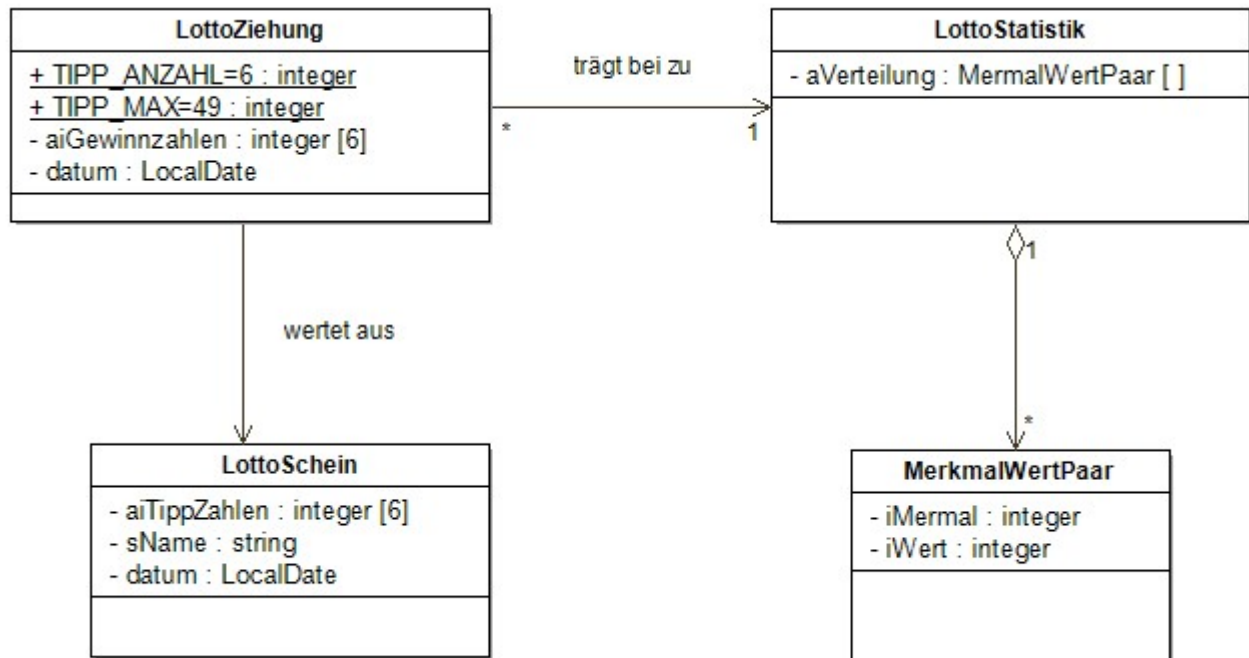


2.6. Simulation Lotto-Tipp „objektorientiert“

Die Lotto Simulation soll jetzt objektorientiert implementiert werden. Das folgende Klassendiagramm zeigt die erforderlichen Fachklassen.



