

4.1. Záznamník terénnych dát

Kľúčové slová

záznamník dát, súbor, CSV formát, formát času, web, chyba, ergonómia aplikácie, odchyťovanie chýb,

Čo sa naučíme a čo si precvičíme

- Analyzovať požiadavky na funkcionality a ovládanie aplikácie pre konkrétne potreby.
- Testovať aplikácie na chyby a ošetriť chybové stavy.
- Vytvárať štruktúrované dáta vo formáte CSV.
- Zvoliť vhodnú dátovú reprezentáciu pre záznamy obsahujúce viac položiek.
- Zaznamenávať dáta do lokálneho aj do vzdialeného súboru.
- Používať geolokačný senzor, čas.

Príprava na výučbu

Žiaci by mali mať k dispozícii mobilné zariadenia s aktívnym geolokačným senzorom (GPS) a s pripojením do siete internet. V rozšírení aplikácie realizujeme ukladanie dát na internetový server. Pre realizáciu tejto časti je potrebné mať prístup na internetový server a možnosť umiestniť tam PHP skripty. Stačí, ak túto možnosť má len učiteľ.

Odporúčaný priebeh výučby

V tejto kapitole sa budeme zaoberať návrhom aplikácie, pomocou ktorej budeme môcť zaznamenávať dáta v teréne. Predpokladá sa, že aplikáciu používa pozorovateľ a svoje pozorovania priebežne zaznamenáva.

Výsledkom prvej časti je jednoduchá aplikácia, ktorá zaznamenáva dáta do CSV súboru vo vhodnom formáte, rieši chybné záznamy pozorovateľa, umožňuje pozorovateľovi aplikáciu otestovať a testovacie dáta následne zmazať. Výsledkom tejto časti je aplikácia, ktorá dokáže zaznamenané dáta odoslať webovej stránke. Záznamy od viacerých pozorovateľov vieme teda zhromaždiť na jednom mieste.

V časti Rozšírenie žiaci riešia niektoré problémové situácie, ktoré môžu počas používania aplikácie nastať:

- Pokus o odoslanie dát na server ak neexistuje lokálny súbor so záznamami.
- Úprava dát tak, aby sme na server neposielali chybné záznamy.
- Chyba pozorovateľa, ktorý označil viac chybných záznamov než mal k dispozícii.
- Posielanie prázdnych dát webovej stránke v prípade, že nie sú zaznamenané žiadne záznamy alebo všetky sú chybné.

Zaradenie úloh v tejto časti je na učiteľovi.

Záznamník hustoty dopravy

Výhody mobilných zariadení už poznáme. Sú ľahko prenositeľné, nezávislé na externom zdroji energie, využívajú bezdrôtové pripojenie do siete, majú zabudované množstvo senzorov a pod. Nečudo, že vďaka týmto vlastnostiam upútali pozornosť aj terénnych výskumníkov a pozorovateľov.

Správa ciest pravidelne monitoruje hustotu dopravy. Vďaka tomu vie lepšie plánovať výstavbu nových ciest, predpovedať ich záťaž a v konečnom dôsledku znižovať ekologickú záťaž životného prostredia. Ich pozorovatelia stoja na vybraných úsekoch ciest a zaznamenávajú dopravné prostriedky, ktoré po ceste prechádzajú. Na záznam využívajú papierové záznamové hárky. Tento spôsob nie je veľmi efektívny, navyše je prácne spojiť záznamy od viacerých pozorovateľov. Oslovili nás, či by sme im vedeli navrhnúť a vyvinúť efektívnejší systém pre záznam dopravy. Pozorovateľ v záznamníku uvádza miesto pozorovania, čas záznamu, typ dopravného prostriedku a jeho smer.

Analýzujeme zadaný problém

Otázky na zamyslenie

Aké dáta potrebujeme zaznamenávať?

Ktoré z týchto dát vieme získať automaticky a ktoré sú na rozhodnutí pozorovateľa?

V akom formáte budeme dáta zaznamenávať?

Kde budeme dáta zaznamenávať (ukladať).

Akú funkčnosť by mala výsledná aplikácia poskytovať?

Čo z predchádzajúceho vieme v prostredí AI2 implementovať a čo nie?

Pomoc k riešeniu úlohy

Žiaci by na základe predchádzajúcej analýzy mali dospieť k nasledovnému:

Zaznamenávať musíme čas, miesto, typ dopravného prostriedku a jeho smer. Čas a miesto vieme zaznamenávať automaticky (systémový čas a GPS senzor zariadenia). Typ dopravného prostriedku a jeho smer je výsledkom pozorovania pozorovateľa.

Keďže dáta budeme neskôr strojovo spracovávať, je vhodné ich zaznamenávať v nejakej štruktúre, napr.: čas, miesto, typ prostriedku, smer. Najjednoduchší spôsob, ako takéto dáta uchovávať je vo formáte CSV (čiarkou oddelené hodnoty) v súbore.

Množstvo dát, ktoré zaznamenávame je pomerne veľké (predstavte si, kolónu áut a každé z nich potrebujeme zaznamenať). Potrebujeme preto minimalizovať množstvo úkonov pozorovateľa. Asi nemôžeme predpokladať, že pozorovateľ bude záznam písať. Na to nie je čas. Ideálne by bolo, ak by na záznam jedného vozidla potreboval spraviť jeden úkon – napr. stlačiť tlačidlo.

Ďalšie vylepšenia, ktoré odporúčame riešiť až po tom, ako žiaci vytvoria funkčnú aplikáciu, sú napr.:

Akú spätnú väzbu dostane pozorovateľ po stlačení tlačidla (primárne sleduje dopravu)? Vibrovaním mu oznámime, že stlačil tlačidlo a aktuálny záznam mu zobrazíme aj na displeji.

