



# SEMESTRÁLNÍ PRÁCE

Kristýna Kohoutová

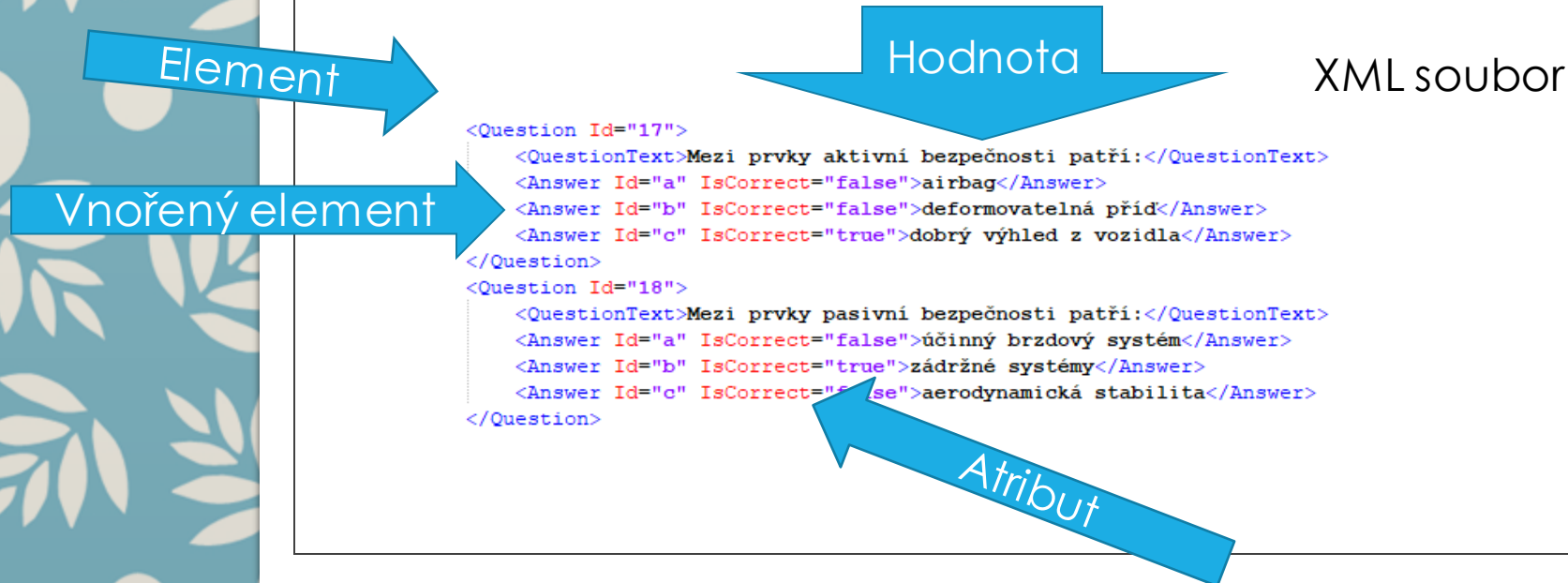
# XML

- Univerzální značkovací jazyk pro ukládání strukturovaných dat
- Hodnoty se vkládají do elementů nebo atributů elementů
- Element je vždy párový tag pojmenovaný podle typu údaje, který obsahuje např. `<Name>Kristýna</Name>`
- Oproti CSV souborům je XML soubor větší, protože obsahuje i názvy jednotlivých elementů (tagů)
- Podobný princip zápisu jako v HTML
- Aby byl soubor tzv. validní musí obsahovat právě 1 kořenový element a potom podřízené elementy

# XML - rozdíly oproti CSV

CSV soubor s oddělovačem ";"

```
17;Mezi prvky aktivní bezpečnosti patří;;a) airbag;b) deformovatelná příd;c) ++dobrý výhled z vozidla  
18;Mezi prvky pasivní bezpečnosti patří;;a) účinný brzdový systém;b) ++zádržné systémy;c) aerodynamická stabilita
```



