * * *
*
```

Explanation:

Two triangles are merged to form star. Side of the triangle is 4

For Solution Refer Page No:20

Indian Flag

Problem Description:

- Karpagam Institute of Technology is going to celebrate 70th Independence day in a grand manner. Students need flag in various sizes for the celebration so they approached "KIT CODERS CLUB" to design and print flags. Our flag has three stripes of equal width, the top being saffron (kesari), followed by white and then green at the bottom. There are two parts, part one represents stripes and part two represents stand.
- Part I: Use 'S' letter for first stripe, 'W' letter for Second stripe, 'G' letter for Third stripe.
- Part II: Use two vertical bar(|) to represent stand. Stand width is depends on the size of flag width.

Sample Input:

3

Sample Output:

SSSSs
WWWWw
GGGGg

||
||
||

Explanation:

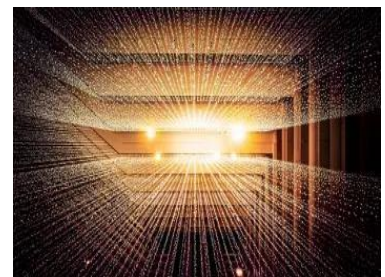
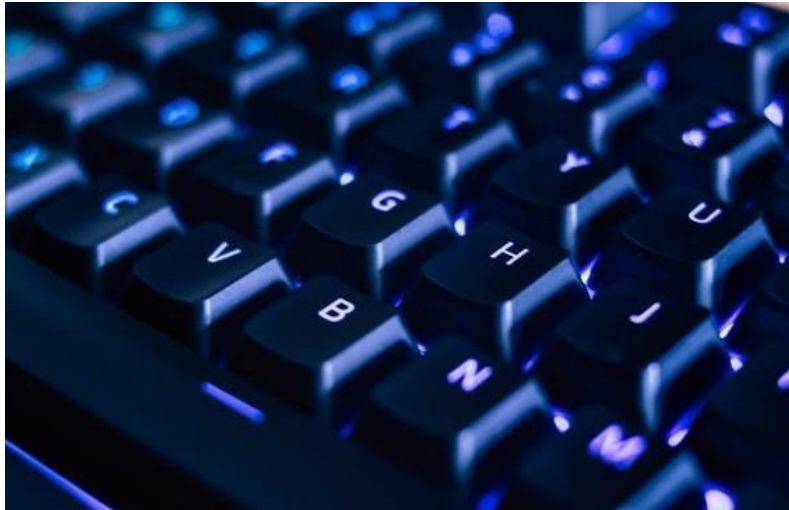
Sample Input:

- Width is 3 and Length is 4.5 so the .5 represented as small letter in 5th column.
- **Hint:** ceil() function in C returns nearest integer value which is greater than or equal to the argument passed to this function. "math.h" header file supports ceil()

For Solution Refer Page No:23

Creating Parquet Files in Java

Apache created the open source Parquet file format for the Hadoop architecture. It began as a file format for Hadoop, but it has now gained enormous popularity, and cloud service providers like AWS have begun to support it. This could only imply that Parquet is performing appropriately. In this article, we'll examine the details of the Parquet file format before looking at a straightforward Java sample that generates or writes Parquet files.



INTRODUCTION:

In the conventional method, we store data as rows. However, Parquet adopts a different strategy, flattening the data into columns prior to storing it. Better data compressions for storage and query performance are both made possible as a result.

THE DEPENDENCIES:

We must take care of the dependencies before we begin developing the code. We'll list all of our dependencies in the pom.xml file because this is a Spring Boot Maven project.

THE PROPERTIES:

We have an application, as usual. where we list all the properties in a properties file. schema.filePath= output.directoryPath=

SCHEMA OF THE PARQUET FILES:

The data schema for the Parquet file that we're going to write needs to be specified. This is so that the data type of each column is kept when a Parquet binary file is produced. The function will format the data in accordance with the schema we specify in a schema file before writing it to the Parquet file.

PREPARING THE DATA FOR PARQUET :

Prepare some data so that it may be written to the Parquet files. For the Parquet file, one data set is represented by a list of strings. Each entry in this list corresponds to the value of the schema file's correcting field.

GETTING THE PARQUET WRITER :

The process is virtually done with this phase. The Custom Parquet Writer class that we previously mentioned just requires that we obtain an instance. We also offer the output file's path here for the writer to use while writing.

THE PARQUET WRITER :

These two classes were on StackOverFlow while I was exploring this a few months ago. I have been utilising these two classes everywhere, albeit I have no idea who wrote this. But I have changed the class names to better fit the project.

CONCLUSION:

Due to all these features, the Parquet file format is used in the majority of large data projects. The quantity of storage space needed is also decreased by parquet files. We typically utilise queries with specific columns. Because each column's data is close to one another in the file format, queries execute more quickly.

JAVA



INTRODUCTION

Even in 2020, Go, Python, and Typescript will be fierce competitors for Java's position as one of the most widely used programming languages for creating online applications.

The Spring Framework has emerged as the de-facto standard for micro-service development within the Java community. So the development process and resource management of the created micro-services will be my primary concerns.

Spring (and most of the Java Platform, actually) haven't exactly had the finest reputation when it comes to resource management, especially when it comes to the overhead needed by a single process. This wasn't a huge issue back when application servers were still in use because there weren't many instances. However, as micro-service designs proliferate and have an enormous number of little instances, this becomes an increasing problem.

HELIDON MICROPROFILE :

