

Maturitná príprava

Prehľad, ťahák a postup pri riešení úloh

Tieto poznámky sú sprievodcom témou Maturitná príprava. Obsahujú **ťahák** najdôležitejších príkazov Pythonu na jednom mieste, **programátorské vzory**, ktoré sa v úlohách opakujú stále dokola, a **riešené úlohy** v štýle praktickej časti maturity. Podrobný výklad každej oblasti nájdeš v poznámkach k jednotlivým témam (Reťazce, Zoznamy, Cykly, Výnimky, Grafika v tkinteri...).

Čo nájdeš v tejto téme

Lekcia	Čo precvičuje	Poznámky na zopakovanie
Reťazce	indexy, rezy, metódy, prechádzanie textu, šifry	Reťazce, Cykly
Zoznamy	náhodné zoznamy, hľadanie, triedenie, štatistika	Zoznamy, Algoritmy, Množiny
Cyklus while	opakovanie s podmienkou, postupnosti, prevody	Cykly, Operátory
Mix úloh	kombinácia všetkého, rozklad na funkcie	Funkcie, Modulárny program, Algoritmické úlohy
Grafika v tkinteri	kreslenie, udalosti, animácia, ovládacie prvky	Grafika v tkinteri
Výnimky	bezpečný vstup, ošetrovanie chýb	Výnimky, Práca so súborami

1 Ako riešiť praktickú úlohu

1. **Prečítaj zadanie dvakrát** a podčiarkni: čo je na vstupe, čo má byť na výstupe a v akom presnom tvare.
2. **Vyskúšaj príklad ručne** – na papieri, s malými číslami.
3. **Rozdeľ úlohu na časti** (načítanie – výpočet – výpis) a každú zapíš ako funkciu.
4. **Programuj po kúskoch** a priebežne spúšťaj. Pomocné výpisy na konci odstráň.
5. **Otestuj hraničné prípady**: prázdny vstup, jeden prvok, nula, záporné čísla, rovnaké hodnoty.
6. **Skontroluj tvar výstupu** – medzery, veľké a malé písmená, počet desatinných miest.

1.1 Ako sa úlohy na Mente kontrolujú

- Testy porovnávajú **iba výstup** príkazu `print` – výzvy v `input("...")` sa pri testoch nevypisujú.
- Ak zadanie predpisuje **funkcie** (napr. `vygeneruj()`, `spracuj(zoznam)`), testy ich volajú priamo. Hlavný program (načítanie a výpis) vtedy patrí pod `if __name__ == "__main__":`, aby sa pri testovaní funkcií nespúšťal.
- Pri náhodných úlohách sa generovanie a spracovanie oddeľujú do rôznych funkcií: spracovanie sa dá otestovať s pevnými údajmi a pri generátore sa kontrolujú vlastnosti (počet, rozsah, rôznosť), nie konkrétne čísla.

```

import random

def vygeneruj(n):
    return [random.randint(1, 100) for _ in range(n)]

def spracuj(zoznam):
    return max(zoznam) - min(zoznam)

if __name__ == "__main__":
    cisla = vygeneruj(10)
    print(cisla)
    print("Rozpätie:", spracuj(cisla))

```

2 Ťahák: Python v kocke

2.1 Vstup, výstup, typy

Zápis	Význam
<code>x = input("Výzva: ")</code>	načíta riadok ako reťazec
<code>n = int(input()), x = float(input())</code>	načíta celé / desatinné číslo
<code>a, b = map(int, input().split())</code>	dve celé čísla na jednom riadku
<code>cisla = [int(c) for c in input().split()]</code>	zoznam čísel z jedného riadku
<code>print(a, b, sep=" ", end="")</code>	výpis s oddeľovačom, bez nového riadku
<code>print(f"{x:.2f} {meno:<10} {n:>5}")</code>	2 desatinné miesta, zarovnanie doľava a doprava
<code>int("42"), str(42), float("2.5"), round(x, 2)</code>	prevody a zaokrúhlenie

2.2 Operátory

Operátor	Význam	Príklad
<code>/, //, %, **</code>	delenie, celočíselné delenie, zvyšok, mocnina	<code>17 // 5</code> je 3, <code>17 % 5</code> je 2
<code>==, !=, <, <=, >, >=</code>	porovnania	<code>1 <= x <= 10</code>
<code>and, or, not</code>	logické spojky	<code>x % 2 == 0 and x > 0</code>
<code>in, not in</code>	výskyt v reťazci, zozname, množine, kľúčoch slovníka	<code>"a" in slovo</code>
<code>+=, -=, *= ...</code>	skrátene priradenie	<code>sucet += x</code>

