

Testování aplikace

1.	Stažení branch	2
2.	Co přibylo nového?	2
3.	Uložené výsledky testování.	3
4.	Získání dat z JAVA.	4
5.	Spuštění testu v MS Excel.	4
6.	Porovnání se zadáním.	6
7.	Jaká data jsme testovali?	7

1. Stažení branch

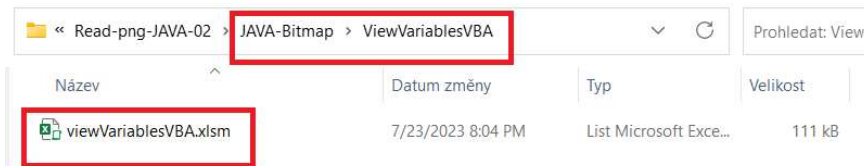
Aplikaci stáhneme z branch V02, zde:

<https://github.com/jonasRower/Read-png-JAVA/tree/V02>

2. Co přibýlo nového?

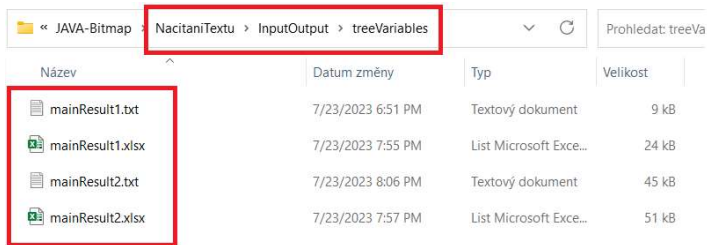
Oproti V01 můžeme aplikaci testovat, s tím, že testovaná data můžeme zobrazit graficky v MS Excelu.

Je zde nová složka obsahující Excelovské makro, pomocí něhož lze zobrazovat výsledky:



Obr. 1 – Umístění viewVariablesVBA.xlsm

Též V02 zatím obsahuje některé výsledky z testování, které se nachází ve složce:



Obr. 2 – Umístění výsledků testování

Zde, txt soubory jsou generovány kódem JAVY, soubory.xlsx jsou uloženy výsledky, které byly získány pomocí makra ve viewVariablesVBA.xlsm. Soubory.xlsx jsou originálem viewVariablesVBA.xlsm, "uložené jako" avšak bez makra, proto, .xlsx.

Též v kódu JAVA přibyla složka (package) logData obsahující nové třídy, které generují vybrané proměnné do txt.