When it was uncertain whether and how Oracle would continue working on Java Enterprise, the MicroProfile project was formed in 2016. MicroProfile is a specification that can be implemented by different vendors, just as its predecessor JEE. Since then, several of these implementations have been made public, most notably Payara Micro and Helidon MP. Payara Micro is the MicroProfile implementation of Payara, a Jakarta EE server developed from GlassFish. In 2018, Oracle launched the Helidon runtime, which offers its own implementation of the MicroProfile specification. Only Helidon MP, with an average time of 8.27 seconds, was slower than Spring.

COMPARING THE FRAMEWORKS:

On three crucial phases, I've compared how well the applications performed: How simple was it to put the example application into

THE SPRING:

In 2003, Spring was created in response to the early Java Enterprise's complexity. A framework for aspect-oriented programming (AOP) with dependency injection (DI) at its core, Spring has developed into a simple web application framework.

Spring uses reflection to carry out DI at runtime. As a result, the classpath is searched for annotated classes when a spring application is launched. The concrete items are created and linked on the basis of this.

The average boot time for the Spring Boot application using Spring Data was 8.16 seconds. This was cut to just 5.8 seconds by removing JPA and Spring Data.

QUARKS :

At 197 MB, Quarkus performed far better. With 414 MB, Helidon MP was comparable to Spring Boot.

In 2019, Red Hat released the Java framework Quarkus, which is native to Kubernetes. Standards like Micro Profile, Vert.x, Netty, and Hibernate are built on top of it. Quarkus aims to elevate Java to the position of a dominant platform in Kubernetes by enabling platforms for container-orchestration to scale up almost instantly, with a near- instantaneous startingtime.

The technology that Quarkus uses are primarily already established standards, however it is flexible. Since the project was only begun a year ago, it is often unclear how mature and compatible certain extensions are. Future changes are probably in store as the platform develops. Lower starting times were delivered by quarks (5.7 seconds).

practise? I had to read the documentation and conduct research on websites like Stack Overflow in order to apply the frameworks. How long does the application take to compile? I've timed how long a clean build takes, including the creation of the Docker image. This includes the time it takes to create the native image for GraalVM. How long does it take an application to launch? Here, I've timed how long it takes from the moment Docker is started until the application correctly responds to the first HTTP request. I also compared the idle applications' measured memory footprints immediately after boot.

LOAD:

How many requests can the application handle at peak? I used JMeter to perform load testing and tested the application with 25% of the requests performing a database write and 75% of them performing only a database read. I then measured the memory footprint of the application on its peak performance again.

CONCLUSION

When compared to current frameworks like Spring and MicroProfile, the new Java frameworks Micronaut and Quarkus claimed quicker startup times and smaller memory footprints.

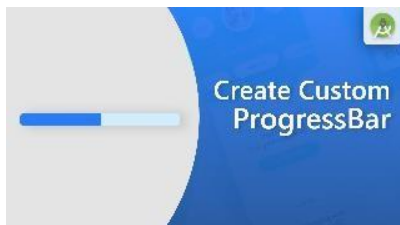
They do follow through on this promise, but only while the engine is idling or lightly loaded. They outperform Spring in this situation, especially when used in conjunction with native GraalVM images. Even when running as native images, they don't provide much of an advantage while under pressure. And even with its subpar startup performance, Spring continues to provide, by far, the best developer experience, making it, in my opinion, the ideal Java framework for a microservice application.



INTRODUCTION

Circular progress indicator bar display android progress by animating an indicator along an invisible circular track in a clockwise direction. They can be applied directly to a surface, such as a button or card. Circular progress indicators support both determinate and indeterminate processes.

- Determinate circular indicators fill the invisible, circular track with color, as the indicator moves from 0 to 360 degrees.
- Indeterminate circular indicators grow and shrink in size while moving along the invisible track.

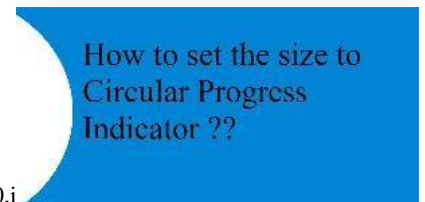


DRAWING

This section regulates how the view is rendered on the screen. The `onDraw()` method can be overridden to allow us to represent the entire visual portion of the view. In this technique, the visual shape of the view, such as an instance custom view, is painted using a Canvas object. For instance, using this technique, you can create a circle, line, or arc, which serve as the visual components of your own custom view. It should be emphasised that this approach evaluates all views equally, custom or built-in views included.

PROGRAM:

```
package com.example.mycustomview;
import android.*;
public class ProgressBar extends View {
    private int backColor=0,
                                int frontColor=0,i
    ,boolean displayValue;
    private int
    outerCircleRadius=200;
    private int
    innerCircleRadius=150;
    private int
    displayTextSize=20; private
    int value=90;
    private int
    centerX=200; private
    int centerY=200;
    private int width=400;
    private int height=400;
    private String
    name="";
    public ProgressBar(Context context, AttributeSet
                                attrs) {
        super(context, attrs);
        setupAttrs(attrs);
    }
    private void setupAttrs(AttributeSet attrs){
```



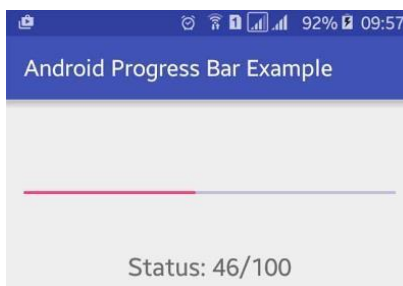
ATTRIBUTING

There are undoubtedly a number of style and format attributes in a custom view that need to be correctly implemented. These XML characteristics are chosen to best match the setups of the custom views. Measuring. The breadth and height of the view are dealt with by measurement. The `onMeasure()` method, which should be overridden, evaluates this component. This approach bases a view's width and height determination on the constraints placed on the parent and the contents of the view. The measured specific width and height are

INTERACTIONS:

Standard views respond to preprogrammed events like touch, click, and other similar ones. The custom view can be written to behave appropriately with each define event, with some of its attributes potentially changing.

For this reason, the main activity defines one instant of the custom view, and its event listeners are developed to carry out some particular activities.



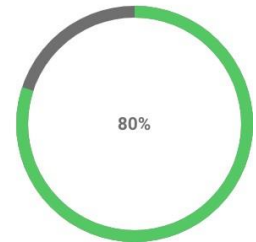
```
try {
    for(int i=0;i<attrs.getAttributeCount();i++) {
        if(attrs.getAttributeName(i).contains("backColor")
        )) {
            backColor = attrs.getAttributeIntValue(i,
            backColor);