2.3 Riadenie behu programu

```

if podmienka:
    ...
elif ina_podmienka:
    ...
else:
    ...

for i in range(od, do, krok):      # do už nepatrí
    ...
for prvok in zoznam_alebo_text:
    ...
while podmienka:
    ...
    break      # ukončí cyklus
    continue   # skočí na ďalšie opakovanie

def funkcia(a, b=0):              # b má predvolenú hodnotu
    return a + b

```

2.4 Reťazce

Zápis	Význam
<code>s[0], s[-1], s[2:5], s[::-1]</code>	prvý, posledný znak, rez, obrátený text
<code>len(s), s.count("a"), s.find("x")</code>	dĺžka, počet výskytov, index (alebo -1)
<code>s.upper(), s.lower(), s.strip(), s.replace(a, b)</code>	nový upravený reťazec
<code>s.split(), s.split(";"), " ".join(zoznam)</code>	rozdelenie na zoznam, spojenie
<code>s.isdigit(), s.isalpha(), s.isupper()</code>	testy znakov
<code>ord("a"), chr(97)</code>	kód znaku a znak z kódu

2.5 Zoznamy, n-tice, množiny, slovníky

Zápis	Význam
<code>z.append(x), z.insert(i, x), z.pop(), z.remove(x)</code>	pridanie, vloženie, odobratie
<code>z.sort(), sorted(z, reverse=True), z.reverse()</code>	triedenie (na mieste / nový zoznam)
<code>len(z), sum(z), min(z), max(z), z.count(x), z.index(x)</code>	základné funkcie
<code>[x * 2 for x in z if x > 0]</code>	generátorový zápis
<code>a, b = b, a</code>	výmena hodnôt (n-tica)
<code>set(z), a & b, a b, a - b</code>	množina, prienik, zjednotenie, rozdiel
<code>d[k] = v, d.get(k, 0), k in d, d.items()</code>	slovník: zápis, bezpečné čítanie, prechádzanie

2.6 Súbory, výnimky, moduly

```
with open("data.txt", encoding="utf-8") as f:      # čítanie
    for riadok in f:
        casti = riadok.strip().split(";")

with open("vystup.txt", "w", encoding="utf-8") as f: # zápis
    f.write("text\n")

try:
    n = int(input())
except ValueError:
    print("To nie je číslo.")

import random, math
random.randint(1, 6); random.choice(z); random.shuffle(z); random.seed(1)
math.sqrt(2); math.pi; math.floor(x); math.ceil(x); math.gcd(a, b)
```

3 Programátorské vzory

Väčšina úloh sa skladá z niekoľkých stále rovnakých „stavebných kameňov“. Keď ich poznáš, riešenie skladáš ako stavebnicu.

3.1 Načítavanie, kým nepríde ukončovacia hodnota

```
cisla = []
while True:
    x = int(input())
    if x == -1:          # ukončovacia hodnota do výsledku nepatrí
        break
    cisla.append(x)
print(cisla)
```

Výstup:

```
12
7
30
-1
[12, 7, 30]
```

3.2 Počítadlo, súčet a priemer s podmienkou

```
z = [12, -4, 7, 0, 15, -9, 8]
kladne = [x for x in z if x > 0]
print("počet kladných:", len(kladne))
print("súčet kladných:", sum(kladne))
print("priemer kladných:", round(sum(kladne) / len(kladne), 2) if kladne else "-")
```

Výstup:

```
počet kladných: 4
súčet kladných: 42
priemer kladných: 10.5
```

3.3 Maximum a jeho pozícia

```
z = [12, 47, 3, 47, 21]
najvacsie = max(z)
pozicie = [i for i in range(len(z)) if z[i] == najvacsie]
print(najvacsie, "na pozíciách", pozicie)
```

Výstup:

```
47 na pozíciách [1, 3]
```

3.4 Existuje? Platí pre všetky?

```
z = [4, 8, 15, 16, 23, 42]
print("existuje nepárne:", any(x % 2 == 1 for x in z))
print("všetky kladné:", all(x > 0 for x in z))

je_tam = False # to isté s príznakom
for x in z:
    if x > 40:
        je_tam = True
        break
print("existuje väčšie ako 40:", je_tam)
```

Výstup:

```
existuje nepárne: True
všetky kladné: True
existuje väčšie ako 40: True
```

3.5 Frekvencia výskytov

```
z = [3, 1, 3, 2, 3, 1]
pocety = {}
for x in z:
    pocety[x] = pocety.get(x, 0) + 1
print(pocety)
najcastejsie = max(pocety, key=pocety.get)
print("najčastejšie:", najcastejsie, pocety[najcastejsie], "krát")
```