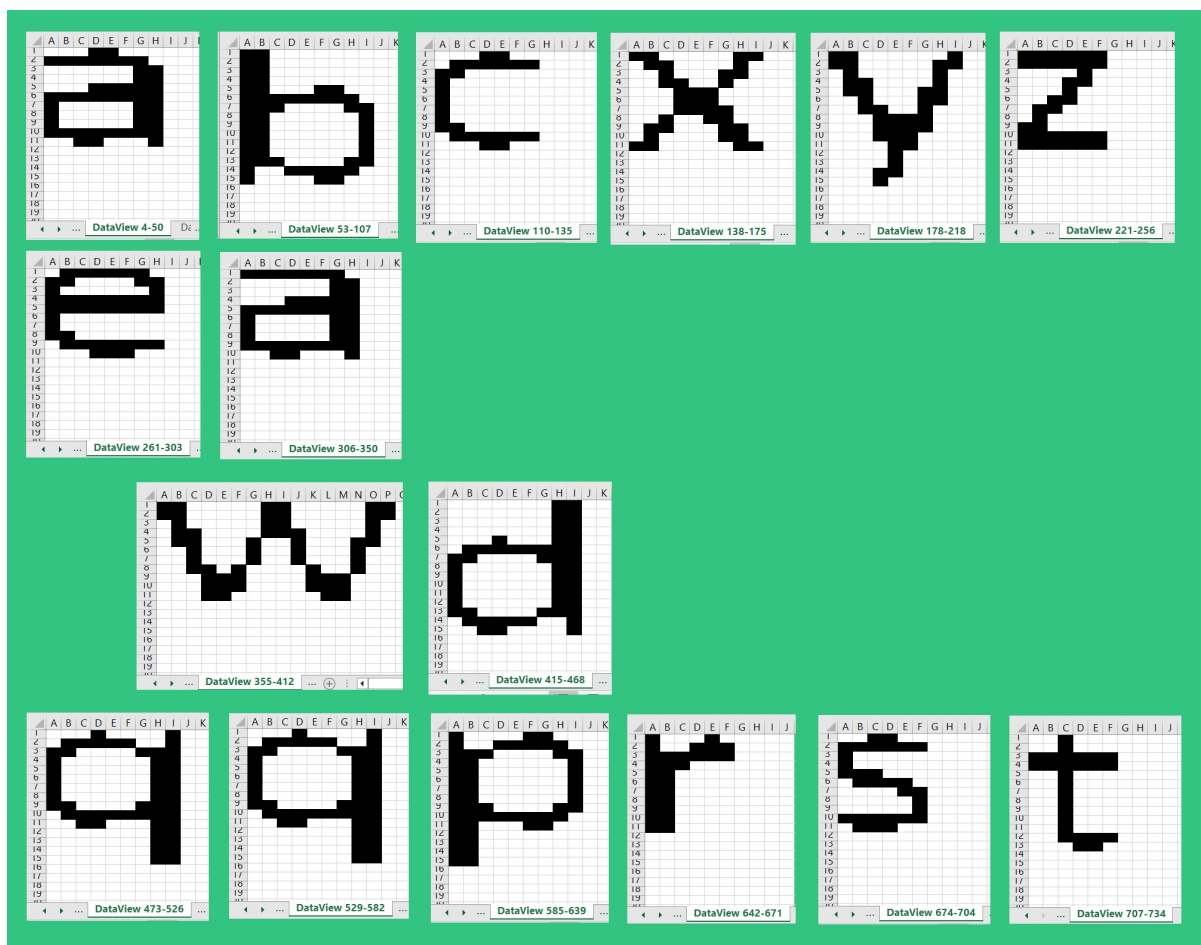
« JAVA-Bitmap > NacitaniTextu > project > src		
Název	Datum změny	Typ
createOutput	7/22/2023 2:41 PM	Složka souborů
logData	7/22/2023 8:36 PM	Složka souborů
nactiPng	7/16/2023 9:07 PM	Složka souborů
porovnaníSouradnic	7/16/2023 9:07 PM	Složka souborů
skupinaPismen	7/16/2023 9:07 PM	Složka souborů
souradniceAbecedy	7/16/2023 9:07 PM	Složka souborů
souradniceJednohoPismene	7/16/2023 9:07 PM	Složka souborů

Obr.3 – Nový část kódu, oproti V01.

3. Uložené výsledky testování.

Ve složce “\NacitaniTextu\InputOutput\treeVariables” lze nalézt uložené výsledky testu, Otevřeme soubor mainResult2.xlsx.

Jednotlivé listy obsahují data:



Obr. 4 – Uložené výsledky testování.

Tento soubor vznikl načtením dat mainResult2.txt. Otevřeme-li tento soubor, můžeme se přesvědčit, že každý list MS Excelu právě obsahuje 1 větev stromové struktury dat. Např. označení Listu "DataView 4-50" obsahuje data mezi 4. a 50. řádkem souboru mainResult2.txt.