        }else
        if(attrs.getAttributeName(i).contains("frontColor")
        )){ frontColor = attrs.getAttributeIntValue(i,
        frontColor);
        }else
        if(attrs.getAttributeName(i).contains("lineColor")
        )){ lineColor = attrs.getAttributeIntValue(i,
        lineColor);
        }else
        if(attrs.getAttributeName(i).contains("displayVal
        ue")){ displayValue =
        attrs.getAttributeBooleanValue(i, false);
        }else
        if(attrs.getAttributeName(i).contains("name"))
        { name = attrs.getAttributeValue(i);
        }else
        if(attrs.getAttributeName(i).contains("layout_wi
        dth")) { String text=attrs.getAttributeValue(i);
        text=text.substring(0,text.length()-5);
        width = Integer.parseInt(text);
        }else
        if(attrs.getAttributeName(i).contains("layout_hei
        ght")) { String text=attrs.getAttributeValue(i);
        text=text.substring(0,text.length()-5);
        height = Integer.parseInt(text);
        }
        }
    }catch(Exception ex)
    {
        Toast.makeText(getContext(),ex.getMessage(),
        Toast.LENGTH_SHORT).show();
    }
}
```



the two integer input arguments for this procedure.



★ Progress Bar



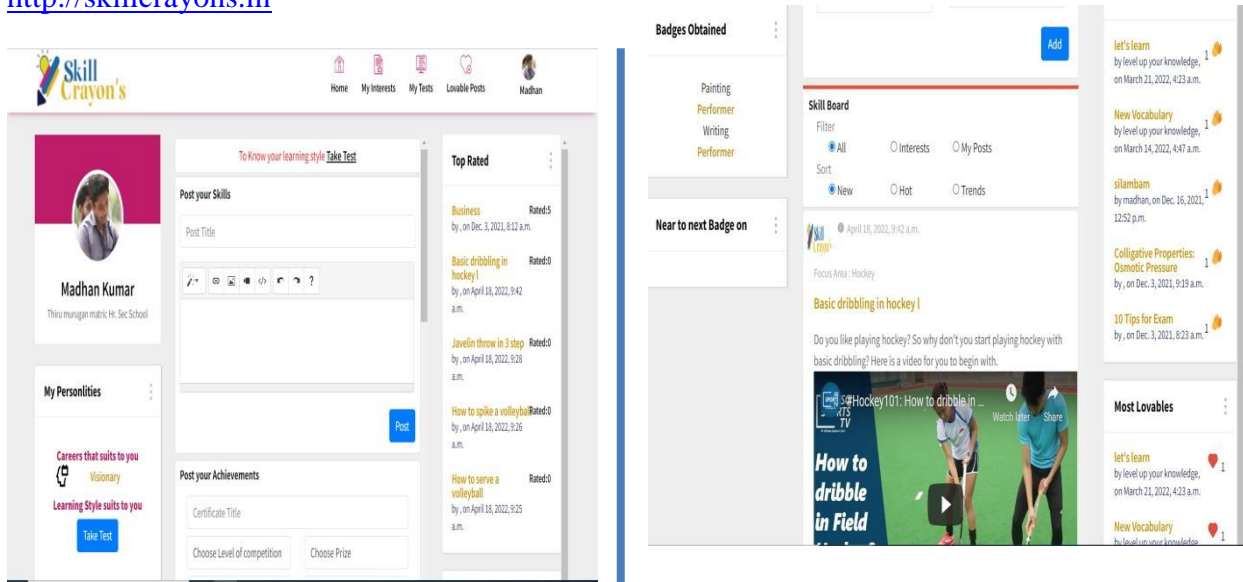
CONCLUSION:

Thus Circular Progress indicators express an unspecified wait time or display the length of a process. Material design provides an implementation of linear and circular progress indicators, compatible back to API 15. Progress Indicators is API-compatible with Android's progressBar class and can therefore be used as a replacement for the progress Bar. Progress Indicators were added to the material design in 1.3.0. Now it's in alpha 02. So you need to include the following dependency to work with the progress indicator.

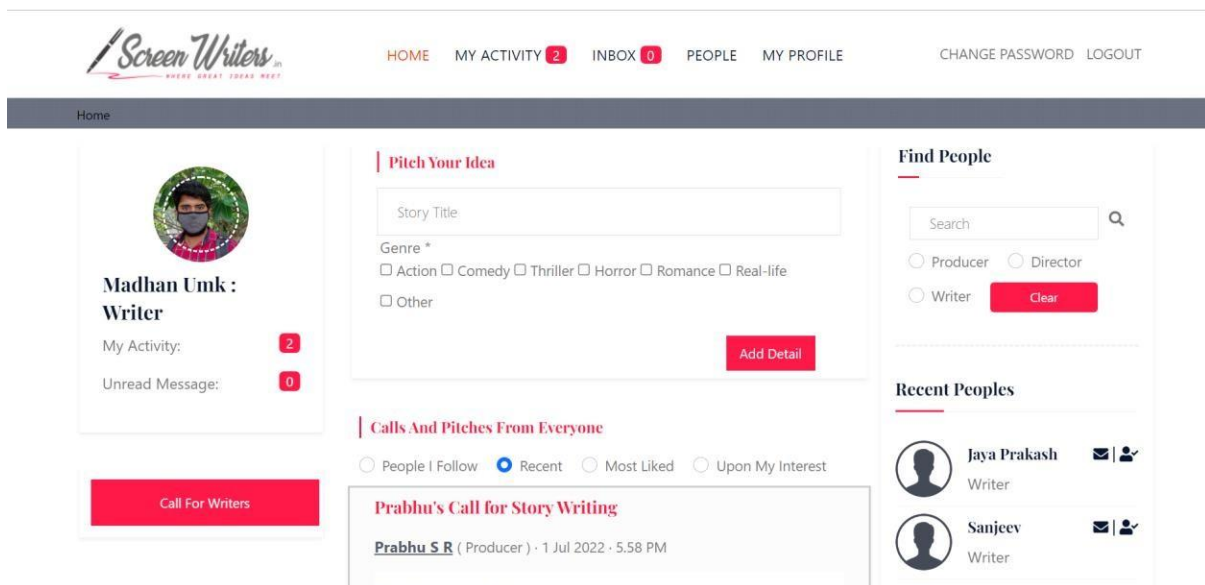
PROJECT GLIMPSES

SKILL CRAYONS