Výstup:

```
{3: 3, 1: 2, 2: 1}
najčastejšie: 3 3 krát
```

3.6 Bez opakovania, v pôvodnom poradí

```
z = [5, 3, 5, 1, 3, 8]
bez_opakovania = []
for x in z:
    if x not in bez_opakovania:
        bez_opakovania.append(x)
print(bez_opakovania)
```

Výstup:

```
[5, 3, 1, 8]
```

3.7 Bezpečné načítanie čísla

```
while True:
    try:
        n = int(input("Zadaj celé číslo: "))
        break
    except ValueError:
        print("To nie je celé číslo.")
print("Mám", n)
```

Výstup:

```
Zadaj celé číslo: dva
To nie je celé číslo.
Zadaj celé číslo: 7
Mám 7
```

3.8 Náhodný zoznam

```
import random

z = [random.randint(1, 20) for _ in range(12)]
print(z)
print("párne:", [x for x in z if x % 2 == 0])
print("zoradený bez opakovania:", sorted(set(z)))
```

Výstup:

```
[20, 9, 12, 17, 1, 15, 8, 2, 6, 4, 12, 16]
párne: [20, 12, 8, 2, 6, 4, 12, 16]
zoradený bez opakovania: [1, 2, 4, 6, 8, 9, 12, 15, 16, 17, 20]
```

3.9 Skladanie nového textu

```
text = "Maturita z informatiky 2026"
novy = ""
for znak in text:
    if znak.isdigit():
        novy += "#"
    elif znak in "aeiouy":
        novy += znak.upper()
    else:
        novy += znak
print(novy)
```

Výstup:

```
MAtUrItA z InfOrmAtIkY ####
```

4 Riešené úlohy

4.1 Kontrola čísla ISBN-10

Zadanie: Kniha má 10-miestne číslo ISBN. Je platné, ak súčet cifier vynásobených váhami 10, 9, 8, ..., 1 je deliteľný 11. Posledný znak môže byť aj X, ktorý znamená 10. Napíš funkciu `je_platne_isbn(isbn)`; pomlčky v zápise ignoruj.

Postup: odstránime pomlčky, overíme dĺžku a znaky, potom v cykle sčítame súčiny cifier a váh. Váha prvej cifry je 10, posledná má váhu 1 – teda $10 - i$ pre index i .

```
def je_platne_isbn(isbn):
    isbn = isbn.replace("-", "")
    if len(isbn) != 10:
        return False
    sucet = 0
    for i in range(10):
        znak = isbn[i]
        if znak.isdigit():
            hodnota = int(znak)
        elif znak == "X" and i == 9:
            hodnota = 10
        else:
            return False
        sucet += (10 - i) * hodnota
    return sucet % 11 == 0

testy = ["80-7178-506-0", "0-306-40615-2", "0-306-40615-3", "123456789X", "12345"]
for isbn in testy:
    print(f"{isbn:<15} {je_platne_isbn(isbn)}")
```

Výstup:

80-7178-506-0	False
0-306-40615-2	True
0-306-40615-3	False
123456789X	True
12345	False

Hraničné prípady v testoch: zlá dĺžka, X na zlom mieste, iný znak ako číslica.

4.2 Štatistika hodov kockou

Zadanie: Program sa opýta na počet hodov, „hodí“ kockou a vypíše, koľkokrát padla každá strana (aj ako riadok hviezdčiek), ktorá strana padla najčastejšie a priemer hodov. Rozdeľ program na funkcie `hody(n)`, `pocty(zoznam)` a hlavný program.

```
import random

def hody(n):
    return [random.randint(1, 6) for _ in range(n)]

def pocty(zoznam):
    vysledok = {strana: 0 for strana in range(1, 7)}
    for h in zoznam:
        vysledok[h] += 1
    return vysledok

if __name__ == "__main__":
    n = int(input("Počet hodov: "))
    z = hody(n)
    p = pocty(z)
    for strana in range(1, 7):
        print(f"{strana}: {'*' * p[strana]:<12} {p[strana]}")
    print("najčastejšie:", max(p, key=p.get))
    print(f"priemer: {sum(z) / n:.2f}")
```

Výstup:

```
Počet hodov: 30
1: *****      6
2: ***           3
3: *****      6
4: ****          4
5: *****      6
6: *****      5
najčastejšie: 1
priemer: 3.53
```

`{strana: 0 for strana in range(1, 7)}` je **generátorový zápis slovníka** – vytvorí slovník so všetkými stranami a počtom 0. Funkciu `pocty` možno otestovať aj bez náhody: `pocty([1, 1, 6])` musí dať 2 pre stranu 1 a 1 pre stranu 6. Ak padlo viac strán rovnako často (tu strany 1, 3 a 5), `max` vráti tú, ktorá je v slovníku prvá.