4. Získání dat z JAVA.

Do kódu JAVA jsme vložili tento řádek:

```

58
59 //ziska data zkoumaneho obrazku
60 zdrojDataZkoum zkoumanaData = new zdrojDataZkoum(adresaProjektu);
61 SouradnicePismena.nactiDataZkoumanehoObrazku(zkoumanaData.getAdresaSlozky(), zkoumanaData.getNazevSouboruPng());
62 MapaVsechPismenNaVsechRadcich = SouradnicePismena.getMapaVsechPismenNaVsechRadcich();
63
64 //tiskne data do logu
65 convertArrayList csvOutput = new convertArrayList(MapaVsechPismenNaVsechRadcich, "mainResult2.txt");
66
67
68

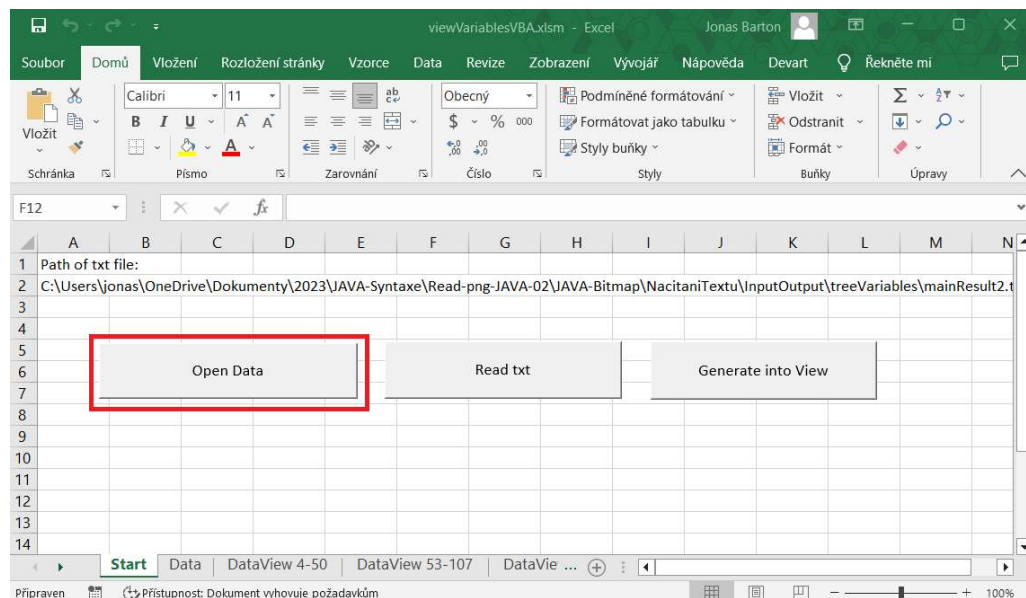
```

Obr. 5 – Za proměnnou, kterou potřebujeme testovat, vložíme tento kód.

Po spuštění JAVA kódu získáme soubor "mainResult2.txt" ve složce "\NacitaniTextu\InputOutput\treeVariables"

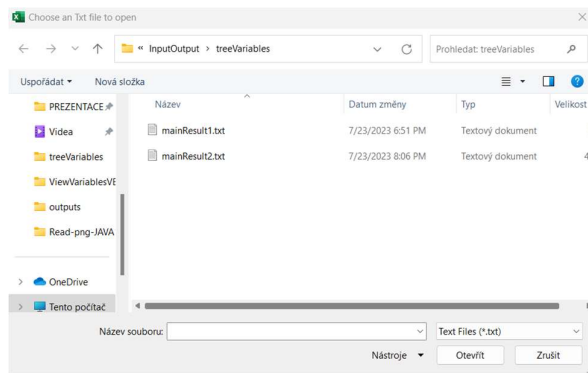
5. Spuštění testu v MS Excel.

Na adrese "\JAVA-Bitmap\ViewVariablesVBA" otevřeme soubor MS Excelu. Otevřeme s stiskneme tlačítko "Open Data".



Obr. 6 – Po otevření viewVariablesVBA.xlsm, načteme zdrojový (vygenerovaný) txt soubor.