# JAXB – knihovna Java

## Architecture for XML Binding

Poskytuje mechanismy pro marshal (zápis) jav a objektů do XML a unmarshal (čtení) XML do objektů. Můžeme říci, že slouží pro konverzi jav a objektů do XML a naopak.

```
QuestionList questions = new QuestionList();
try{
    File file = new File(fileName);
    JAXBContext jaxbContext = JAXBContext.newInstance(QuestionList.class);
    Marshaller marshaller = jaxbContext.createMarshaller();
    marshaller.setProperty(Marshaller.JAXB_FORMATTED_OUTPUT, true);
    marshaller.marshal(questions, file);
} catch (JAXBException e) {
    e.printStackTrace();
}
```



zápis

```
File xmlFile = new File(fileName);
try{
    JAXBContext jaxbContext = JAXBContext.newInstance(QuestionList.class);
    Unmarshaller unmarshaller = jaxbContext.createUnmarshaller();
    questions = (QuestionList) unmarshaller.unmarshal(xmlFile);
} catch (JAXBException e) {
    e.printStackTrace();
}
```



čtení

# JAXB

nastavení elementů a atributů v class

```
import javax.xml.bind.annotation.XmlRootElement;  
import javax.xml.bind.annotation.XmlElement;  
import javax.xml.bind.annotation.XmlAttribute;  
import javax.xml.bind.annotation.XmlValue;
```

```
@XmlRootElement(name = "Questions")
```

```
public class QuestionList { ... }
```

Tato třída bude kořenovým elementem

```
@XmlElement(name = "Question")
```

```
private ArrayList<Question> questionList = new ArrayList<Question>();
```

Tato proměnná bude vnořeným elementem

```
@XmlAttribute(name = "Id")
```

```
private int qNumber;
```

Tato proměnná bude atributem elementu

# Autoškola

- Program, který testuje uživatele z otázek týkajících se autoškoly
- V souboru typu XML/CSV uchovává jednotlivé otázky
- Uživatel zadá své jméno a příjmení
- Program uživatele otestuje deseti otázkami a poté vytvoří soubor
  - Jehož název bude vypadat následovně: jméno.příjmení.txt
- Po ukončení testu zobrazí uživateli, kolik měl správných a kolik špatných odpovědí a jak dlouho test trval
- Poté umožní různé možnosti výpisu výsledků

# Rozložení do tříd I

- Třída pro komunikaci s uživatelem
  - AutoškolaUI
    - Načítá od uživatele vstupy
    - Volá metody z interface
    - Vypisuje na konzoli výstupy



# Rozložení do tříd II

- Třída pro propojení uživatelského rozhraní a samotné logiky programu:
  - AutoškolaInterface



# Rozložení do tříd III

- Třída pro samotnou logiku programu, nevypisuje nic na konzoli, nepřijímá přímo vstupy z uživatelského rozhraní
  - Třída AutoškolaEditor
    - Pod ní je mnoho dalších tříd, které uživatel vůbec nevidí
    - Propojuje skrz interface metody a vytváří hlavní logiku programu
    - Spolupracuje například s třídami:
      - QuestionList
        - Question
          - Answer
      - Result
      - TestPerson
      - ImportCSV

# AUTOŠKOLA - ČÁST KÓDU

```
public String giveResults(){
    joinSelectedListAndResults();
    int correct = 0, wrong = 0;
    String elapsedTime = "";
    String resultOutput = "";
    String fileContent = "";
    for (int i = 0; i < 10; i++) {
        if(results.get(i).getAnswered().equals(results.get(i).getCorrectAnswer())){
            correct++;
        }
        else{
            wrong++;
        }
    }
    elapsedTime = Duration.between(starts, ends).toString();
    elapsedTime = elapsedTime.replace("PT", "").replace("S", " sec").replace("M", " min a ").replace(".", ",");

    fileContent = String.format("%s;%s;%s\n", correct, wrong, elapsedTime);
    writeResultToFile(this.testPerson.getJmeno(), this.testPerson.getPrijmeni(), fileContent);
    resultOutput = ("Spravne zodpovezenych : " + correct + "\n" + "Spatne zodpovezenych: " + wrong + "\nDelka trvani testu byla: " + elapsedTime);
    return resultOutput;
}
```



DĚKUJI ZA  
POZORNOST