

```

public static void main (String[] args)
{
    hplanszaGraf.DrawRectangle(ps_Dlugopis, PS_X -
        PS_Margines, PS_X - PS_Margines, PS_Y / 2, PS_X /
        PS_Widoczny);
}
public static void main (String[] args)
{
    ps_Dlugopis.Dispose();
    BufferedReader File_reader = new BufferedReader (new InputStreamReader
    String text;
    while (!text=file_reader.readLine(file_contents)).endsWith()) System.
    int a;
    public override void PS_Wymaz()
    {
        for (int i=0; i<PS_Widoczny)
        {
            a++;
            for (int j=0; j<PS_Wysokosc)
            {
                x[j] = '\0';
                ps_Dlugopis = new Pen(hForm1.PS_img Backg
                this.PS_grubosc);
                ps_Dlugopis
            }
            z[a+j]=x[j];
        }
    }
}

```

Grundlagen Programmierung 1

PB6 – Listen, Tupel und Formatierung der Ausgabe

Hausaufgaben und Fragen

/ Ergebnis Präsentation & Besprechung

/ Fragen?

Listen

Eigenschaften
Operatoren
Funktionen

Eigenschaften

- / Sequenz von Objekten
- / Im Gegensatz zu String veränderbar
- / Listen sind in eckigen Klammern angegeben

`z = [3, 6, 12.5, -8]`

`s = ['Hamburg', 'Augsburg', 'Berlin']`

- / Index gleich wie bei String, `z[0] = ?`
- / Mehrdimensionale Listen

`md = [['Paris', 'FR', 3500000], ['Rom', 'IT', 4200000]]`

Operatoren

- / Welche Operatoren kennen Sie von Strings noch?
- / Ähnliche Operatoren auch für Listen zusammensetzen: +, *
- / Listen durchgehen mit for-Schleifen:

```
fr = ["Paris", "Lyon", "Marseille"]  
it = ["Rom", "Pisa"]
```

```
stadtliste = fr + it * 2  
for stadt in stadtliste[3:6]:  
    print(stadt)
```

Funktionen

/ Listenänderung mittel Selektion:

- / `fr = ["Paris", "Lyon", "Marseille"]`
- / `fr[2] = "Lens"` → Ersetzen eines Elements durch Element
- / `fr[1:3] = fr = ["Nancy", "Metz", "Gap"]` → Ersetzen eines Teilbereichs durch Liste
- / `del fr[3:]` → Entnehmen eines Teilbereichs
- / `fr[0] = fr = ["Paris-Nord", "Paris-Sud"]` → Ersetzen eines Elements durch eine Liste

/ Listenänderung mittels Funktionen:

- / `insert()` → Element einsetzen
- / `sort()` → Liste sortieren
- / `reverse()` → Liste umdrehen
- / `remove()` → Element entfernen
- / `append()` → Element am Ende hinzufügen
- / `count()` → Anzahl bestimmter Elemente suchen
- / `index()` → Bestimmtes Element suchen

Einzelaufgabe – 30min

- / Erstelle eine Liste von Zahlen. Zum Beispiel: zahlen = [3, 7, 1, 9, 4, 6].
- / Schreibe eine Funktion `summe(liste)`, die als Eingabe eine Liste von Zahlen erhält und die Summe aller Zahlen in der Liste zurückgibt.
- / Schreibe eine Funktion `groesste_zahl(liste)`, die als Eingabe eine Liste von Zahlen erhält und die größte Zahl in der Liste zurückgibt.
- / Schreibe eine Funktion `liste_sortieren(liste)`, die als Eingabe eine Liste von Zahlen erhält und die Zahlen in aufsteigender Reihenfolge sortiert zurückgibt.
- / Schreibe eine Funktion `entferne_doppelte(liste)`, die als Eingabe eine Liste von Zahlen erhält und eine neue Liste zurückgibt, in der doppelte Elemente entfernt wurden, während die Reihenfolge erhalten bleibt.

Tupel

Eigenschaften
Operationen
Tupel entpacken

Eigenschaften

- / Unterschied zu Liste → Tupel kann nicht verändert werden
- / Tupel können ohne Klammern oder mit runden Klammern () erzeugt werden
- / Alle gleichen Operatoren wie bei Listen gültig, nur nicht diejenigen, die Veränderungen vornehmen!
- / Welche wären diese Funktionen?