Uvažovať by sme mali aj o tom, že pozorovateľ spraví chybný záznam a bude ho chcieť opraviť. Namiesto mazania záznamu v súbore je jednoduchšie vložiť špeciálny záznam, ktorým povieme, že predchádzajúci záznam je potrebné ignorovať.

Pozorovateľ si zrejme bude chcieť aplikáciu pred prvým použitím otestovať. Aplikácia by preto mala mať možnosť zmazať testovacie dáta (zmazať obsah súboru).

Ako spojiť dáta od viacerých pozorovateľov? Môžeme uvažovať aj o tom, že by sme dáta uchovávali nie len lokálne, ale aj centrálné na jednom mieste. Tu využijeme schopnosť mobilného zariadenia pripojiť sa do siete a dáta priebežne odosielať alebo jednorazovo odoslať do centrálného úložiska.

Možnosti vylepšenia aplikácie sú takmer nekonečné a je len na žiakoch ako sa problému zhostia.

Nepredpokladáme, že takúto analýzu a jej závery zvládnu žiaci naraz (napr. možnosť opravy chybného záznamu). Niektorým faktom možno nebudú pripisovať potrebnú dôležitosť (napr. vytvoriť záznam len stlačením jedného tlačidla). Aby sme sa vyhli opakovanej úprave aplikácie, odporúčame otázkami žiakov na tieto aspekty nasmerovať.

Vysvetlíme si

CSV formát (1) (Comma-separated values) je jednoduchý súborový formát. Dáta sú zaznamenané vo formáte textu, pričom každý riadok súboru predstavuje jeden záznam. Položky záznamu v riadku sú oddelené čiarkou. Výhodou formátu CSV je, že je jednoduchý a vedia s ním pracovať rôzne aplikácie (napr. tabuľkový kalkulátor). Každý riadok by mal obsahovať rovnaký počet položiek. Ak by niektorá položka záznamu mala obsahovať čiarku, je potrebné celú položku uzavrieť do úvodzoviek.

Ukážka súboru `vyplaty.csv`, ktorý obsahuje zoznam ľudí a výšku ich platu:

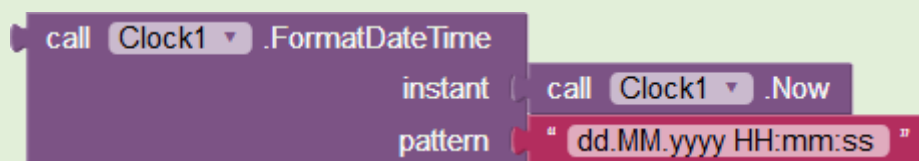
```
Jožko,Mrkvička,"580,32"  
Karol,Petrík,702  
Danka,Šikovná,"987,65"
```

Vysvetlíme si

Pri zaznamenávaní časovej značky by sme mali uvažovať, v akom formáte čas zaznamenať. Ak predpokladáme, že zaznamenané dáta o doprave budeme spracovávať v tabuľkovom tabulátore, mali by sme zvoliť formát, ktorý bude tabuľkový kalkulátor interpretovať ako čas,

resp. ako dátum a čas. Ak preskúame, aké formáty času podporuje tabuľkový kalkulátor, nájdeme medzi nimi aj formát: 30. 8. 2018 7:38:43. V tomto formáte by sme mohli zaznamenávať čas aj my.

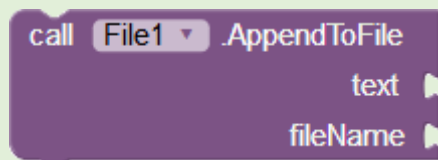
Pre prácu s časom použijeme komponent `Clock` (nájdeme ho v skupine `Sensors`) pomocou ktorého vieme pristupovať k systémovému času zariadenia (pozri: <http://ai2.appinventor.mit.edu/reference/components/sensors.html#Clock>). Súčasťou komponentu `Clock` je aj metóda pre formátovanie času (`FormatDateTime()`) ktorá upraví aktuálny čas podľa zadaného formátovacieho reťazca, resp. znakov tohto reťazca. (napr. hodina, mesiac apod.) Zoznam formátovacích znakov pre čas nájdeme na <https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>. Pre naše potreby môžeme čas formátovať nasledovne:



Výsledkom metódy `FormatDateTime()` je reťazec reprezentujúci čas v zadanom formáte. Výsledný reťazec použijeme ako súčasť záznamu.

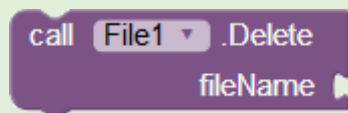
Vysvetlíme si

Pre prácu so súborom použijeme komponent `File` (nájdeme ho v skupine `Storage`). Komponent `File` ponúka niekoľko užitočných metód:



Pripojí text k obsahu súboru. Ak súbor neexistuje, vytvorí ho. Ak potrebujeme do súboru zapísať znak konca riadku, použijeme znak `\n`. Nasledujúci text bude pokračovať v ďalšom riadku.

Ak názov súboru začína prefixom lomka `/`, pokúsi sa systém lokalizovať súbor na SD karte. Ak prefix `/` vynecháme, systém lokalizuje súbor súkromnom priestore našej aplikácie.



Zmaže súbor.

Úloha 1

Vytvorte aplikáciu pre zaznamenávanie hustoty dopravy v dvoch smeroch. Aplikácia by mala umožniť zaznamenávať všetky dáta tak, ako to robia pozorovatelia do papierového záznamníka.

Pomôcka: Premyslite si, ktoré hodnoty viete získať automaticky a ktoré sú na rozhodnutí pozorovateľa.

