

- Klausurvorbereitung PR1 2024:

Aufgabe 1):

Korrigierter Text:

Erstellen Sie eine Klasse "Kind" mit den folgenden **privaten** Attributen:

- (String) name
- (String) id
- (int) birthdate (Zufallszahl zwischen 1 und 366 einschließlich)
- (String) sex
- ArrayList<String> sexList, gefüllt mit den Werten (male, female)

Erstellen Sie einen Konstruktor, in dem "name" und "sex" festgelegt werden. Beachten Sie dabei Folgendes:

- Wenn ein **neues** Geschlecht "sex" vom Benutzer angegeben wird, soll es übernommen und zur Liste "sexList" hinzugefügt werden.

Erstellen Sie einen neuen Konstruktor für den Fall, dass nur der "name", jedoch nicht das Geschlecht des Kindes bekannt ist. In diesem Fall soll dem "sex" ein zufälliges Geschlecht aus der Liste "sexList" zugewiesen werden.

Was Sie nicht wissen, ist, dass diese Kinder außerirdisch sind!

Dort, woher sie kommen, ergibt sich das Datum aus dem reinen Durchzählen der Tage.

Der 31. Januar (irdisch) entspricht 31 (außerirdisch), der 1. Februar (irdisch) entspricht 32, der 3. Februar (irdisch) entspricht 34 usw.

Um dies zu implementieren, erstellen Sie bitte die folgenden Methoden:

- ``num2birth`` (private):
 - o Diese Methode wandelt die Zufallszahl von „birthdate“ in einem irdischen Datum in Form eines Arrays [earthMonth, earthDay] um.
 - o Beachten Sie den Fall, wo „birthdate“ = 366 (Schaltjahr) ist. Was sollte in diesem Fall geändert werden?
 - o Diese Methode soll automatisch bei der Erstellung eines neuen Objekts „Kind“ ausgeführt werden. Dafür sollten Sie zwei neue Instanzvariablen (birthMonth, birthDay) erstellen und ihnen die Werte des Arrays zuweisen.

- o Beachten Sie beim Zugriff auf die Instanzvariablen die Kapselung (Getter und Setter).

- ``name2id`` (private):
 - o Jedes dieser Außerirdischen hat eine ID, die aus der Hexadezimal-Zeichenkette des „name“ besteht (name = Amin -> id = „416d696e“).
 - o Auch diese Methode soll automatisch bei der Erstellung eines neuen Objekts ausgeführt werden. Passen Sie dementsprechend Ihren Konstruktor an.
 - o Achten Sie auch hier auf die Kapselung der Daten.
 - o Tipp: Für die Umwandlung könnten Sie die Methode „Integer.toHexString()“ verwenden.

- ``toString`` (Beispiel: Name: Thanos, ID: 5468616e6f73, Geschlecht: männlich, Geburtsdatum: 363, Geburtstag: 28, Geburtsmonat: 12)

Aufgabe 2):

Erstellen Sie eine Klasse „Family“ mit den folgenden **privaten** Attributen:

- (String) **name**
- (int) **membersNum**
- (int) **address**
- (String) **planet** entweder Mars, Merkur oder Pluto (per Zufall)

Erstellen Sie dazu einen passenden Konstruktor und beachten Sie dabei folgendes:

- Die Namen der Familien auf allen Planeten werden alphabetisch aufsteigend vergeben (Familie A, Familie B, ...).
- Eine außerirdische Familie besteht aus **zwei Eltern** und **mindestens** einem oder mehreren Kindern.
- Beachten Sie, dass jede Familie außer den Kindern aus zwei Eltern besteht.
- Für das Hinzufügen von Kindern erstellen Sie die folgenden Methoden:
 - o **generateKidName()**: Diese Methode gibt einem Kind einen Namen aus vier Buchstaben, z. B. (Avfs oder Jdgh):
 - Der erste Buchstabe wird aus einem zufälligen ASCII-Wert zwischen 65 und 90 ausgewählt.
 - Die restlichen Buchstaben werden zufällig zwischen 97 und 122 gewählt.
 - Die Methode liefert den Namen als Zeichenkette zurück.
 - o **generateKind()**: Diese Methode nimmt ein Geschlecht als Argument und verwendet die Methode generateKidName für die Namensvergabe.
 - Die Methode liefert ein Kind-Objekt zurück.