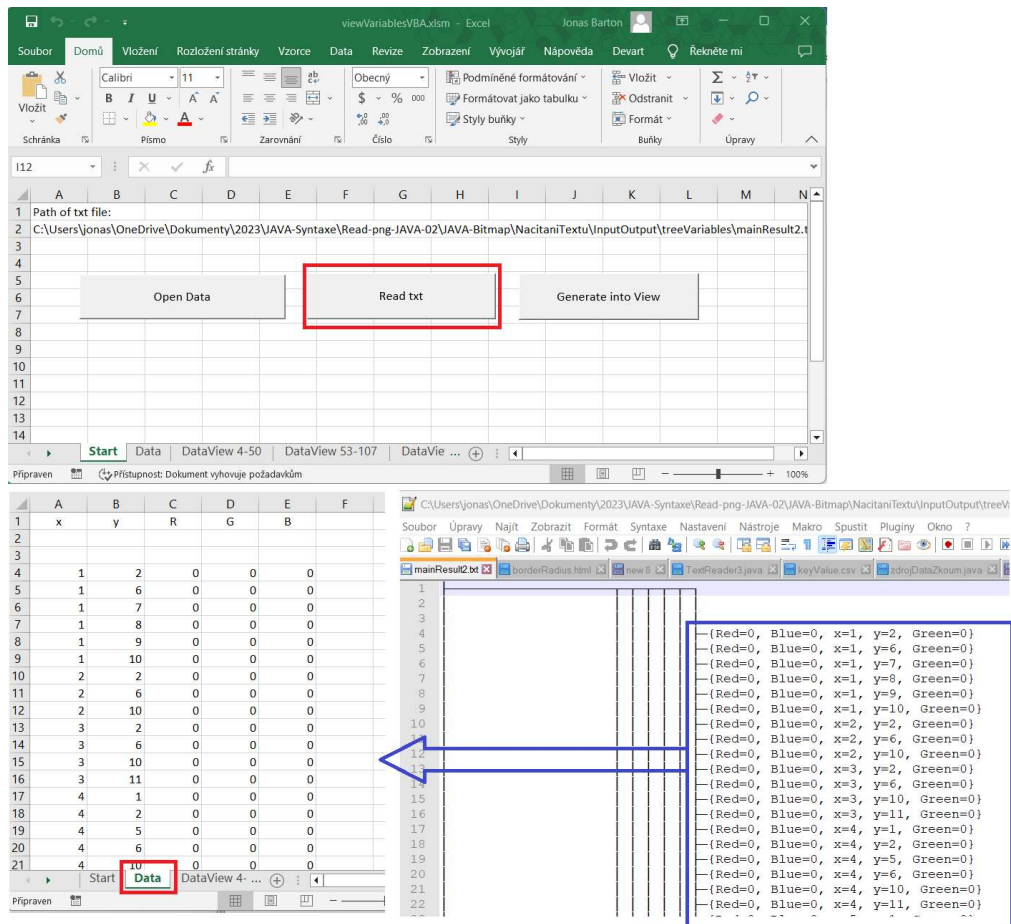
Následně jsme vyzváni k výběru txt souboru:



Obr. 7 – Vybrání zdrojového souboru.

