marketing sunt:

 numărul de plăți cu valoare până în 5000RON (inclusiv); numărul de plăți cu valoare mai mare de 5000RON:

nu au împlinit vârsta majoratului; români născuți în București;

valoarea plăţii.

solicitate de departamentul de marketing. Având în vedere că în trecut s-a constatat faptul că uneori CNP-urile clienților ce ajung la nivelul plăților nu sunt valide și, în plus, datele puse la

Datele de intrare se găsesc într-un fișier csv folosind ca separator de câmpuri;. Fiecare linie din acest fișier reprezintă o plată.

Acestea trebuie ignorate. **Exemplu:** 1930107189436;15324.98 2961022513249;1860 1960608409971;18.1

• Suma plății folosește caracterul "." (punct)

Este posibil ca în fișier să apară linii goale.

ca separator de zecimale.

de următorul tip abstract de date:

bigPayments: Integer,

errors: List of {

line: Integer,

type: Integer

paymentsByMinors: Integer,

- PayMetrics { averagePaymentAmount: BigDecimal, smallPayments: Integer,
- totalAmountCapitalCity: BigDecimal, foreigners: Integer,

```
bigPayments este numărul de plăți cu
valoare mai mare de 5000RON;
paymentsByMinors este numărul de plăți
```

}

}

unde:

```
prin ;;
               1 - CNP-ul nu este valid;
Datele din această structură vor fi serializate într-
un fișier.
Cerințe suplimentare
```

Având în vedere că metricele ce trebuie extrase din aceste date de intrare se pot schimba la un moment dat și faptul că pot apărea alte situații în

care e nevoie să se valideze CNP-uri și să se

validării și interpretării unui CNP într-un API distinct, reutilizabil cu urmatoarea structura:

interface CnpValidator {

* acestuia.

interface CnpParts {

Sex sex():

/**

*/

/**

/**

*/

/**

/**

* Judetul.

Judet judet();

* Data nasterii.

CalDate birthDate();

* Numarul de ordine.

Short orderNumber();

/**

}

extragă informații din ele, se cere implementarea

* Posesorul CNP-ului este cetatean s * @return {@code true} daca este cet contrar */ Boolean foreigner();

Short year(); Byte month(); Byte day(); }

Toate clasele și interfețele trebuie să fie în sub-

Implementarea interfeței CnpValidator se va face într-o clasă cu numele CnpValidatorImpl

 clasele şi interfețele referitoare la validarea CNP-ului vor fi în sub-package-ul cnp, iar cele referitoare la procesarea plătilor în

Cerințe detaliate

pachete ale package-ul

pay;

în acelaşi package.

*

* */

/**

/**

*/

/**

/**

/**

*

*

+ validateCnp(in cnp: String): CnpParts ..

<<implementation Class>:
CnpValidatorImpl

+ process(in paymentsIn: InputStream, in metricsOut: OutputStream) *

Integer type();

*/

}

}

ro.axonsoft.internship21

```
Integer bigPayments();
* Numarul de plati pana in 5000RON i
Integer smallPayments();
* Media sumelor tuturor platilor. Va
```

BigDecimal averagePaymentAmount();

* Totalul platilor efectuate de ceta

* Numarul liniei la care s-a obtinut

BigDecimal totalAmountCapitalCity();

* Erorile de procesare.

Set<PayError> errors();

public interface PayError {

Integer line();

* Tipul erorii:

```
Descrierea formatului CNP este disponibilă aici
In implementare se pot folosii oricare din
bibliotecile open-source, disponibile pe
```

* 0 linie invalida

* 1 pentru CNP invalid

* 2 pentru suma plata invalida</

Livrare Livrarea se va face sub forma unei arhive zip cu numele [nume] - [prenume].zip

Evaluare Se vor evalua respectarea cerințelor cu privire la: validatorul de CNP procesarea tranzacțiilor.

codul vostru sau ați avut nelămuriri cu privire la cerințe, treceți-le într-un

fișier readme inclus în arhivă.