Operationen

/ Tupel erzeugen

```
z = (3, 6, -8, 5.5)
print("Tupel 1:", z)
```

```
z = 6, 8, -3
print("Tupel 2:", z)
```

```
# mehrdimensionales Tupel
x = (("Paris", "Fr", 35000000),
     ["Rom", "It", 42000000])
print("Mehrdimensionales Tupel:")
print(x)
```

/ Tupel ersetzen

Wieso führt die nachfolgend erste Codezeile zu einem Fehler?

```
x[0][0] = "Lyon"
x[1][0] = "Pisa"
```

/ Tupel Iterieren und Zuweisen

```
# Tupel bei for-Schleife
for i in 4, 5, 12:
    print("i:", i)
```

```
# Tupel zuweisen
x,y = 2,18
print("x:", x, "y:", y)
```

Tupel entpacken

- / Wir haben gelernt, dass Tupel:
 - / unveränderbare Werte speichert
 - / eine Mehrfachzuweisung vornehmen kann
- / Besonderheiten gibt es dennoch!
- / Gemeinsames erarbeiten des Kapitels 4.4.3 (Theis) ab S. 120

Gruppenarbeit (2 Personen) – 20min

/ Aufgabenstellung:

- / Erstellen Sie eine Liste mit den drei Schweizer Kantone Zürich Basel und Luzern
- / Ersetzen Sie Luzern mit Genf
- / Fügen Sie weitere vier Städte ihrer Wahl hinzu
- / Erstellen Sie ein Tupel mit all ihnen bekannten Datentypen
- / Gegeben sei der Tupel $\text{tuple} = 2, 4, 5, 6, 2, 3, 4, 4, 7$. Geben Sie aus, wie oft die Nummer 4 vorkommt

Einzelarbeit – 40min

- / Erstelle eine Liste von Tupeln, wobei jedes Tupel den Namen einer Person und ihren Geburtstag enthält. Zum Beispiel: `personen = [("Max", "01.01.2000"), ("Anna", "15.03.1998"), ("Tom", "10.07.2005")]`.
- / Schreibe eine Funktion `aeltere_personen(personen)`, die als Eingabe eine Liste von Tupeln erhält und die Namen der Personen zurückgibt, die vor dem Jahr 2000 geboren wurden.
- / Schreibe eine Funktion `neue_liste(personen)`, die als Eingabe eine Liste von Tupeln erhält und eine neue Liste von Tupeln zurückgibt, in der die Namen und Geburtstage der Personen vertauscht sind. Zum Beispiel: Aus `("Max", "01.01.2000")` wird `("01.01.2000", "Max")`.
- / Schreibe eine Funktion `doppelte_geschenke(personen)`, die als Eingabe eine Liste von Tupeln erhält und prüft, ob es Personen mit demselben Geburtstag gibt. Diese Funktion soll eine Liste der Geburtstage zurückgeben, an denen mehr als eine Person Geburtstag hat.

Ausgabe und Formattierung

Weiterführende Erkenntnisse zur Funktion `print()`

Funktion `print()`

- / Bereits kennengelernt, für die Ausgabe gedacht
- / Sie kann mehrere Variablen gleichzeitig ausgeben (Komma separiert)
- / Beispiel im Buch Seite 151, Kap. 5.2.1 Theis

Formatierung

- / Python hat die Möglichkeit, Zahlen und Zahlenketten einheitlich formatiert auszugeben
- / Sie dient zur Darstellung in Tabellenform oder weiteren erwünschten Formen

String-Interpolation

```
name = "Alice"
```

```
alter = 30
```

```
print(f"Mein Name ist {name} und ich bin {alter} Jahre alt.")
```

str.format()-Methode

```
name = "Bob"
```

```
alter = 25
```

```
print("Mein Name ist {} und ich bin {} Jahre alt.".format(name, alter))
```

Prozentformatierung (%-Operator)

```
name = "Charlie"
```

```
alter = 40
```

```
print("Mein Name ist %s und ich bin %d Jahre alt." % (name, alter))
```

Mehrzeilige Strings mit Zeilenumbrüchen

```
name = "David"  
alter = 35  
print("Mein Name ist\n{}\nund ich bin\n{} Jahre alt.".format(name,  
alter))
```

Manuelle Konvertierung in Strings

```
name = "Eva"
```

```
alter = 28
```

```
print("Mein Name ist " + name + " und ich bin " + str(alter) + " Jahre  
alt.")
```

Einzelübung (20min)

/ Aufgabenstellung

/ Bestehend ihres Umrechners von Pfund und Kg, sollten sie nun eine Ausgabe erzeugen, welches eine Tabelle darstellt. Es sollten jeweils 2 Nachkommastellen für Kg und Pfund angezeigt werden und rechtsbündig ausgerichtet sein. Zahlen sollten in 3er Schritte zwischen 1 und 50 erfolgen.

MPH	KMH
1.00	1.61
4.00	6.44
7.00	11.27
10.00	16.09
13.00	20.92
16.00	25.75
19.00	30.58
22.00	35.41
25.00	40.23
28.00	45.06
31.00	49.89
34.00	54.72
37.00	59.55
40.00	64.37
43.00	69.20
46.00	74.03
49.00	78.86

Hausaufgaben

/ PDF:

Hausaufgaben_PB6_Listen_Tupel_Localt
ime.pdf