- In Mars:

- o Muss die **gesamte** Anzahl der Familienmitglieder immer **ungerade** Aufsteigend sein.
- o Adresse wird zufällig zwischen A und Z vergeben werden.
- In **Mercur**:
 - o Hier muss die **gesamte** Anzahl der Familienmitglieder **gerade** absteigend sein (4,6, 8, ...).
 - o Adresse wird in Mercur durch die Summe den ASCII-Werten von dem Namen der Familie z.B. (Familie A -> Adresse = 792)
- In **Pluto**:
 - o Gesamte Anzahl der Familienmitglieder ergibt sich aus der **Summe** von Folgenden:
 - Anzahl die männliche Kinder (ist zufällig zwischen 2 und 5)
 - Anzahl die weibliche Kinder (ist zufällig zwischen 3 und 7)
 - Beachten Sie auch die Eltern!!
 - o Adresse zufällig zwischen **100** und **z** soll als Zahl dargestellt.

Erstellen Sie außerdem die folgenden Methoden:

- Die außerirdischen Adressen sind sehr wichtig und sollten daher verschlüsselt werden, indem sie binär kodiert werden. Erstellen Sie daher die folgende Methode:
 - o **`adress2binaer`**: Diese Methode konvertiert die dezimale Zahl der Adresse **rekursiv** in eine binäre Zahl und gibt den binären Wert als **Zeichenkette** zurück.
- **`toString`**: Beispiel: Name: Familie A, Planet: Mars, Anzahl der Familienmitglieder: 7, Adresse: G
- **`introduceFam`**: Diese Methode stellt alle Familienmitglieder (Kinder) vor.
- **`showAllFam`**: Erstellen Sie dafür eine private statische **`TreeMap<id, Familie>`**, die alle erstellten Familien speichert. Jeder Familie wird automatisch eine inkrementierte **id** als Key zugewiesen.
 - o Passen Sie hierfür den Konstruktor entsprechend an, um jede Instanz direkt in der **`TreeMap`** zu speichern. Überlegen Sie sich, wie Sie dies implementieren können.
 - o Für die Id-Vergabe benötigen Sie keine zusätzliche Methode.
 - o Die Methode **`showAllFam`** soll lediglich die **`TreeMap`** durchlaufen und die **`toString`**-Methode jeder einzelnen Familie ausführen, z. B.:
 - o Name: A-Familie
 - o Planet: Pluto
 - o Anzahl der Familienmitglieder: 10
 - o Adresse: 115
- **`getFamByName`**: Diese Methode liefert eine Familie anhand des Namens zurück oder **`null`**, wenn die Familie nicht existiert.
- **`showFamByName`**: Diese Methode ruft die **toString**-Methode einer Familie anhand ihres Names.
- **`addKindToFam`**: Diese Methode fügt ein neues Kind zu einer Familie anhand des Namens dieser Familie hinzu (verwenden Sie dafür die Methode **`getFamByName`**).
 - o Sie sollte entweder "true" zurückliefern, wenn das Kind erfolgreich hinzugefügt wurde, oder "false", wenn nicht.
- **`addKinderAllFam`**: Diese Methode nimmt die Anzahl der hinzuzufügenden Kinder (n) als Argument und fügt n-Kinder zu allen Familien.

Erstellt von Amin Bkdaliya, 05.01.24

o Dafür können Sie die Methoden `addKinToFam` und `generateKidName` verwenden

Erstellen Sie außerdem eine statische Methode `run()` als Verwaltungsmethode.