- Erstellen Sie in Eclipse ein neues Package für das Programm
- Holen Sie von Moodle die Zip Datei `DatenLottoSimulation` und entpacken diese.
- Die Startklasse `LottoStart` kann aus `DatenLottoSimulation` kopiert und in das Package der Anwendung eingebunden werden. (Einfach die Datei `LottoStart.java` in das Package in Eclipse ziehen.)
- Die Beschreibung der notwendigen Objektmethoden findet sich ebenfalls in `DatenLottoSimulation`. Dort in `LottoDok` die Datei `index.html` doppelklicken. In der Dokumentation sind die Objektmethoden der einzelnen Klassen beschrieben.
- Implementieren Sie das Klassendiagramm, indem Sie die angegebenen Attribute der Klassen anlegen und die in der Dokumentation beschriebenen Methoden in den Klassen programmieren.

```
public class LottoSchein
{
    // Anfang Attribute
    private int[] aiTippZahlen;
    private String sName;
    private LocalDate datum;
    // Ende Attribute
    // Anfang Methoden
    * Initialisiert sName mit "-leer-"; []
    public LottoSchein(int iTippAnzahl)
    {
        this.aiTippZahlen = new int[iTippAnzahl];
        this.sName = "-leer-";
        this.datum = LocalDate.now();
    }

    * @return Die Tipp-Zahlen im int-Array (sortiert)[]
    public int[] getTippZahlen()[]

    * Tipp-Zahlen können im int-Array übergeben werden[]
    public void setTippZahlen(int[] aiTippZahlen)[]

    public String getSName()[]

    public void setSName(String sName)[]

    public LocalDate getDatum()[]

    public void setDatum(LocalDate datum)[]

    * Ein Lotto-Tipp wird zufällig ermittelt und gespeichert;[]
    public void generiereTipp()
    {
        int iTipp;
        Random zufall = new Random();
        int iZahl = 0;

        while (iZahl < this.aiTippZahlen.length)
        {
            iTipp = zufall.nextInt(LottoZiehung.TIPP_MAX) + 1;
            if (ArrayTools.sucheSequenziell(this.aiTippZahlen, 0, iZahl - 1, iTipp) == -1)
            {
                this.aiTippZahlen[iZahl] = iTipp;
                iZahl++;
            }
        }
        Arrays.sort(this.aiTippZahlen);
    }

    /* (non-Javadoc)[]
    * Ein Lotto-Tipp wird in folgender Form als String[]
    public String toString()
    {
        StringBuilder sbAlles = new StringBuilder("Tipp von:");

        sbAlles.append(this.sName);
        sbAlles.append(" ");
        sbAlles.append(Arrays.toString(this.aiTippZahlen));

        return sbAlles.toString();
    }
    // Ende Methoden
}
```

```
public class LottoZiehung
{
    // Anfang Attribute
    public static final int TIPP_ANZAHL = 6;
    public static final int TIPP_MAX = 49;
    private LocalDate datum;
    private int[] aiGewinnzahlen;
    // Ende Attribute
    // Anfang Methoden
    * Erzeugt das int-Array für die Gewinnzahlen;
    public LottoZiehung()
    {
        int iTipp;
        int iZahl = 0;
        aiGewinnzahlen = new int[TIPP_ANZAHL];
        Random zufall = new Random();

        this.datum = LocalDate.now();

        while (iZahl < this.aiGewinnzahlen.length)
        {
            iTipp = zufall.nextInt(TIPP_MAX) + 1;
            if (ArrayTools.sucheSequenziell(this.aiGewinnzahlen, 0, iZahl - 1, iTipp) == -1)
            {
                this.aiGewinnzahlen[iZahl] = iTipp;
                iZahl++;
            }
        }
        Arrays.sort(this.aiGewinnzahlen);
    }

    public LocalDate getDatum()
    {
        return datum;
    }

    public void setDatum(LocalDate datum)
    {
        this.datum = datum;
    }

    public int[] getAiGewinnzahlen()
    {
        return aiGewinnzahlen;
    }

    public void setAiGewinnzahlen(int[] aiGewinnzahlen)
    {
        this.aiGewinnzahlen = aiGewinnzahlen;
    }

    * Für den im Parameter gegebenen Lottoschein wird die Anzahl der Treffer
    public int ermittleTreffer(LottoSchein einLottoSchein)
    {
        int iRichtige = 0;

        for (int iTipp : einLottoSchein.getTippZahlen())
        {
            if (Arrays.binarySearch(this.aiGewinnzahlen, iTipp) >= 0)
                iRichtige++;
        }

        return iRichtige;
    }

    /* (non-Javadoc)
    * Eine Lotto-Ziehung wird in folgender Form als String
    public String toString()
    {
        StringBuilder sbAlles = new StringBuilder("Ziehung vom : ");

        sbAlles.append(datum.format(DateTimeFormatter.ofPattern("dd/MM/yy")));
        sbAlles.append(" ");

        sbAlles.append(Arrays.toString(this.aiGewinnzahlen));

        return sbAlles.toString();
    }
    // Ende Methoden
}
```

```
public class MerkmalWertPaar implements Comparable<MerkmalWertPaar>
{
    private int iMerkmal;
    private int iWert;

    * @param iMerkmal[]
    public MerkmalWertPaar(int iMerkmal, int iWert)
    {
        this.iMerkmal = iMerkmal;
        this.iWert = iWert;
    }
    public int getiMerkmal(){}
    public void setiMerkmal(int iMerkmal){}
    public int getiWert(){}
    public void setiWert(int iWert){}
    public void inkrementWert()
    {
        this.iWert++;
    }

    /* (non-Javadoc)[]
    /**
     * compareTo Methode definiert die Ordnung der MerkmalWertPaare
     * siehe 20_OOP_Arrays - Sortierung
     */
    @Override
    public int compareTo(MerkmalWertPaar anderesPaar)
    { // Das erste Minuszeichen sorgt für absteigende Sortierung
        return -(this.getiWert() - anderesPaar.getiWert());
    }
}
```

```
public class LottoStatistik
{
    MerkmalWertPaar[] aVerteilung;

    * Erzeugt ein MerkmalWertPaar Array mit Länge iAnzahlWerte zu Speichern
    public LottoStatistik(int iAnzahlWerte)
    {
        this.aVerteilung = new MerkmalWertPaar[iAnzahlWerte];
        // Initialisiere die Verteilung mit Werten und der Häufigkeit 0
        for (int i = 0; i < this.aVerteilung.length; i++)
        {
            this.aVerteilung[i] = new MerkmalWertPaar(i+1, 0);
        }
    }

    public MerkmalWertPaar[] getaVerteilung()
    {
        return this.aVerteilung;
    }

    * Die im Parameter gegebene LottoZiehung wird der Häufigkeitsverteilung h
    public void addiereZiehung(LottoZiehung eineZiehung)
    {
        int[] ailiste = eineZiehung.getAiGewinnzahlen();
        for (int i : ailiste)
        {
            if (i <= this.aVerteilung.length)
            {
                this.aVerteilung[i-1].inkrementWert();
            }
        }
    }

    * Eine Kopie der gespeicherten Häufigkeitsverteilung wird sortiert (entsp
    public MerkmalWertPaar[] getNachHäufigkeit()
    {
        MerkmalWertPaar[] aSortiert;

        aSortiert = Arrays.copyOf(aVerteilung, aVerteilung.length);
        Arrays.sort(aSortiert);
        return aSortiert;
    }
}
```