Pomôcka: Nezabudnite, že geolokačný senzor je potrebné po spustení aplikácie zapnúť.

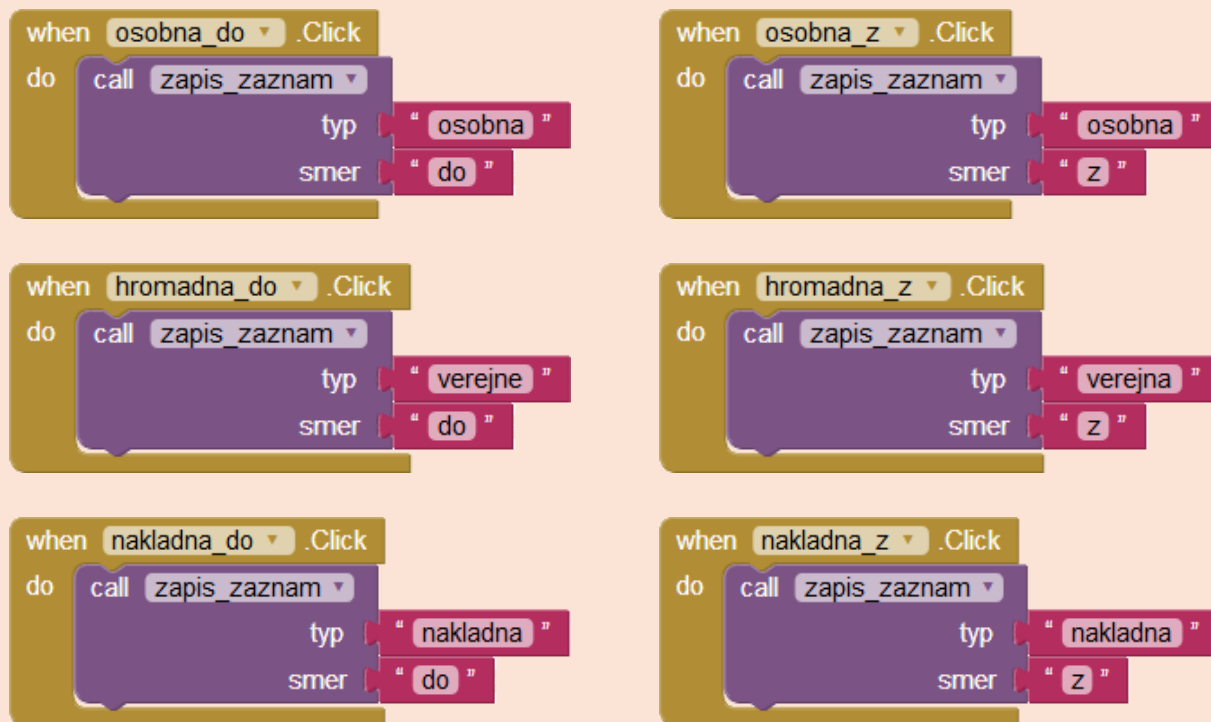
Pomoc k riešeniu úlohy

Podľa požiadaviek by sme mali zaznamenávať: miesto pozorovania, čas záznamu, typ dopravného prostriedku a jeho smer. Prvé dve hodnoty vieme prečítať zo senzorov zariadenia, druhé dve sú na rozhodnutí pozorovateľa.

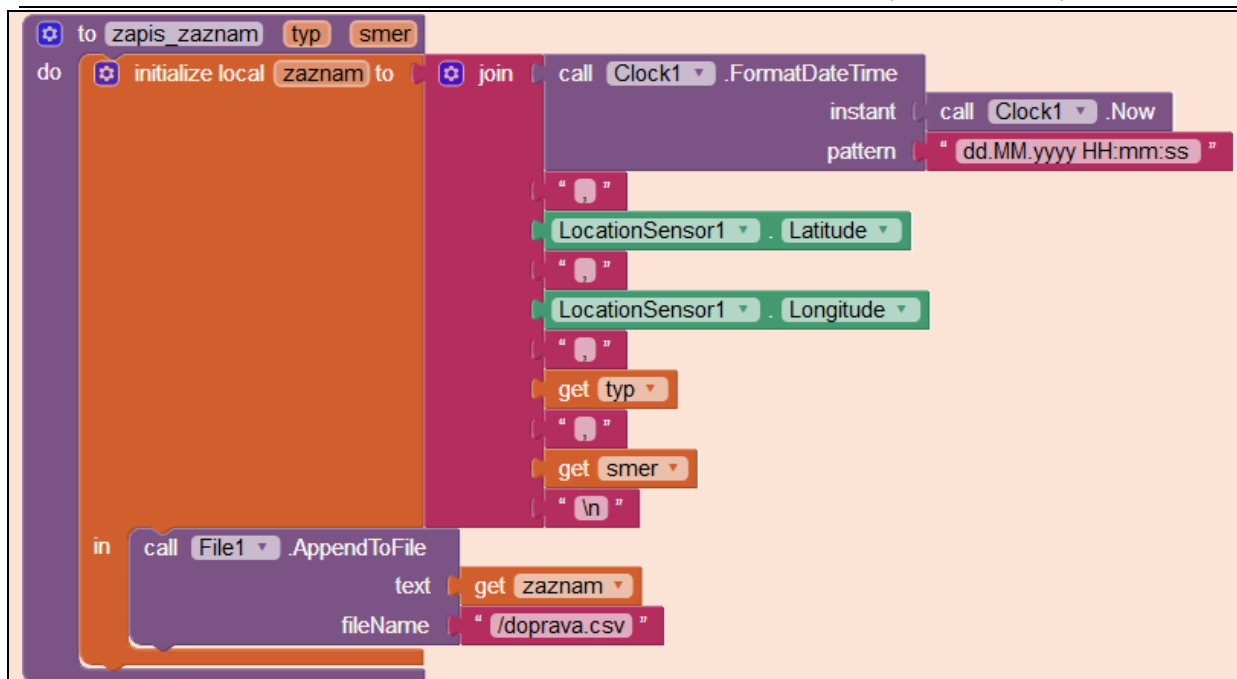
Uvažujme nasledujúce typy dopravných prostriedkov: osobné auto, prostriedok hromadnej dopravy a nákladné auto. Môžeme uvažovať dva smery, napr. smer do centra a smer z centra alebo smer do mesta a smer z mesta. Ak chceme minimalizovať počet úkonov pre záznam jedného dopravného prostriedku, môžeme pre každý typ prostriedku a jeho smer vytvoriť samostatné tlačidlo. Celkovo teda budeme potrebovať šesť tlačidiel:

osobna_do	osobna_z
hromadna_do	hromadna_z
nakladna_do	nakladna_z

Po stlačení každého z nich vložíme jeden záznam do súboru. Výhodné je vytvoriť si pre zápis záznamu samostatnú procedúru (`zapis_zaznam`). Pomocou parametrov jej oznámime, aké dáta do súboru zaznamenať. Udalosť kliknutie nastavíme jednotlivým tlačidlám nasledovne.



Procedúru `zapis_zaznam()` s dvoma parametrami `typ` a `smer` môžeme definovať nasledovne:



Minimalistická verzia aplikácie ponúka šesť tlačidiel. Po stlačení každého z nich sa uloží príslušný záznam do súboru.

Úloha 2

Pozorovateľ môže spraviť chybu a stlačiť nesprávne tlačidlo. Takýto záznam by zrejme znehodnotil prieskum hustoty dopravy.

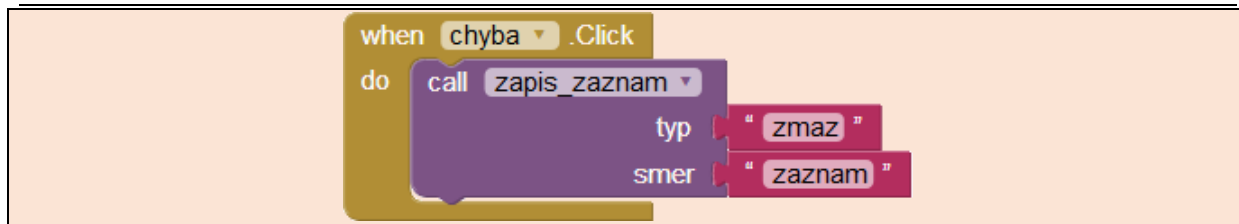
Navrhňte a implementujte spôsob, ako postupovať v prípade chybného zápisu (chyby pozorovateľa).

Pomôcka: Prediskutujte vzájomne rôzne riešenia a ich efektívnosť.

Pomoc k riešeniu úlohy

V prípade chyby pozorovateľa sa chybný záznam uloží do súboru. Zmazať záznam v súbore nie je jednoduché. Potrebovali by sme celý obsah súboru, okrem chybného záznamu, prekopírovať do pomocného súboru. Pôvodný súbor zmažeme a obsah pomocného súboru prekopírujeme do súboru s menom pôvodného súboru. Ak by súbor obsahoval veľa dát, tento postup je prakticky nepoužiteľný.

Chybný záznam v súbore nemusíme mazať, stačí si len do súboru poznamenať, že záznam je chybný. Pri spracovaní dát, budeme takto označený záznam ignorovať. Vložme za chybný záznam informáciu o tom, že záznam je chybný. Vytvorme si na to samostatné tlačidlo (chyba).

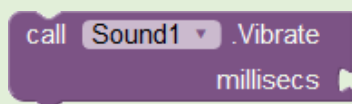


Úloha 3

Pozorovateľ pri zaznamenávaní musí sledovať dopravu a zároveň pracovať s aplikáciou. Niekedy si nie je istý tým, či tlačidlo stlačil alebo nie. Navrhnete a implementujete spôsob ako poskytnúť pozorovateľovi možnosť overiť si, či a aké tlačidlo stlačil.

Vysvetlíme si

Pri aplikáciách ovládaných dotykom na obrazovke je problém v tom, že primárne nemáme spätnú väzbu po stlačení tlačidla (resp. po dotyku na obrazovke). Riešením je spustiť nejakú relevantnú reakciu, napr. zmenu obrazovky, zobrazenie textu alebo zavibrovanie zariadenia.

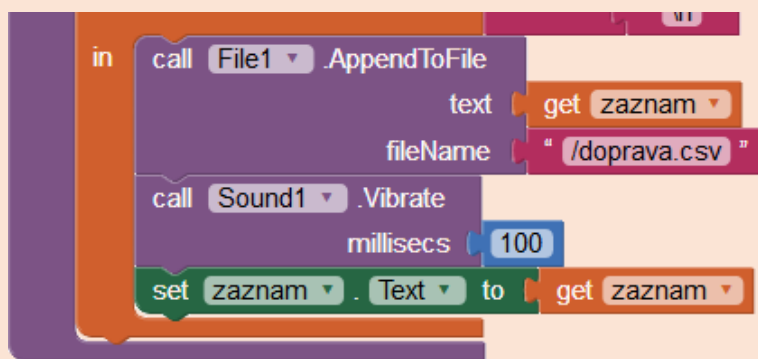


Spustí vibrovanie zariadenia na zadaný počet milisekúnd.