Rufen Sie diese Methode von einem Objekt der Klasse Familie über eine `main`-Methode einer Klasse Ihrer Wahl auf. Orientieren Sie sich dabei an der folgenden Ausgabe.

```
public class testing {  
    public static void main(String[] args) {  
        Family family1 = new Family();  
        Family family2 = new Family();  
        Family family3 = new Family();  
        family3.run();  
    }  
}
```

```
Willkommen im All:  
List möglicher Befehle:(Bitte wählen Sie eine Zahl)  
1. Zeig alle Familien  
2. Zeig eine Familie  
3. Eine Familie vorstellen  
4. Füge ein Kind zu einer Familie  
5. Füge n-Kinder zu allen Familien  
6. Kodiere die Adresse in binäre  
7. Programm beenden  
Input:1  
  
Name: A-Familie  
Planet: Mars  
Anzahl der Familienmitglieder: 3  
Adresse: 72  
  
Name: B-Familie  
Planet: Pluto  
Anzahl der Familienmitglieder: 9  
Adresse: 101  
  
Name: C-Familie  
Planet: Merkur  
Anzahl der Familienmitglieder: 8  
Adresse: 807  
  
List möglicher Befehle:(Bitte wählen Sie eine Zahl)  
1. Zeig alle Familien  
2. Zeig eine Familie  
3. Eine Familie vorstellen  
4. Füge ein Kind zu einer Familie  
5. Füge n-Kinder zu allen Familien  
6. Kodiere die Adresse in binäre  
7. Programm beenden  
Input:2  
  
Bitte geben Sie den Buchstaben der Familie ein:
```

```
Bitte geben Sie den Buchstaben der Familie ein:
b
Name: B-Familie
Planet: Pluto
Anzahl der Familienmitglieder: 9
Adresse: 101

List möglicher Befehle:(Bitte wählen Sie eine Zahl)
1.Zeig alle Familien
2.Zeig eine Familie
3.Eine Familie vorstellen
4.Füge ein Kind zu einer Familie
5.Füge n-Kinder zu allen Familien
6.Kodiere die Adresse in binäre
7.Programm beenden
Input:3

Bitte geben Sie den Buchstaben der Familie ein:
a
A-Familie begrüßt euch, unsere Familie besteht aus dem Vater,
der Mutter und folgenden Kindern

Mein Name ist: Zwqm, ID: 5a77716d, Sex: female, Birth Date:
325, Birthday: 21, Birthmonth: 11

Input:4

Bitte geben Sie den Buchstaben der Familie:
a
Ein Kind wurde erfolgreich hinzugefügt.
Name: A-Familie
Planet: Mars
Anzahl der Familienmitglieder: 4
Adresse: 72

Input:3

Bitte geben Sie den Buchstaben der Familie ein:
a
A-Familie begrüßt euch, unsere Familie besteht aus dem Vater,
die Mutter und folgende Kinder

Mein Name ist: Zwqm, ID: 5a77716d, Sex: female, Birth Date:
325, Birthday: 21, Birthmonth: 11

Mein Name ist: Chyl, ID: 4368796c, Sex: female, Birth Date:
308, Birthday: 4, Birthmonth: 11

Input:5

Bitte geben Sie die Anzahl den Kindern:
2
```

```
Einfügen war erfolgreich
Name: A-Familie
Planet: Mars
Anzahl der Familienmitglieder: 6
Adresse: 72

Name: B-Familie
Planet: Pluto
Anzahl der Familienmitglieder: 11
Adresse: 101

Name: C-Familie
Planet: Merkur
Anzahl der Familienmitglieder: 10
Adresse: 807

Input:3

Bitte geben Sie den Buchstaben der Familie:
b
B-Familie begrüßt euch, unsere Familie besteht aus dem Vater,
der Mutter und folgenden Kindern:

Mein Name ist: Kxit, ID: 4b786974, Sex: male, Birth Date: 359,
Birthday: 25, Birthmonth: 12

Mein Name ist: Ndge, ID: 4e646765, Sex: male, Birth Date: 227,
Birthday: 15, Birthmonth: 8

Mein Name ist: Gujt, ID: 47756a74, Sex: female, Birth Date:
276, Birthday: 3, Birthmonth: 10

Mein Name ist: Inch, ID: 496e6368, Sex: female, Birth Date:
210, Birthday: 29, Birthmonth: 7

Mein Name ist: Itbg, ID: 49746271, Sex: female, Birth Date:
130, Birthday: 10, Birthmonth: 5

Mein Name ist: Qnsf, ID: 516d7366, Sex: female, Birth Date:
335, Birthday: 1, Birthmonth: 12

Mein Name ist: Yqih, ID: 59676968, Sex: female, Birth Date:
266, Birthday: 23, Birthmonth: 9

Mein Name ist: Lrzg, ID: 4c727a67, Sex: male, Birth Date: 305,
Birthday: 1, Birthmonth: 11

Mein Name ist: Gxii, ID: 47786969, Sex: female, Birth Date:
28, Birthday: 28, Birthmonth: 1

Input:6
```

```
Encryption mode activated

Input:1

Name: A-Familie
Planet: Mars
Anzahl der Familienmitglieder: 6
Adresse: 1001000

Name: B-Familie
Planet: Pluto
Anzahl der Familienmitglieder: 11
Adresse: 1100101

Name: C-Familie
Planet: Merkur
Anzahl der Familienmitglieder: 10
Adresse: 1100100111

Input:7
```