Delpech Nicolas
Etudiant Informatique
Groupe A1

13/12/2024

Calendrier

Delpech Nicolas
Etudiant Informatique
Groupe A1

Introduction

Dans ce document vous aller découvrir des testes pour analyser la performance et la structure de l'exercice qu'on a fait pour le DS de la SAE12 qui consiste de faire une class Calendrier qui prends un nombre de jour pour ensuite le convertir en nombre de mois et en nombre d'année.

Exercice n°1

méthode (à part test)

Dans ma classe Calendrier à 3 attribut qui permet de mettre un nombre de jours , de mois et d'année ce constructeur possèdes des vérification pour répondre au besoins :
« Nous pouvons stocker la date avec 3 entiers correspondants aux numéro du jour du mois (de 1 à 31), du
numéro du mois (de 1 à 12) et de l'année (exemples 1900, . . ., 2024). »

Une fois que les vérifications sont mise nous pouvons créer notre date avec 3 entier.

```
public class Calendrier
{
	private int jour;
	private int mois;
	private int annee;

	public Calendrier(int jour, int mois, int annee)
	{
		/*----------------------*/
		/*	Vérification	*/
		/*----------------------*/
		if(annee < 1900)
		{
			this.annee = 1900;
		}
		if(annee > 2024)
		{
			this.annee = 2024;
		}

		if(mois < 1)
		{
			this.mois = 1;
		}
		if(mois > 12)
		{
			this.mois = 12;
		}

		if(jour < 1)
		{
			this.jour = 1;
		}
		if(jour < 31)
		{
			this.jour = 31;
		}


		this.jour = jour;
		this.mois = mois;
		this.annee = annee;
```

```
	}

	public String toString()
	{
		return this.jour + "/" + this.mois + "/" + this.annee;
	}
```

J’ai fait un fichier TestCalendrier.java qui permet de faire mes testes pour mon calendrier en temps normale on vois que juste pour mon constructeur nous avons mit pour cette date 1/1/1900 :

Temps d'exécution: 1 299 934 nano moyenne

Maintenant je vais faire un autre teste avec le constructeur pour voir si le temps change et si il sera meilleur pour si on enlève les vérifications :

```
public class Calendrier
{
	private int jour;
	private int mois;
	private int annee;

	public Calendrier(int jour, int mois, int annee)
	{
		this.jour = jour;
		this.mois = mois;
		this.annee = annee;
	}

	public String toString()
	{
		return this.jour + "/" + this.mois + "/" + this.annee;
	}
}
```

Après le Test j’ai pu avoir comme résultat : Temps d'exécution: 1 367 995 nano moyenne

Nous pouvons en déduire que les vérification pour se constructeur ne se joue pas à grand-chose.

Exercice 1 (méthode demander)
Mais cette fois nous allons faire un autre constructeur qui prends que les jours.
Ce constructeur va permettre d’afficher la note en fonction en nombre de jour exemple :
— le 1er jour correspond au 1/1/1900
— le 32ème jour correspond au 1/2/1900
— le 60ème jour correspond au 1/3/1900
— le 366ème jour correspond au 1/1/1901
— le 1 000ème jour correspond au 27/9/1902

— le 36 525ème jour correspond au 1/1/2000
— le 45 641ème jour correspond au 16/12/2024

Malheuresement mon code n’est pas terminer et i n’est pas parfait il y a des erreurs. Cependant il y a le même principe.

Présentation du code de la classe Calendrier

```
public class Calendrier
{
	private int jour = 1;
	private int mois = 1;
	private int annee = 1900;

	public Calendrier(int jour)
	{
		int cptMois;
		int[] tabJour;

		tabJour = Calendrier.tabJour(this.annee);

		cptMois = 1;
		for(int cptJour=2; cptJour <= jour ; cptJour++)
		{
			this.jour ++;

			if(this.jour == tabJour[cptMois])
			{
				this.jour = 1;
				cptMois++;
				this.mois  = cptMois;
			}

			if(cptMois == 12)
			{
				this.annee = this.annee + 1;
				this.jour = 1;
				cptMois = 1;
				this.mois = cptMois;

				tabJour = Calendrier.tabJour(this.annee);
			}
		}
	}

	public static boolean estBissextiles(int annee)
	{
		if(annee % 4 == 0 && (annee / 100 == 1 || annee % 400 ==0 ))
		{
			return true;
		}
		return false;
	}

	public static int dureeAnne(int annee)
	{
		if(Calendrier.estBissextiles(annee))
		{
			return 366;
		}
		return 365;
	}

	public static int dureeMois(int annee, int mois)
```

```
	{
		int[] tabMois;
		int[] tabJour;

		tabJour = new int[12];
		tabMois = new int[12];

		for(int cptmois=1; cptmois <= tabMois.length; cptmois++)
		{
			tabMois[cptmois] = cptmois;
			if(cptmois % 3 == 1)
			{
				tabJour[cptmois] = 31;
			}
			else
			{
				tabJour[cptmois] = 30;
			}
		}

		return 0;

	}

	public static int[] tabJour(int annee)
	{
		int[] tabJour;
		int[] tabMois;

		tabJour = new int[13];
		tabMois = new int[13];

		for(int cptmois=1; cptmois <= 12 ; cptmois++)
		{
			tabMois[cptmois] = cptmois;

			if(cptmois % 3 == 1 || cptmois == 1)
			{
				tabJour[cptmois] = 31;
			}
			else
			{
				if(tabMois[cptmois] == 2)
				{
					if(Calendrier.estBissextiles(annee))
					{
						tabJour[cptmois] = 29;
					}
					else
					{
						tabJour[cptmois] = 28;
					}
				}
				else
				{
					tabJour[cptmois] = 30;
				}
			}
		}

		return tabJour;
	}

	public String toString()
	{
		return "\n " + this.jour + "/" + this.mois + "/" + this.annee;
	}
}
```