Skill Crayons is a social networking portal designed and developed for students to share their skills and skill related activities. Portal motivates the students by rewarding points and badges for their activities. The Portal also includes Quiz Platform to assess student knowledge in specific skill sets. The Portal is hosted in the domain <http://skillcrayons.in>



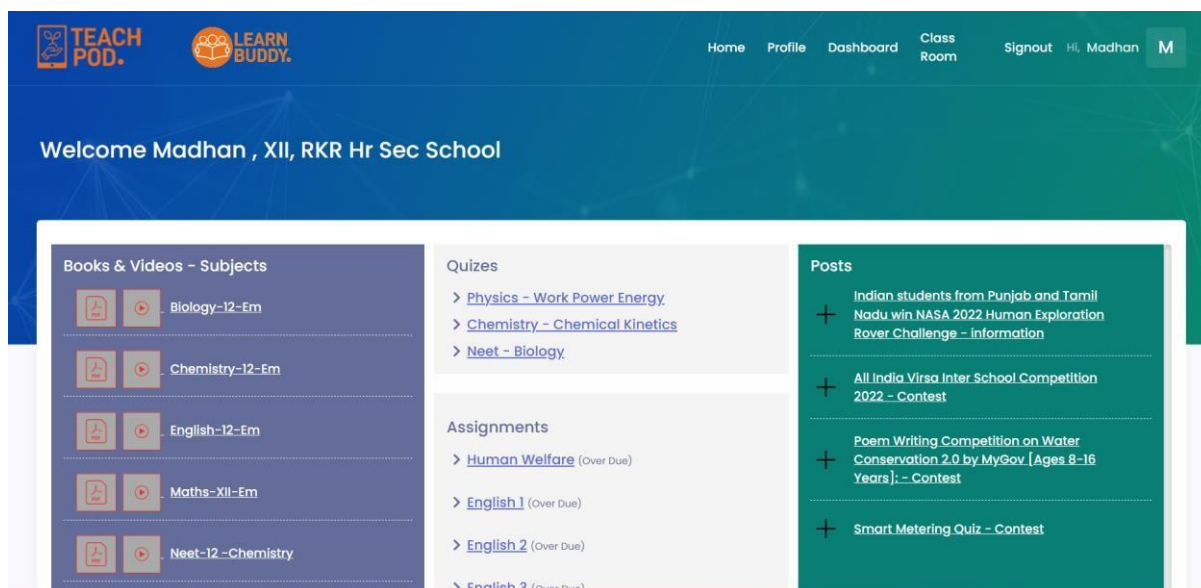
SCREEN WRITERS



Screen Writers is a networking portal designed and developed for Writers, Directors and Producers. Screen Writers enables producers and directors to search script writers. Writers can submit their scripts to producers and directors. This Portal is developed using Django, HTML, CSS and JQuery. The Portal has other functionalities like following,

communicating, and profile building. The Portal is developed under industrial project development. The Portal is hosted in <https://screenwriters.in>

TEACHPOD



Teachpod is a Digital Learning system to help Children, Teachers and Schools. It covers the complete Teaching-Learning Process for Class 1 to 12. Teachpod supports is suitable for both CBSE and State board Syllabus. Teachpod has Smart E-book reader, Self-paced and personalized teaching videos, Assessments, discussion forums, Personality tests, circular management, classrooms, and more. The portal is hosted in the domain <https://teachpod.in>

„DO IT LEARNIT“-SOLUTIONS:

Troop Selection

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int p, a[5], n, i, s, r;
    int count;
    scanf("%d %d", & p, & n);
    for (i = 0; i < n; i++) {
        scanf("%d", & a[i]);
    }

    for (i = 0; i < n; i++) {
        s = a[i];
        while (s != 0) {
            r = s % 2;
            if (r == 1) {
                count++;
            }
            s = s / 2;
        }

        if (count < p) {
            printf("week");
        } else if (count == p) {
            printf("enough");
        }

        if (count > p) {
            printf("strong");
        }

        return 0;
    }
}
```