Pomoc k riešeniu úlohy

Aj tu by sme mali zohľadniť časový stres pozorovateľa. Spätná väzba pri stlačení tlačidla by mala byť automatická. To, že tlačidlo bolo stlačené vieme signalizovať napr. krátkou vibráciou zariadenia. Informáciu o tom, ktoré tlačidlo bolo stlačené, zobrazíme textom na obrazovke. Využijeme komponent `Label` zo skupiny User Interface. V programe sme ho pomenovali `zaznam`.

Pri stlačení niektorého z tlačidiel sa volá procedúra `zapis_zaznam()`. Je preto rozumné spustenie vibrovania a zobrazenie informácie o stlačení tlačidla umiestniť práve tam.



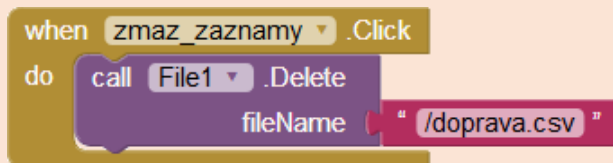
Úloha 4

Pri zaškoľovaní pozorovateľov sa do súboru uloží množstvo testovacích záznamov. Zrejme by sa dali ignorovať použitím tlačidla `chyba`, ale nie je to veľmi praktické riešenie. Upravte aplikáciu tak, aby sa dali testovacie dáta v súbore zmazať.

Pomoc k riešeniu úlohy

Riešenie je jednoduché. Stačí súbor zmazať. Ak súbor neexistuje, pri nasledujúcom zápise (metóda `AppendToFile()`) sa súbor vytvorí.

Aj pre tento prípad si môžeme vytvoriť samostatné tlačidlo.



Úloha 5

Nami navrhnutý záznamník pracuje spoľahlivo a pozorovateľom značne uľahčuje ich prácu. Správa ciest pri monitorovaní dopravy však využíva viacero pozorovateľov súčasne. Každý zaznamenáva hustotu dopravy na inom mieste. Pre výsledné spracovanie by bolo potrebné tieto záznamy spojiť v jednom, centrálnom bode.

Správa ciest má vytvorenú webovú stránku, ktorá čaká na dáta od pozorovateľov a tie potom uloží do súboru na serveri. V našej aplikácii potrebujeme vyriešiť to, ako dáta zo súboru `doprava.csv` prečítať a ako ich webovej stránke poslať.

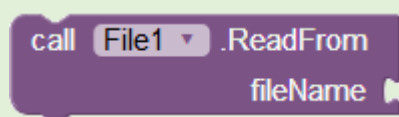
Navrhните a implementujte spôsob ako dáta zo súboru prečítať a poslať ich webovej stránke. Ak webová stránka dáta akceptuje, odošle späť odpoveď `ok`.

Pomoc k riešeniu úlohy

V súbore `zaznamenaj.php` je jednoduchý php skript (2), ktorý prečíta zaslané dáta a uloží ich do súboru `data.csv`. Súbor skopírujte na verejne dostupný webový server. Pričínok, kde je skript umiestnený, musí mať nastavené práva pre zápis (aby skript mohol na serveri vytvoriť súbor `data.csv`). Skript očakáva, že odosielané dáta pomenujeme menom `data`. Zaslané dáta skript uloží do samostatného riadku v súbore `data.csv`. Ak všetko prebehlo v poriadku, skript späť odošle správu `ok`. Po prijatí tejto správy môžeme lokálne uložené záznamy zmazať. Skript nijako nekontroluje formát dát. Je možné ho preto využiť pre ľubovoľné dáta.

Odporúčame, aby učiteľ skript na server skopíroval a žiakom poskytol adresu skriptu.

Vysvetlíme si



Prečíta obsah súboru. Následne je vyvolaná udalosť `GotText()`.

when **File1** .GotText
text
do

Udalosť nastane, ak prečítame obsah súboru. Obsah súboru je prístupný v premennej `text`.

Vysvetlíme si

Komponent `Web` (v skupine `Connectivity`) je určený pre komunikáciu s webovou stránkou. Komponent `Web` umožňuje posilať stránke dáta a rovnako od stránky dáta prijímať.

set **Web1** .Url to

Nastaví adresu stránky, s ktorou chceme komunikovať.

call **Web1** .PostText
text

join ("nazov=" "hodnota")

Pošle dáta stránke. Ak posielame dáta stránke, pomenujeme ich, napr. `nazov=`.

when **Web1** .GotText
url responseCode responseType responseContent
do

Udalosť nastane, ak stránka pošle nejaké dáta našej aplikácii (najčastejšie ako odpoveď, keď na stránku pristupujeme). Tento fakt využijeme, aby sme si potvrdili, že stránka naše dáta dostala, rozumie im a že si ich uložila do súboru na serveri.

Pomoc k riešeniu úlohy

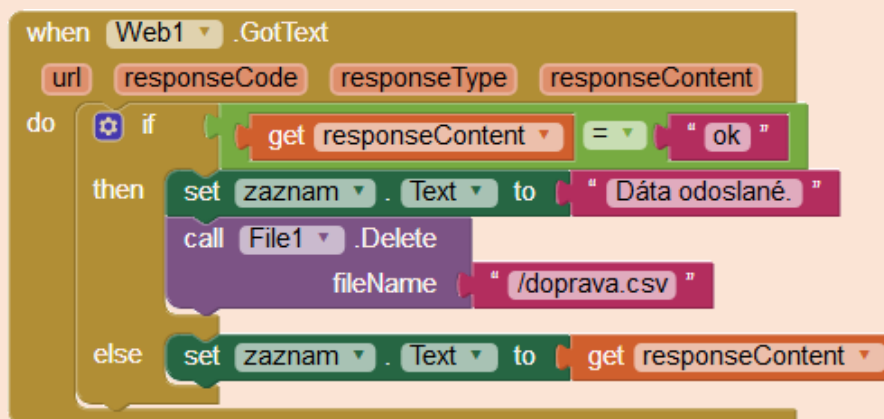
Pre odosielanie dát na server si vytvoríme samostatné tlačidlo. Po jeho stlačení prečítame obsah súboru, kde máme lokálne uložené záznamy.

when **odosli_na_server** .Click
do
call **File1** .ReadFrom
fileName "/doprava.csv"

Prečítanie dát so súboru vyvolá udalosť `GotText` komponentu `File`. Ak táto udalosť nastane, môžeme dáta odoslať webovej stránke. Nezabudnite nastaviť správnu adresu skriptu.