J’ai fait un test pour voir se que ça donnais et je vois ceci :

1/1/1900
Temps d'exécution: 468875 nano

2/2/1900
Temps d'exécution: 3795 nano

3/3/1900
Temps d'exécution: 3482 nano

15/2/1901
Temps d'exécution: 11098 nano

1/1/1900
Temps d'exécution: 1 414 731 nano

2/2/1900
Temps d'exécution: 4009 nano

3/3/1900
Temps d'exécution: 3549 nano

15/2/1901
Temps d'exécution: 25636 nano

On remarque le premier teste et le deuxième sont quasiment presque pareil au niveau de temps d’exécution en nanoseconde

Exercice 2

Maintenant dans notre class Calendrier on va tester nos méthodes :

— public static boolean estBissextiles(int annee)
retourne true si l’année est bissextile, false sinon.

— public static int dureeAnnee(int annee)
retourne le nombre de jours de l’année.

— public static int dureeMois(int annee, int mois)
retourne le nombre de jours du mois selon l’année.

— public static int[] jour2Date(int nbJour)

Pour la Méthode bissextile

```
public static boolean estBissextiles(int annee)
	{
		if(annee % 4 == 0 && (annee / 100 == 1 || annee % 400 ==0 ))
		{
			return true;
		}
		return false;
	}
```

Dans cette méthode on vérifie si l'année est divisible par 4 et non par 10 ou par 400 pour savoir s'il y a 366 jours ou bien 365 jours dans une année.

Test :
1/1/1900
false
Temps d'exécution: 1605 nano

1/1/1900
false
Temps d'exécution: 1028 nano

On voit le temps d'exécution pour cette méthode est vraiment très courte en moyenne car cette méthodes contient juste une vérifications et quelque calcule qui prends du temps dans cette méthode.

Pour la Méthode dureeAnnee

```
public static int dureeAnne(int annee)
	{
		if(Calendrier.estBissextiles(annee))
		{
			return 366;
		}
		return 365;
	}
```

Cette méthode contient juste une vérification pour savoir si l'année est bissextile ou pas si oui alors nous avons 366 jours dans l'année sinon ça veut dire que dans une année nous avons 365 jours

Pour la Méthode dureeMois

```
public static int dureeMois(int annee, int mois)
	{
		int[] tabJour;
		tabJour = Calendrier.tabJour(annee);

		return tabJour[mois];
	}
```

Cette méthode permet de créer un tableau de jour qui correspond à l'année et ensuite il retourne le nombre de jours du mois qu'il a choisie.

Test :

```
debut = System.nanoTime();
nbJourMois = Calendrier.dureeMois(1900,1);
fin = System.nanoTime();
System.out.println(nbJourMois);
System.out.println("Temps d'exécution: " + (fin - debut) + " nano");
```

```
31
Temps d'exécution: 3138 nano
```

```
debut = System.nanoTime();
nbJourMois = Calendrier.dureeMois(2004,2);
fin = System.nanoTime();
System.out.println(nbJourMois);
System.out.println("Temps d'exécution: " + (fin - debut) + " nano");
```

```
28
Temps d'exécution: 4265 nano
```

On a tester cette méthode pour vérifier que ça marche bien et on peut en déduire avec quelque teste que les vérification pour le mois de Février sont un peux lourd car il y a des vérifications pour savoir si on est dans une année bissextile ou pas.

Conclusion

On peut en conclure que l’exercice 2 est plus avantageux et plus rapide que celui de l’exercice 1 en temps exécution car les méthodes met moins de temps que tous les vérification plus des méthode qui utilise le constructeur pour faire la date.