Strange Counter

Solution:

```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
    import java.util.regex.*;
public class StrangeCounter {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        long t = in.nextLong();
        long d = 3;
        long ans = 0;
        while(2>1){
                                if(t<=d){
                                    ans = d+1-t;
                                    break;
                                }
                                else{
                                    t-=d;
                                    d =2*d;
                                }
                                }
        System.out.println(ans);
    }
}
```

Star Pattern

Solution:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {
    int n,i,j;
    scanf("%d",&n);
    if(n%2 == 0){
        for(i = 1 ; i <=n-3 ;++i){
            if(i%2!= 0) {
                for(j = 1 ; j <= (2*n)-1 ; ++j) {
                    if(j >= n-(i-1) && j < n+i ){
                        if(j%2==0)
                            printf("*");
                        else
                            printf(" ");
                    }
                    else
                        printf(" ");
                }
                else{
                    for(j = 1 ; j <= (2*n)-1 ; ++j) {
                        if(j >= n-(i-1) && j < n+i ){
                            if(j%2!=0)
                                printf("*");
                            else
                                printf(" ");
                        }
                        else
                            printf(" ");
                    }
                    printf("\n");
                }
            }
            else{
                for(i = 1 ; i <=n-3 ;++i){
                    if(i%2== 0) {
                        for(j = 1 ; j <= (2*n)-1 ; ++j) {
                            if(j >= n-(i-1) && j < n+i ){
                                if(j%2==0)
                                    printf("*");
                                else
                                    printf(" ");
                            }
                        }
                    }
                }
            }
        }
    }
```

