“WS_Agent.java”

```
import java.util.ArrayList;

public class WS_Agent {
	private ArrayList<String> list;
	private String regex;
	private int letters = 0;

	public WS_Agent () {
		list = new ArrayList<String>();
	}

	protected void SetRegex (String s) {regex = s;}
	protected void SetLetters (int i) {letters = i;}
	protected int GetLetters () {return letters;}
	protected ArrayList<String> GetList () {return list;}

	// option - 1= débutant par
	// 2= terminant par
	// 3= avec joker (longueur fixée)
	// note : regex doit être connu
	protected void FilterList (int option) {
		ArrayList<String> filtered = new ArrayList<String>();

		switch(option) {
			case 1 :  for(String s : list)
						if(VerifyWord(s, 0)) AddWord(s, filtered);
				  break;
			case 2 :  for(String s : list)
						if(VerifyWord(s, 1)) AddWord(s, filtered);
				  break;
			case 3 :  FilterWithJoker();
				  break;
			default : break;
		}
		if(option == 1 || option == 2) {
			list.clear();
			for(String s : filtered) AddWord(s, list);
		}
		RemoveDuplicates();
	}
```

```
private void FilterWithJoker () {
	int count;
	boolean ok = true;
	ArrayList<String> filtered = new ArrayList<String>();

	for(String s : list) {
		count = 0;
		while(count < regex.length() && ok == true) {
			if(regex.charAt(count) == '.') {count++;continue;}
			if(regex.toLowerCase().charAt(count) == s.toLowerCase().charAt(count)) ok = true;
			else ok = false;
			count++;
		}
		if(ok) AddWord(s, filtered);
		else ok = true;
	}
	// ici filtered est ok
	list.clear();
	for(String s : filtered) AddWord(s, list);
}

// option - 0: débute par
// 1: termine par
private boolean VerifyWord (String s, int option) {
	boolean ok = false;
	switch(option) {
		case 0:	if(s.regionMatches(true, 0, regex, 0, regex.length())) ok = true;
			else ok = false;
			break;
		case 1:	if(s.regionMatches(true, (s.length()-regex.length()), regex, 0, regex.length())) ok = true;
			else ok = false;
			break;
	}
	return ok;
}

protected void AddWord (String s, ArrayList<String> l) {
	l.add(s);
}

protected void DisplayList () {
	for(String s : list) System.out.println(s);
}
```

```
    private void RemoveDuplicates () {
        ArrayList<String> temp = new ArrayList<String>();
        for(String s : list)
            if(!Exists(s, temp)) AddWord(s, temp);
        list.clear();
        for(String s : temp) AddWord(s, list);
    }

    private boolean Exists (String s, ArrayList<String> l) {
        for(String d : l)
            if(d.equals(s)) return true;
        return false;
    }
}
```

“WS_Library.java”

```
import java.io.*;

public class WS_Library {
	public WS_Library () {}
	protected String[] BuildWordListFromFile (int letters) {
		String text = "";
		if(letters == 2) text = MergeTexts(text, "./lib/WS_2.txt");
		else if(letters == 3) text = MergeTexts(text, "./lib/WS_3.txt");
		else if(letters == 4) text = MergeTexts(text, "./lib/WS_4.txt");
		else if(letters == 5) text = MergeTexts(text, "./lib/WS_5.txt");
		else if(letters == 6) text = MergeTexts(text, "./lib/WS_6.txt");
		else if(letters == 7) {
			text = MergeTexts(text, "./lib/WS_7A.txt");
			text += MergeTexts(text, "./lib/WS_7B.txt");
		}
		else if(letters == 8) {
			text = MergeTexts(text, "./lib/WS_8A.txt");
			text += MergeTexts(text, "./lib/WS_8B.txt");
			text += MergeTexts(text, "./lib/WS_8C.txt");
		}
		else if(letters == 9) {
			text = MergeTexts(text, "./lib/WS_9A.txt");
			text += MergeTexts(text, "./lib/WS_9B.txt");
			text += MergeTexts(text, "./lib/WS_9C.txt");
		}
		else if(letters == 10) {
			text = MergeTexts(text, "./lib/WS_10A.txt");
			text += MergeTexts(text, "./lib/WS_10B.txt");
			text += MergeTexts(text, "./lib/WS_10C.txt");
			text += MergeTexts(text, "./lib/WS_10D.txt");
		}
		else if(letters == 11) {
			text = MergeTexts(text, "./lib/WS_11A.txt");
			text += MergeTexts(text, "./lib/WS_11B.txt");
			text += MergeTexts(text, "./lib/WS_11C.txt");
			text += MergeTexts(text, "./lib/WS_11D.txt");
		}
		else if(letters == 12) {
			text = MergeTexts(text, "./lib/WS_12A.txt");
			text += MergeTexts(text, "./lib/WS_12B.txt");
			text += MergeTexts(text, "./lib/WS_12C.txt");
		}
		else if(letters == 13) {
			text = MergeTexts(text, "./lib/WS_13A.txt");
			text += MergeTexts(text, "./lib/WS_13B.txt");
```

```
            text += MergeTexts(text, "./lib/WS_13C.txt");
        }
        else if(letters == 14) {
            text = MergeTexts(text, "./lib/WS_14A.txt");
            text += MergeTexts(text, "./lib/WS_14B.txt");
        }
        else if(letters == 15) text = MergeTexts(text, "./lib/WS_15.txt");
        else if(letters == 16) text = MergeTexts(text, "./lib/WS_16.txt");
        else if(letters == 17) text = MergeTexts(text, "./lib/WS_17.txt");
        else if(letters == 18) text = MergeTexts(text, "./lib/WS_18.txt");
        else if(letters == 19) text = MergeTexts(text, "./lib/WS_19.txt");
        else if(letters == 20) text = MergeTexts(text, "./lib/WS_20.txt");
        else if(letters == 21) text = MergeTexts(text, "./lib/WS_21.txt");

        String[] list = text.split("\s");
        return list;
    }

    private String MergeTexts (String text, String filename) {
        try {
            String line = "";
            FileReader fr = new FileReader(filename);
            BufferedReader buf = new BufferedReader(fr);
            do{
                line = buf.readLine();
                if(line != null) text += line + "\s";
            } while(line != null);

            buf.close();
            fr.close();
        }
        catch(IOException e) {System.out.println(e.getMessage());}

        return text;
    }

    protected void ConvertToArrList (String[] list, WS_Agent spy) {
        for(int i = 0 ; i < list.length ; i++) spy.AddWord(list[i], spy.GetList());
    }
}
```