Încărcați arhiva prin intermediul acestei

aplicații accesând secțiunea Trimite.

din acestea sau să copiați codul opensource, caz în care va trebui să mentionati acestea în dependencies sau readme. Puncte bonus vor fi acordate pentru:

- În cadrul unei linii, pe prima poziție se află CNP-ul, iar pe a doua suma plății.
- Date de ieșire Aplicația va furniza datele solicitate de departamentul de marketing și erorile găsite în datele de intrare într-un format structurat descris
- averagePaymentAmount reprezintă media valorilor plăților; smallPayments este numărul de plăți cu valoare până în 5000RON (inclusiv);

vârsta majoratului;

născuți în București;

efectuate de persoane ce nu au împlinit

totalAmountCapitalCity este suma

foreigners este numărul de cetățeni

totală a plăților efectuate de cetățeni români

- străini ce au efectuat plăti; errors este lista de erori, o eroare având proprietățile: line - numărul liniei la care a apărut eroarea; type - tipul erorii; una din valorile: 0 - linia este invalidă, i.e. nu conține două valori separate 2 - suma plății nu este validă.
 - * @param cnp CNP-ul de validat * @return partile componente ale CNP * @throws CnpException daca CNP-ul nu este va */ CnpParts validateCnp(String cnp) thro

* Sexul determinat pe baza primei ci

* Valideza CNP-ul primit ca parametr

} interface CalDate {

```
implementând următoarea interfată:
interface PayMetricsProcessor {
     * Proceseaza platile din {@code paym
     * {@code metricsOutputStream}
     * @param paymentsInputStream
```

* @param metricsOutputStream

Obiectul serializat în OutputStream trebuie să

void process(InputStream paymentsInpu

* Numarul de cetateni straini care a

* Numarul de plati efectuate de catr

* Numarul de plati peste pragul de 5

* @throws IOException

implementeze următoarea interfață:

Integer foreigners();

Integer paymentsByMinors();

interface PayMetrics {

input stream al fisier

output stream al fișie

obiectul conținând met

daca apare o eroare de

In ceea ce privește procesarea plăților, aceasta se va face într-o clasă PayMetricsProcessorImpl

mavenCentral. Diagrama API-ului validatorului de CNP-uri <<interface> <<interface>
CalDate

Diagrama API-ului procesatorului de plăţi

<interface: PayErro

Despre livrare și evaluare

AB CJ (BU pentru Bucuresti:)

{Serializeaza iri OutputStream un obiect implementand PayMetrics:}

<<exception>> IOException

Foloseste o implementare de CnpValidator pentru a valida si extrage partile componente ale CNP-urilor.

CnpValidator

<<nterface>>
PayMetrics

averagePaymentAmount(). BigDecimal bigPayments(): Integer smallPayments(): Integer paymentsByMinors(): Integer totalAmountCapitalCity(): BigDecimal foreigners(): Integer errors(): PayError [*]

continând un director src cu package-urile de surse dacă folosiți biblioteci externe de pe mavenCentral treceți numele lor într-un fișier cu numele dependencies inclus în arhivă. dacă aveți precizări cu privire la

 În mavenCentral există cel puţin o bibliotecă implementând validarea de CNP-

uri. Dacă nu implementați propriul vostru validator de CNP-uri puteți să folosiți una

- cod uşor de citit, respectând paradigmele OOP și bunele practici
- de programare folosirea cu rost a cât mai multor

comentarea codului, în special în

API-uri din JDK folosirea cu rost a unor biblioteci utilitare externe (exceptând implementarea validatorului de CNP) folosirea sintaxei Java 8

format javadoc

scrierea de teste unitare

Mult succes!

- intermediul site-ului.
- media valorilor plăţilor;
- Cerințe funcționale Departamentul de marketing al unei companii care deține un magazin online dorește să obțină o serie de date statistice despre plățile efectuate prin Astfel, metricele solicitate de departamentul de
- Cuprins Cerințe funcționale 2. Cerinte tehnice 3. Date de intrare 4. Date de iesire 5. Cerințe suplimentare 6. Cerințe detaliate
- Cerințe

- Date de intrare
- Cerinte tehnice Să se dezvolte o aplicație Java care să proceseze departamentul comercial și să extragă informațiile
- dispoziție de departamentul comercial vor fi procesate manual, se cere ca pe lângă informațiile solicitate de departamentul de marketing, în datele de ieșire să apară și erorile de procesare.
- datele despre plățile puse la dispoziție de
- Departamentul comercial acceptă să pună la dispoziția programatorului plățile încasate, dar fără a dezvălui identitatea completă a clienților ce au efectuat aceste plăți. Astfel datele ce pot fi puse la dispoziția programtorului sunt: CNP-ul plătitorului și
- plăți.
- numărul de cetățeni străini ce au efectuat
- numărul de plăți efectuate de persoane ce suma totală a plăților efectuate de cetățeni
- 7. Despre livrare și evaluare