```

        printf(" ");
        }
        else
        printf(" ");
        }
        }
        else{
            for(j = 1 ; j <= (2*n)-1 ; ++j) {
if(j >= n-(i-1) && j < n+i ){
            if(j%2!=0)
                printf("*");
            else
                printf(" ");
            }
            else
                printf(" ");
            }
        }
        printf("\n");
        }
        }
        for(i = 1 ; i <= 3 ; ++i) {
for(j = 1 ; j <= (2 * n)-1 ; ++j){
            if(i == 1 || i == 3){
if(j % 2 !=0)
                printf("*");

            else
                printf(" ");
            }
            else
                printf(" ");
            }
            if(j % 2 == 0)
                printf("*");

            else
                printf(" ");
            }
            printf("\n");
        }
        if(n%2 == 0){
for(i = n-3 ; i >=1 ;--i){
            if(i%2!= 0) {
                for(j = 1 ; j <= (2*n)-1 ; ++j) {
if(j >= n-(i-1) && j < n+i ){
                    if(j%2==0)
                        printf("*");
                    else
                        printf(" ");
                    }
                    }
                }
            }
        }
    }

```



```

                                for(j = 1 ; j <= (2*n)-1 ; ++j) {
if(j >= n-(i-1) && j < n+i ){
    if(j%2!=0)
        printf("*");
    else
        printf(" ");
    }
    else
        printf(" ");
    }
    printf("\n");
}
else{
for(i = n-3 ; i >= 1 ; --i){
    if(i%2== 0) {
        for(j = 1 ; j <= (2*n)-1 ; ++j) {
if(j >= n-(i-1) && j < n+i ){
    if(j%2==0)
        printf("*");
    else
        printf(" ");
    }
    else
        printf(" ");
    }
    else{
        for(j = 1 ; j <= (2*n)-1 ; ++j) {
if(j >= n-(i-1) && j < n+i ){
    if(j%2!=0)
        printf("*");
    else
        printf(" ");
    }
    else
        printf(" ");
    }
    }
    printf("\n");
}
}
return 0;
}

```

Indian Flag - Pattern

Solution:

```
#include <stdio.h>
#include
<string.h>#include
<math.h> #include
<stdlib.h> int
main(){
    int w;
    float b,b1;
    scanf("%d", & w);
    int w1 = w / 3;

    int , i, j;

    b = w * 1.5;
    int l1 = b;
    for (i = 0; i < w1; i++) {
        for (j = 0; j < b - 1; j++) {

            printf("S");
            }
        if (w % 2 == 0) {
            printf("S");

            }
        printf("\n");
        }

    for (i = 0; i < w1; i++) {
        for (j = 0; j < b - 1; j++) {

            printf("W");
            }
        if (w % 2 == 0) {
            printf("W");

            }
        printf("\n");
        }

    for (i = 0; i < w1; i++) {
        for (j = 0; j < b - 1; j++) {

            printf("G");
            }
        if (w % 2 == 0) {
            printf("G");

            }
        printf("\n");
        }

    for (i = 0; i < w; i++) {
        printf("||");
    }
```

2020



Editorial Faculty Member

Mr.P.N.Ramesh

Editorial Student Members

Mr.Joel Johnson.S (IV-CSE)

Ms.Divya.B (IV-CSE)

Mr.Vikash.v (III-CSE)

Ms.Pradeepa.S (III-CSE)

Mr.Sushvin.M (II-CSE)

