

4. Pliki

Zadanie 4.1 Dane zawodników

Korzystając z pliku CSV z danymi skoczków narciarskich (zawodnicy.csv) napisz programy, które wczytują ten plik i:

1. liczy ile łącznie ważą reprezentanci Polski (np. żeby sprawdzić czy zmieszczą się w windzie na skocznię ;)). Pozwól użytkownikowi podać kraj (niekoniecznie musi być Polska).
2. wypisuje najwyższego, najniższego, najcięższego i najlżejszego skoczka; gdyby kilku miało taką samą wagę lub wzrost, to wystarczy wypisać jednego z nich.
3. dla wszystkich (!) krajów oblicza ilu jest zawodników z tego kraju; tzn. ma się wypisać, być może w innej kolejności:
AUT – 2
FIN – 3
GER – 5
NOR – 3
POL – 3
USA – 1
4. jak wyżej, ale liczy jeszcze dla każdego kraju średni wzrost zawodników.

Zadanie 4.2 Dane z pliku "sprzedaz"

W pliku sprzedaz.csv znajdują się dane opisujące sprzedaż różnych sklepów - w każdej linii jest jedna transakcja. Cena jest ceną jednostkową, a w osobnej kolumnie jest podana ilość sztuk. Dopiero ich iloczyn jest wartością transakcji – trzeba to liczyć w programie.

Napisz programy, które czytają dane z tego pliku i obliczają:

1. Sumę wartości wszystkich transakcji z całego pliku.
2. Ilość, sumę oraz minimalną, maksymalną i średnią wartość transakcji z wybranego miasta (niech program pyta o nazwę miasta na początku).
3. Dla każdego miasta występującego w pliku – sumę transakcji z tego miasta (wypisać bez powtórzeń), czyli zastosować schemat grupowania.

Zadanie 4.3 Filtrowanie sprzedaży

Napisz program, który filtruje dane z pliku takiego jak sprzedaz.csv:

- Pozwala za pomocą JFileChooser wybrać plik, który będzie czytany,
- Pyta o kategorię towaru (to ta kolumna, w której są: meble, wyposażenie, ...)
- Za pomocą JFileChooser pozwala wybrać lokalizację pliku, który będzie zapisany.
- Czyta dane z pierwszego pliku i zapisuje do drugiego pliku tylko te rekordy, w których są transakcje z wybranej kategorii. Wypisuje informację ile było takich rekordów.

Można to wykonać jako zwykły program z mainem, ale można też spróbować jako aplikację okienkową "z guziczkami". Wtedy atrakcyjny byłby wybór kategorii z listy (JComboBox lub JList).

5. Aplikacje okienkowe (Swing)

Stwórz jedną lub więcej aplikacji okienkowych w technologii Swing. Wygląd interfejsu przygotuj w edytorze wizualnym, np. w NetBeans (New JFrameForm), ewentualnie w Eclipse z doinstalowaną wtyczką Window Builder (New WindowBuilder > Swing Designer > Application Window) lub w IntelliJ.

Można stworzyć aplikację według własnego pomysłu (mile widziane), albo wybrać coś poniższych propozycji.

Zadanie 5.1 BMI (tylko, jeśli ktoś potrzebuje bardzo prostego zadania)

Człowiek podaje swój wzrost i wagę, a otrzymuje wyliczony współczynnik BMI i informację tekstową czy jest w normie, czy ma się odchudzać, czy raczej przytyć.

Niektóre źródła na temat BMI rozróżniają normy ze względu na wiek lub płeć. Opcjonalnie możesz w swoim programie uwzględniać także te informacje.

Zadanie 5.2 Konwerter jednostek

Napisz program, który służy do przeliczania wartości między różnymi systemami miar. Minimum to program, który przelicza jedną parę, np. mile na kilometry. Można spróbować zrobić bardziej rozbudowaną aplikację, np. po jednej stronie ekranu umieścić pola na dane w jednostkach metrycznych (centymetry, metry, kilometry, kilogramy, Celcjusze), a z drugiej brytyjskich (cale, stopy, mile, funty, Fahrenheity), a za pomocą przycisków można przeliczać w jedną lub w drugą stronę. Własne pomysły na układ okna i sposób działania mile widziane.

Zadanie 5.3 Automat na monety

Zrealizuj przykład z automatem parkingowym jako aplikację okienkową. Użytkownik podaje liczbę godzin, za które płaci, automat wylicza opłatę, następnie „wrzuca się monety” (np. przyciski dla różnych monet), a automat odejmuje wrzucone monety od kwoty do zapłaty, na końcu „wydaje resztę”.

Zamiast automatu parkingowego można wymyślić np. automat biletowy (z biletami normalnymi i ulgowymi), z kawą, z biletami do ZOO (normalne, ulgowe, rodzinne – z listy do wyboru) itp.

Zadanie 5.4 Kółko i krzyżyk

Program, który pozwala przeprowadzić rozgrywkę w kółko i krzyżyk. Można np. w oknie розміścić 9 przycisków, w których będą odpowiednie symbole. Gdy użytkownik kliknie w przycisk, jest to traktowane jako ruch i symbol się zmienia. Program prawidłowe kliknięcia traktuje na przemian jako ruchy jednego lub drugiego gracza. Program sam powinien rozpoznać kiedy dochodzi do wygranej lub kiedy wszystkie pola zostają zajęte i gra kończy się remisem.

To zadania można spróbować zrobić bez edytora graficznego, pisząc wszystko od zera i używając GridLayout do rozmieszczenia 9 guzików.

Zadanie 5.5 Wilk, koza, kapusta

Starożytna łamigłówka: Po jednej stronie rzeki znajdują się **wilk**, **koza** i **kapusta** oraz przewoźnik, który ma łodzią przewieźć je na drugą stronę rzeki. Problem polega na tym, że w łodzi zmieści się tylko jedno zwierzę/rzecz na raz – przewoźnik musi więc przewozić po jednej rzeczy i zostawiać je na brzegu. Jeśli jednak bez opieki pozostaną wilk i koza – wilk zje kozę; gdy zostaną zaś koza i kapusta – koza zje kapustę. Łamigłówka polega na tym, aby ustalić jak bezpiecznie przewieźć wszystkie trzy elementy na drugą stronę rzeki.

Zgodnie z opisanymi zasadami napisz prostą grę jako program w technologii Swing. W prostszej wersji możesz użyć wyłącznie przycisków i etykiet tekstowych. W miarę możliwości i dostępnego czasu możesz spróbować wprowadzić obrazki (poczytaj np. o ImageIcon i dodawaniu ich do JLabel/JButton).

Zadanie 5.6 Analiza danych z pliku CSV

Program ma umożliwiać analizę danych wczytywanych z pliku takiego jak emps.csv (ewentualnie można zrobić wersję dla zawodnicy.csv lub sprzedaz.csv...).

Po naciśnięciu przycisku "Przeglądaj" za pomocą JFileChooser można wybrać plik z dysku. Używając FileNameExtensionFilter wyświetlaj tylko pliki .csv. Po wybraniu pliku, jego dane powinny zostać wczytane (jako List<Employees>) i w odpowiednich labelach wyświetlone informacje sumaryczne: liczba pracowników, średnia pensja (ew. jeszcze min i max pensja).

Jeśli się uda: W oknie umieść JComboBox, z którego można wybrać nazwę departamentu. Lista departamentów (jako DefaultComboBoxModel) powinna być ustalona na podstawie danych z pliku! Po wybraniu departamentu z listy, w odp. miejscach okna wyświetl informację o ilości pracowników w tym departamencie oraz ich średniej pensji.