“WS_Setup.java”

```
import java.util.Scanner;

public class WS_Setup {
	private Scanner scan;
	private WS_Library lib;
	private String[] tab;

	public WS_Setup () {
		lib = new WS_Library();
		scan = new Scanner(System.in);
	}

	protected void SetRegex (WS_Agent spy) {
		String temp;
		System.out.print("Chercher (JOKER : . ) : ");
		temp = scan.nextLine();
		spy.SetRegex(temp);
	}

	protected void SetLetterCount (WS_Agent spy, int num) {
		spy.SetLetters(num);
	}

	// choice1 - 1= recherche par longueur de mot
	// 2= recherche tous les mots
	// choice2 - 1= débutant par
	// 2= terminant par
	// 3= avec joker (longueur fixée)
	protected void GetWordList (int choice1, int choice2, WS_Agent spy) {
		if(choice1 == 1) { // recherche par longueur de mot
			MakeUpList(spy);
			switch(choice2) {
				case 1:	spy.FilterList(1);
					break;
				case 2:	spy.FilterList(2);
					break;
				case 3:	spy.FilterList(3);
					break;
				default:	break;
			}
		}
		else if(choice1 == 2){ // recherche tous les mots
			MakeUpALL(spy);
			switch(choice2) {
				case 1:	spy.FilterList(1);
					break;
```

```
                case 2:   spy.FilterList(2);
                          break;
                default:  break;
            }
        }
    }

    private void MakeUpList (WS_Agent spy) {
        tab = lib.BuildWordListFromFile(spy.GetLetters());
        lib.ConvertToArrList(tab, spy);
    }

    private void MakeUpALL (WS_Agent spy) {
        tab = lib.BuildWordListFromFile(2);
        lib.ConvertToArrList(tab, spy);
        for(int i = 3 ; i < 22 ; i++) {
            tab = lib.BuildWordListFromFile(i);
            ComposeWordLists(tab, spy);
        }
    }

    protected void ComposeWordLists (String[] tab, WS_Agent spy) {
        for(int i = 0 ; i < tab.length ; i++) spy.AddWord(tab[i], spy.GetList());
    }

    protected String[] BuildStringArray (String s) {
        return s.split("\s");
    }
}
```

“WS.java”

```
import java.util.Scanner;

public final class WS {
	private static Scanner scan;
	private static WS_Setup ws;
	private static WS_Agent spy;

	public static void main (String[] args) {
		ws = new WS_Setup();
		scan = new Scanner(System.in);
		int choice1 = 0, choice2 = 0;
		spy = new WS_Agent();

		while(true) {
			// section collecte d'entrées utilisateur
			choice1 = MainMenu();
			if(choice1 == 1 || choice1 == 2) choice2 = SubMenu1();
			if(choice2 >= 1 && choice2 < 4) {
				switch(choice2) {
					case 1:	System.out.print("Mot débutant par : ");
						break;
					case 2:	System.out.print("Mot terminant par : ");
						break;
					// utilisé seulement pour "par longueur de mot"
					case 3:	System.out.print("Positions avec joker (.) : ");
						break;
					default:	System.out.println("Mauvais choix");
						break;
				}
				// input de la chaîne à rechercher
				ws.SetRegex(spy);
			}
			// choice1 - 1= recherche par longueur de mot
			// 2= recherche tous les mots
			// choice2 - 1= débutant par
			// 2= terminant par
			// 3= avec joker (longueur fixée)
			// section recherche selon options choisies
			if(choice1 == 1) {
				ws.GetWordList(choice1, choice2, spy);
				System.out.println();
				spy.DisplayList();
				System.out.println(spy.GetList().size() + " mots trouvés\n");
			}
```

```
            else if(choice1 == 2) {
                ws.GetWordList(choice1, choice2, spy);
                spy.DisplayList();
            }
        }
    }

    private static int MainMenu () {
        // effacer écran
        System.out.println("1. Rechercher par longueur de mot");
        System.out.println("2. Rechercher tous les mots");
        System.out.println("3. Quitter");
        System.out.print("\t--> Choix : ");
        int choice;
        choice = scan.nextInt();
        scan.nextLine();

        if(choice == 1) {
            System.out.print("Nombre de lettres : ");
            int i = scan.nextInt();
            scan.nextLine();
            System.out.println("\nletters --> " + i);
            ws.SetLetterCount(spy, i);
        }
        else if(choice == 3) System.exit(0);
        return choice;
    }

    private static int SubMenu1 () {
        // effacer écran
        System.out.println("1. Mots débutant par [ ]");
        System.out.println("2. Mots terminant par [ ]");
        System.out.println("3. Positions variées (joker : . (point)");
        System.out.println("4. Menu principal\n\n");
        System.out.print("\t--> Choix : ");

        int choice = scan.nextInt();
        scan.nextLine();

        return choice;
    }
}
```