TACS 22/23 - CV Maker - assignment 1

Group members

- Ana Ines Oliveira de Barros <up201806593@fe.up.pt>
- Joao Alexandre Lobo Cardoso <up201806531@fe.up.pt>
- Joao de Jesus Costa <up201806560@fe.up.pt>

Design decisions

Enums' *unset* element

Many times enums are optional. When this happens it's important to include a default *unset* value for the enums. This is a value meaning that the enum instance doesn't have any value set. This is because enums in the generated code can't be set to `null`.

(Sub)section's names

Section can contain subsections (recursively). We enforced that each section's name should be unique within its siblings. This allows tree structures with names like the following:

```
   SectionLayer
    /        \
Section1  Section2
   |          |
 Name       Name
```

But not like the following:

```
  SectionLayer
   /        \
Section   Section
```

Note: Sections can't contain themselves (as subsections). This restriction is applied recursively. This forbids cycles on the tree structure.

Section and content ordering

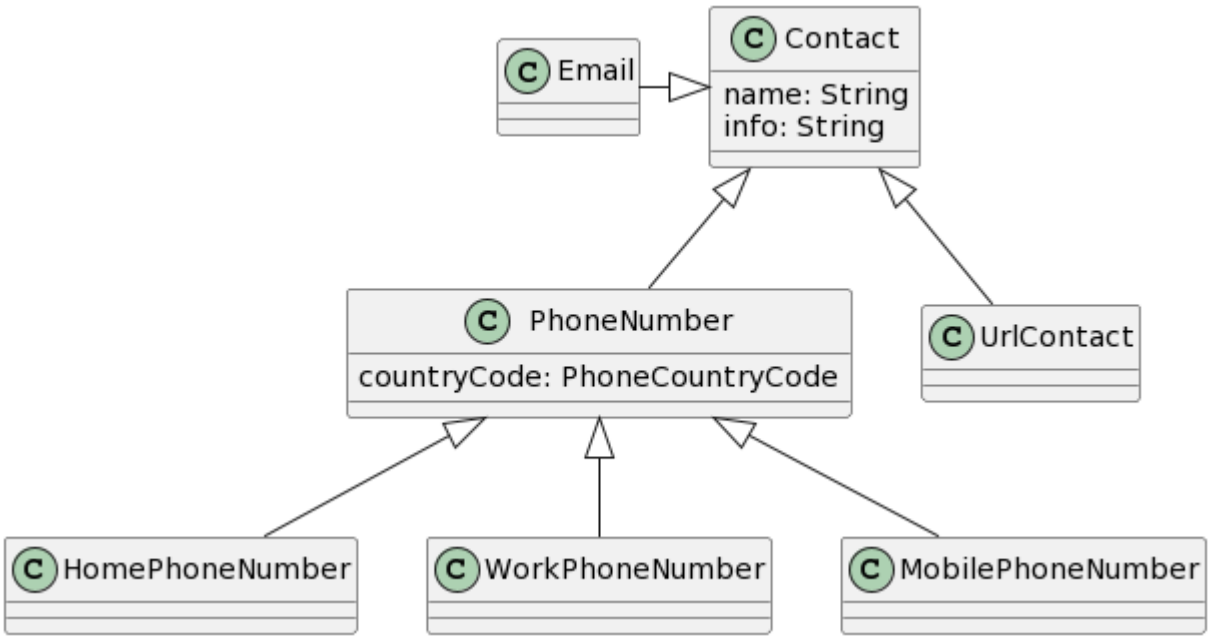
The user is able to order both sections and their content arbitrarily (within their parent). To achieve this, we created an `Orderable` interface, which they implement.

All `Orderable` sections/content have an integer representing their order. These integers are forced to be higher than 0, unique within their parent, and always in sequence (e.g., 1, 2, 3, etc...).

Note: The `IdentificationSection` is the only *non-orderable* section. This is the only mandatory section, and it always appears in the first position (before the rest) of the document.

Contacts

A `Contact` is composed by a name and the contact itself (*info*). We defined some commonly used types of contacts: phone number (which takes a country code), email and url. This way, it is possible to take advantage of the pre-created types or to the create another type of contact that might be missing.




```
@startuml
class Contact {
  name: String
  info: String
}

class PhoneNumber {
  countryCode: PhoneCountryCode
}

Email -|> Contact
PhoneNumber -u-|> Contact
HomePhoneNumber -u-|> PhoneNumber
WorkPhoneNumber -u-|> PhoneNumber
MobilePhoneNumber -u-|> PhoneNumber
UrlContact -u-|> Contact
@enduml
```

Time frames

Usually, work experience is accompanied by a time frame (start date - end date). For this reason, we created the abstract class `TimeFrame` that contains a *startDate*. Then, there is `StartEndTimeFrame` extending `TimeFrame` for cases where the final date of the experience is known, and `TimeFrameDateValid` to be used when the experience is still ongoing.

At first, we wanted to use `null` for the **end date** to signal that it was an *ongoing* time frame, but this caused problems with OCL restrictions: OCL comparisons against `null` were generating semantically incorrect code. We aren't sure why this was happening, but it happened with code snippets from the documentation about `null` comparisons as well.