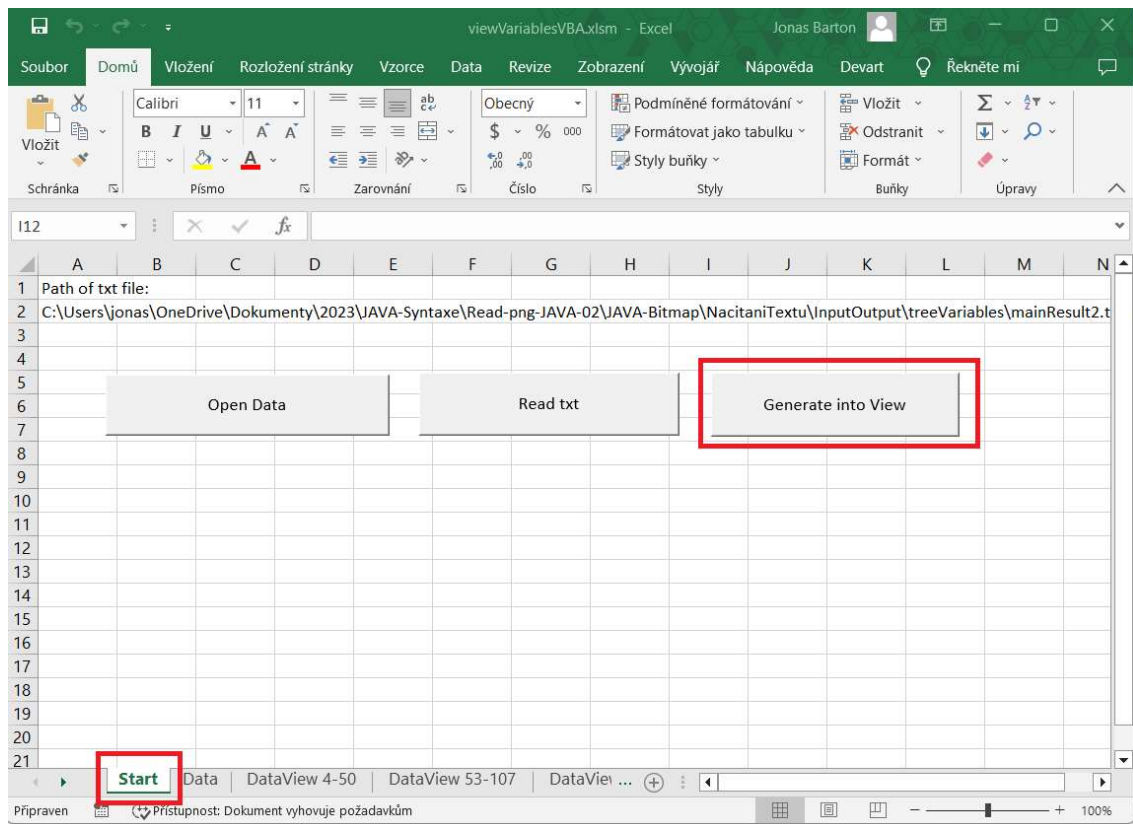
Vybereme tedy mainResult2.txt.

Následně klikneme na tlačítko "Read txt". MS Excel zapíše data stromu na List "Data"



Obr. 8 – MS Excel načte data ze zdrojového txt souboru a uloží je na list "Data".

Následně klikneme na tlačítko “Generate into View”.



Obr. 8 – Pomocí tlačítka “Generate into View”, se jednotlivá data vygenerují na příslušné listy.

MS Excel vytvoří jednotlivé listy, dle stromové struktury a vloží na ně příslušná data. Názvy listů nesou označení mezi kterými řádky zdrojového souboru se data nachází.

Všechny předchozí listy se mažou a vytváří se znovu, tudíž již neplatná data nám nevadí.

6. Porovnání se zadáním.

Výstup z testu jsme si již prohlédli, na obr. 4.

Porovnáme-li se zadáním, dospějeme k závěru, že data jsou správná.

a b c x y z
e a
w d
q q p r s t

Obr. 9 – Testovaný obrázek - zadání.

7. Jaká data jsme testovali?

Otestovali jsme tyto data:

The screenshot shows the source code of `TextReader.java` and the `Variables` window. The code defines a `MapaVsechPismenNaVsechRadcich` map and converts it to a CSV file. The variable inspector shows the internal structure of the map, which is a `HashMap` containing several `ArrayList` objects.

Name	Value	Type
MapaVsechPismenNaVsech	"size = 4"	ArrayList<ArrayList<Arra
[0]	"size = 6"	ArrayList
[0]	"size = 47"	ArrayList
[1]	"size = 55"	ArrayList
[2]	"size = 26"	ArrayList
[3]	"size = 38"	ArrayList
[4]	"size = 41"	ArrayList
[5]	"size = 36"	ArrayList
[1]	"size = 2"	ArrayList
[0]	"size = 43"	ArrayList
[1]	"size = 45"	ArrayList
[2]	"size = 2"	ArrayList
[0]	"size = 58"	ArrayList
[1]	"size = 54"	ArrayList
[3]	"size = 6"	ArrayList
[0]	"size = 54"	ArrayList
[1]	"size = 54"	ArrayList
[2]	"size = 55"	ArrayList
[3]	"size = 30"	ArrayList
[4]	"size = 31"	ArrayList
[5]	"size = 28"	ArrayList

Testování jiných dat můžeme provést obdobným způsobem.