Po odoslaní dát skriptu, skript odpovie. Očakávaná odpoveď je `ok`. V tomto prípade sa dáta na server uložili a lokálne dáta môžeme zmazať. Informáciu o úspechu, resp. neúspechu môžeme pozorovateľovi zobrazíť na obrazovke.



Čo by sme mohli viac preskúmať? Ako vylepšiť či rozšíriť záznamník dopravy?

Nasledujúce úlohy predstavujú možné vylepšenia aplikácie. Niektoré zvyšujú komfort používania aplikácie, iné riešia problémové situácie. Je možné, že niektoré z problémov ste postrehli pri predchádzajúcich úlohách a už ich vyriešili.

Úloha 6

Otestujte vami naprogramovanú aplikáciu v rôznych situáciách. Použite ju spôsobom, ktorý ste nepredpokladali pri jej vývoji. Správa sa aplikácia vždy korektne alebo ste ju dokázali dostať do chybného stavu? Vie aplikácia v každej situácii vykonať vami požadovaný úkon?

Úloha 7

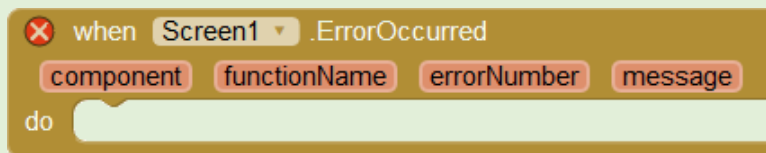
Ak lokálny súbor neexistuje (pri prvom spustení aplikácie, po zmazaní testovacích dát), pri pokuse o odoslanie dát na server nastane chyba. Navrhnete spôsob, ako tejto chybe predísť.

Vysvetlíme si

Pri práci s aplikáciou môžu nastať chyby, ktoré je vopred ťažké predpovedať alebo im zabrániť. Výsledkom je, že aplikácia zobrazí nejakú chybovú, pre pozorovateľa nejasnú správu. Takéto správanie nie je žiadúce. Dobre naprogramovaná aplikácia by takéto chyby nemala

zobrazovať. Používateľa by mala upozorniť vhodne zvoleným spôsobom a presne popísať, aká chyba nastala.

App inventor obsahuje jednoduchý mechanizmus pre odchyťávanie chýb. Pomocou metódy `ErrorOccured()` komponentu `Screen` vieme odchytiť a zareagovať na chybný stav.



Úloha 8

Pri odosielaní dát na server odosielame aj chybné dáta (ak pozorovateľ urobil chybný záznam). Výhodnejšie by bolo uložené dáta analyzovať a chybné záznamy vynechať. Navrhните a implementujte spôsob ako z lokálnych dát vynechať chybné záznamy a rovnako aj záznamy informujúce o chybe.

Vysvetlíme si

Štruktúrované dáta môžeme reprezentovať rôznymi spôsobmi. O formáte CSV sme hovorili v predchádzajúcej časti. Tento formát sa hodí skôr pre súbory. Pri práci s dátami CSV je vhodné ich transformovať do zoznamu – každý záznam (riadok) z CSV súboru bude samostatným prvkom zoznamu.

list from csv table text CSV text skonvertuje do zoznamu. Jednotlivé záznamy (riadky) textu sú transformované do samostatných položiek zoznamu.

list to csv table list Skonvertuje zoznam do textu v CSV formáte. Jednotlivé položky zoznamu sú transformované do samostatných riadkov v texte.

Pomoc k riešeniu úlohy

Analyzovať záznamy a vynechať tie, ktoré sú chybné má zmysel robiť hromadne. Teda nie vždy pri vytvorení záznamu. Vhodné je to spraviť pred odosielaním záznamov na server.

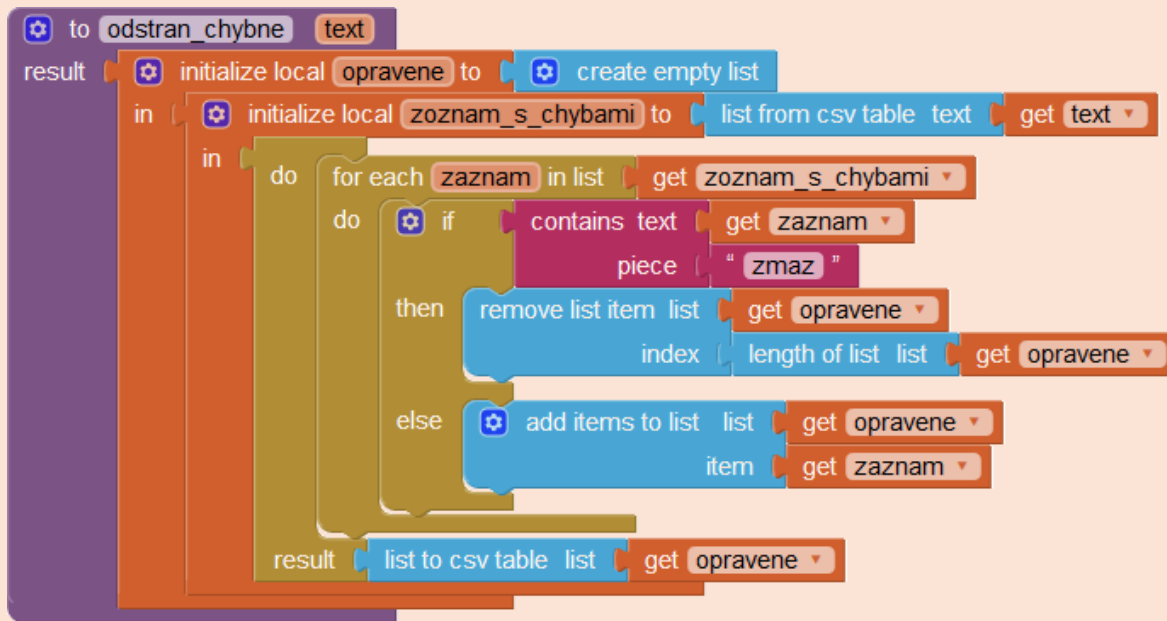
Odstrániť chybné záznamy môžeme postupnosťou nasledovných krokov:

- Prečítame text zo súboru (`File.ReadFrom()`).
- Prečítaný text skonvertujeme do zoznamu (`list.list from csv table()`). Každý riadok súboru predstavuje jeden prvok zoznamu.
- Vytvoríme si nový pomocný prázdny zoznam (`list.create empty list()`). Do tohto zoznamu budeme kopírovať záznamy zo zoznamu, ktorý sme vytvorili zo záznamov v súbore.
- Prejdeme položkami pôvodného zoznamu.
Ak je položkou záznam dopravy, prekopírujeme ho do nového zoznamu.

Ak je položkou záznam o chybe, z nového zoznamu odstránime naposledy vloženú položku.

- Položky nového zoznamu spojíme do jedného textu (`list.list to csv table()`). Jednotlivé položky vo výslednom texte budú oddelené znakom konca riadku.

Odporúčame na túto úpravu vytvoriť procedúru, ktorej pošleme text prečítaný zo súboru a procedúra vráti text bez chybných záznamov.

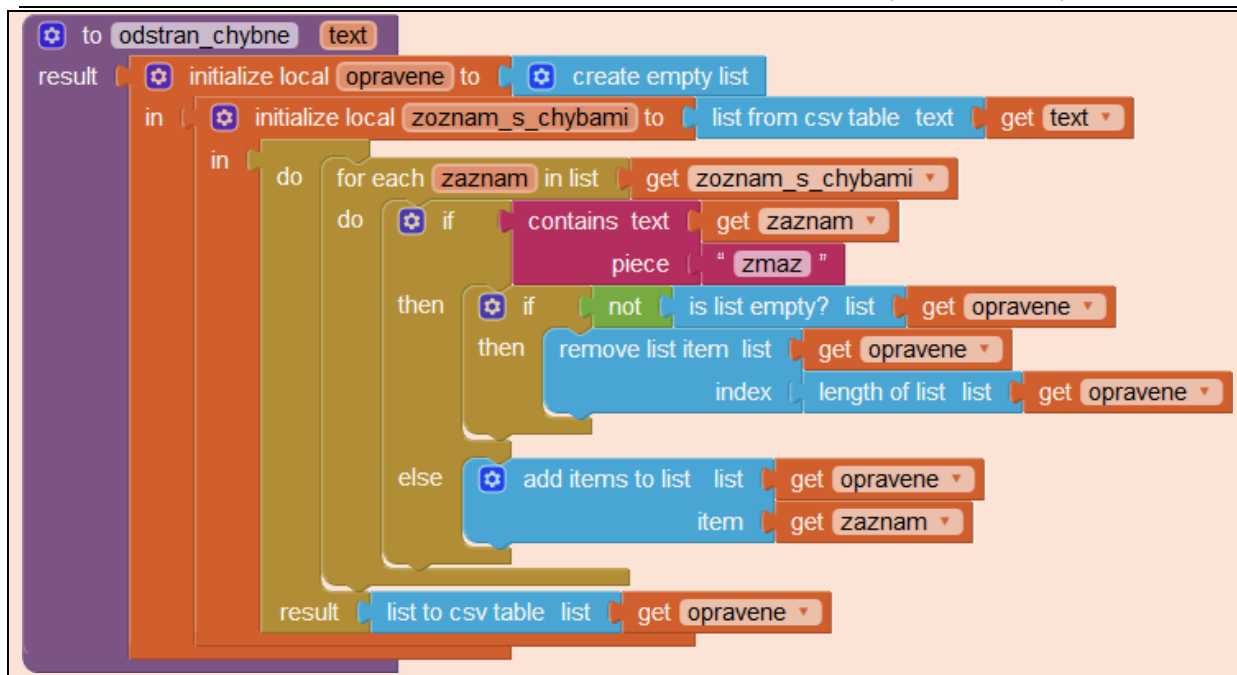


Úloha 9

Pozorovateľ mohol spraviť aj ďalší typ chyby. Tlačidlo pre zmazanie chybného záznamu stlačil viackrát než ako bol počet záznamov pred tým. Pri pokuse o odstránenie chybných záznamov (pred odoslaním na server) sa pokúsime odstrániť záznam z prázdneho zoznamu. To samozrejme spôsobí chybu. Navrhните a implementujte spôsob ako tejto chybe predísť.

Pomoc k riešeniu úlohy

Riešenie je v podstate jednoduché. Pred odstránením prvku zo zoznamu otestujeme, či zoznam nie je prázdny. V prípade prázdneho zoznamu odstránenie vynecháme.