Concept tables

Concept	Intrinsit Properties	Extrinsic Properties
CV		sectionLayer: One `SectionLayer`

`order` package

Concept	Intrinsit Properties	Extrinsic Properties
`Orderable` *(interface)*	order: int	

`content` package

Concept	Intrinsit Properties	Extrinsic Properties
`Content` *(extends: Orderable)*	name: String description: String *(optional)* url: String *(optional)*	files: Arbitrary number of `File`
`File`	path: String createdDate: Date	type: `FileType` *(Enum)*
`FileType` *(enum)*	`PDF` | `PNG` | `JPG` | `SVG`	
`TimeFrame` *(abstract)*	startDate: Date	
`StartEndTimeFrame` *(extends: TimeFrame)*	endDate: Date	
`OnGoingTimeFrame` *(extends: TimeFrame)*		
`Address`	line1: String *(optional)* line2: String *(optional)* postalCode: String (optional) city: String *(optional)* country: String	

`section` package

Concept	Intrinsit Properties	Extrinsic Properties
`SectionContainer` *(abstract)*		sections: Arbitrary number of `Section`
`Section` *(abstract) (extends: SectionContainer)*	name: String	content: Arbitrary number of `Content`
`OrderedSection` *(extends: Section, Orderable)*		
`SectionLayer` *(extends: SectionLayer)*		identificationSection: One `IdentificationSection`

`workexp` package

Concept	Intrinsit Properties	Extrinsic Properties
`WorkExperienceSection` *(extends: OrderedSection)*		workExperiences: One or more `WorkExperience`
`WorkExperience`	occupation: String employer: String description: String *(optional)*	timeFrame: One `TimeFrame` address: Optional `Address`

`edutrain` package

Concept	Intrinsit Properties	Extrinsic Properties
`EducationTrainingSection` *(extends: OrderedSection)*		educationTraining: One or more `EducationTraining`
`EducationTraining`	title: String organization: String *(optional)* description: String *(optional)* finalGrade: String *(optional)*	fieldsOfStudy: Arbitrary number of `FieldOfStudy` *(enum)* eqf: Optional `EQF` *(enum)*
`FieldOfStudy` *(enum)*	`GENERIC` | `EDUCATION` | `ARTS_HUMANITIES` | ...	
`EQF` *(enum)*	`EQF1` | `EQF2` | `EQF3` | ...	

`identification` package

Concept	Intrinsit Properties	Extrinsic Properties
`IdentificationSection` *(extends: Section)*		One Person
`Person`	firstNames: String lastNames: String *(optional)* title: String *(optional)* dateOfBirth: Date *(optional)* aboutMe: String *(optional)* nationalities: List of String	gender: One `Gender` *(enum)* maritalStatus: One `MaritalStatus` *(enum)* picture: Optional `File` contacts: Arbitrary number of `Contact` addresses: Arbitrary number of `NamedAddress`
`Gender` *(enum)*	`MALE` | `FEMALE` | ...	
`MartialStatus` *(enum)*	`MARRIED` | `DIVORCED` | `SEPARATED` | ...	
`NamedAddress` *(extends: Address)*	name: String	
`HomeAddress` *(extends: NamedAddress)*		

Concept	Intrinsit Properties	Extrinsic Properties
`WorkAddress` *(extends: NamedAddress)*		

`identification.contact` package

Concept	Intrinsit Properties	Extrinsic Properties
`Contact`	name: String info: String	
`Email` *(extends: Contact)*		
`PhoneNumber` *(extends: Contact)*		countryCode: One `PhoneCountryCode` *(enum)*
`PhoneCountryCode` *(enum)*	`UNITED_STATES` | `AFGHANISTAN` | `ALBANIA` | ...	
`HomePhoneNumber` *(extends: PhoneNumber)*		
`WorkPhoneNumber` *(extends: PhoneNumber)*		
`MobilePhoneNumber` *(extends: PhoneNumber)*		
`UrlContact` *(extends: Contact)*		

`skill` package

Concept	Intrinsit Properties	Extrinsic Properties
`SkillSection` *(extends: OrderedSection)*	softSkills: Arbitratry number of String	hardSkills: Arbitrary number of `HardSkill` motherTongues: Arbitrary number of `MotherTongue` secondLanguages: Arbitrary number of `SecondLanguage`
`HardSkill`	name: String proficiency: Float	
`Language` *(abstract)*	name: String	
`MotherTongue` *(extends: Language)*		

Concept	Intrinsit Properties	Extrinsic Properties
`SecondLanguage` *(extends: Language)*		conversation: One `LanguageSkillLevel` *(enum)* reading: One `LanguageSkillLevel` *(enum)* writting: One `LanguageSkillLevel` *(enum)* comprehension: One `LanguageSkillLevel` *(enum)* peerReview: One `LanguageSkillLevel` *(enum)*
`LanguageSkillLevel` *(enum)*	`A1` | `A2` | `B1` | `B2` | `C1` | `C2`	