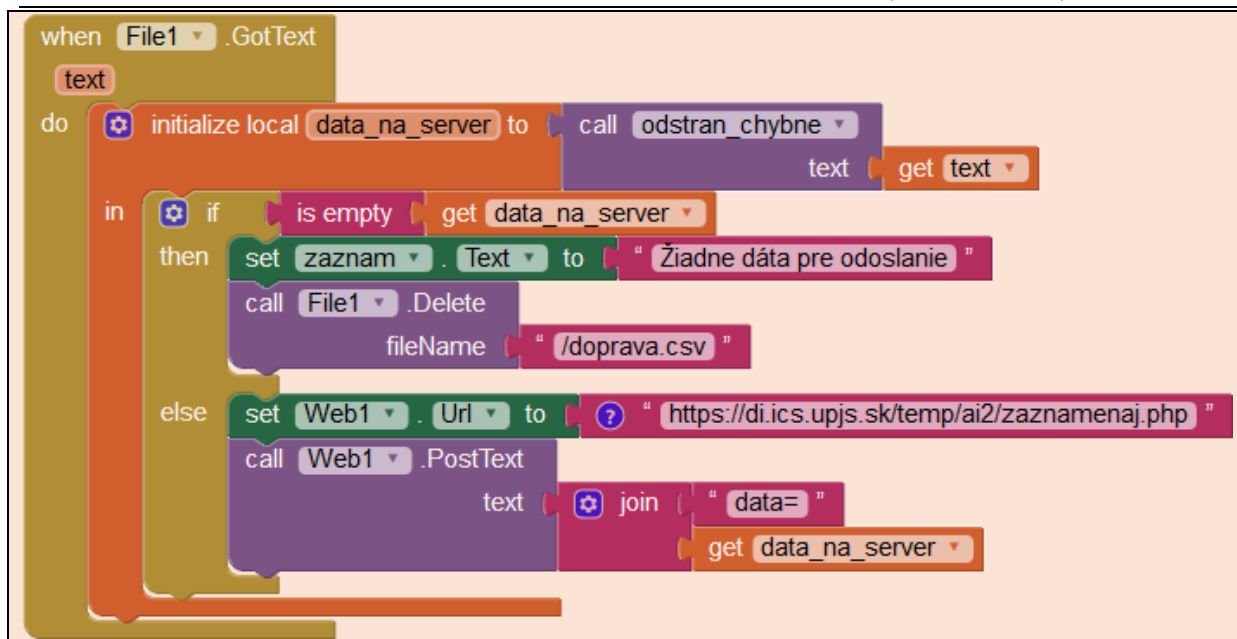


Úloha 10

Pri úprave záznamov sa môže stať, že výsledkom je prázdny zoznam. Súbor záznamov síce obsahoval nejaké záznamy, ale obsahoval aj informáciu o tom, že tieto záznamy sú chybné. Výsledkom je, že na server posielame prázdny reťazec (žiadne záznamy). Navrhните spôsob ako tomuto zabrániť a ako informovať pozorovateľa, že nie sú žiadne záznamy pre odoslanie na server.

Pomoc k riešeniu úlohy

Dáta posielame webovej stránke po prečítaní súboru doprava.csv. (udalosť `File.GotText()`). V tejto procedúre otestujeme, či máme čo poslať na server. Stačí otestovať, či procedúra `odstran_chybne()` vrátila prázdny reťazec. Ak je výsledkom úpravy dát prázdny reťazec, vypíšeme informáciu pre pozorovateľa a súbor môžeme zmazať. Aj keď by obsahoval nejaké záznamy, všetky sú chybné. V opačnom prípade záznamy odošleme.



Čo sme sa naučili

- analyzovať požiadavky používateľov a implementovať ich do návrhu aplikácie,
- testovať aplikáciu s cieľom detegovať chybné stavy a ošetriť tieto chybné stavy,
- popísať dáta v štruktúrovanom formáte CSV,
- ukladať a čítať dáta zo súboru,
- konvertovať dáta z formátu CSV do zoznamu a naopak,
- komunikovať z webovou stránkou.

Sebahodnotiaca karta

	s výdatnou pomocou učiteľa	s pomocou učiteľa	samostatne
Viem vytvoriť program, ktorý zapisuje do súboru			
Viem vytvoriť program, ktorý zapisuje dáta do súboru v CSV formáte			
Viem vytvoriť program, ktorý zistí systémový čas			
Viem vytvoriť program, ktorý formátuje systémový čas			
Viem vytvoriť procedúru s parametrami			
Viem ako zistiť, že nastala chyba pri vykonávaní programu			
Viem vytvoriť program, ktorý pošle dáta webovej stránke			
Viem konvertovať dáta z CSV formátu do zoznamu a naopak			

Skúsenosti z výučby

Predpokladané žiacke riešenia, ich možné miskoncepce, alternatívny prístup, ... doplníme po overení.

Bibliografia

1. **prispievatelia Wikipédie**. Comma-separated values. *Wikipédia, Slobodná encyklopédia*. [Online] 2017. https://sk.wikipedia.org/wiki/Comma-separated_values.
2. **Refsnes Data**. PHP 5 Tutorial. *W3Schools Online Web Tutorials*. [Online] <https://www.w3schools.com/php/default.asp>.

Register pojmov

C

Clock
 FormatDateTime, 4

F

File, 4
 AppendToFile, 4
 Delete, 4
 GotText, 9
 ReadFrom, 8
formát
 csv, 3

C

chyba
 odchytávanie. *Pozri* Screen:ErrorOccurred

L

list
 list from csv table, **11**
 list to csv table, **11**

S

Screen
 ErrorOccurred, 11
Sound
 Vibrate, 7

W

Web, 9
 GotText, 9
 PostText, 9
 Url, 9