4.3 Teploty zo súboru

Zadanie: V súbore je na každom riadku dátum a teplota oddelené bodkočiarkou. Niektoré riadky sú poškodené. Vypíš priemernú teplotu, najteplejší deň a počet dní nad priemerom; poškodené riadky preskoč a vypíš ich čísla.

Súbor `teploty.txt`:

```
1.10.;12.5
2.10.;14.0
3.10.;chyba
4.10.;9.5

5.10.;17.0
6.10.;15.5
```

```
zaznamy = []
poskodene = []
with open("teploty.txt", encoding="utf-8") as f:
    for cislo, riadok in enumerate(f, start=1):
        try:
            datum, teplota = riadok.strip().split(";")
            zaznamy.append((datum, float(teplota)))
        except ValueError:
            poskodene.append(cislo)

teploty = [t for _, t in zaznamy]
priemer = sum(teploty) / len(teploty)
najteplejsi = max(zaznamy, key=lambda z: z[1])
print(f"priemer: {priemer:.2f} °C")
print(f"najteplejšie: {najteplejsi[0]} ({najteplejsi[1]} °C)")
print("dní nad priemerom:", len([t for t in teploty if t > priemer]))
print("poškodené riadky:", poskodene)
```

Výstup:

```
priemer: 13.70 °C
najteplejšie: 5.10. (17.0 °C)
dní nad priemerom: 3
poškodené riadky: [3, 5]
```

Jeden blok `try` pokryje obe chyby: riadok bez bodkočiarky (`split` vráti iný počet častí a rozbalenie zlyhá s `ValueError`) aj text namiesto čísla (`float` vyvolá `ValueError`). Prázdný riadok (riadok 5) je tiež poškodený.

4.4 Grafika: graf teplôt

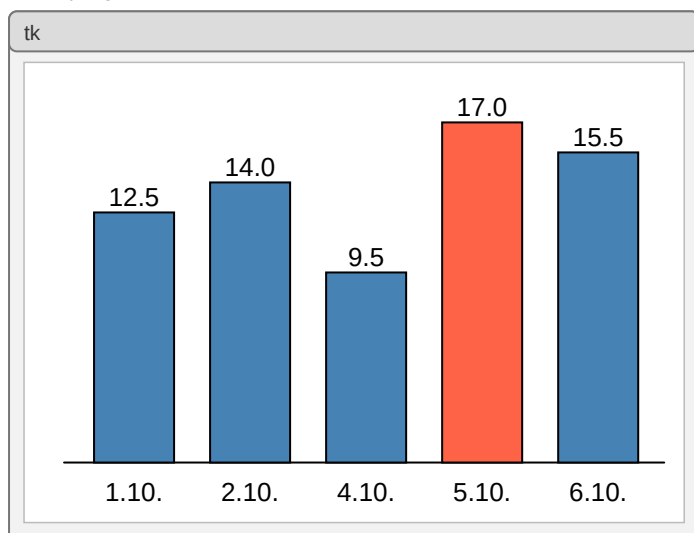
Zadanie: Údaje z predchádzajúcej úlohy zobraz ako stĺpcový graf v tkinteri: výška stĺpca zodpovedá teplote (10 pixelov na stupeň), pod stĺpcom je dátum.

```
import tkinter

zaznamy = [("1.10.", 12.5), ("2.10.", 14.0), ("4.10.", 9.5),
           ("5.10.", 17.0), ("6.10.", 15.5)]

platno = tkinter.Canvas(width=330, height=230, bg="white")
platno.pack()
zaklad = 200 # y vodorovnej osi
platno.create_line(20, zaklad, 320, zaklad)
for i, (datum, t) in enumerate(zaznamy):
    x = 35 + i * 58
    vyska = t * 10
    platno.create_rectangle(x, zaklad - vyska, x + 40, zaklad,
                           fill="tomato" if t == 17.0 else "steelblue")
    platno.create_text(x + 20, zaklad - vyska - 8, text=str(t))
    platno.create_text(x + 20, zaklad + 14, text=datum)
platno.mainloop()
```

Okno programu:



Os y plátna smeruje nadol, preto vrch stĺpca je `zaklad - vyska`.

5 Kontrolný zoznam pred odovzdaním

- Robí program to, čo chce zadanie – nie viac, nie menej?
- Zodpovedá výstup **presne** požadovanému tvaru (texty, medzery, desatinné miesta)?
- Funguje program aj pre **hraničné** vstupy (prázdný, jeden prvok, nula, záporné, rovnaké hodnoty)?
- Sú čísla zo vstupu prevedené na `int` či `float`?
- Majú funkcie, ktoré zadanie predpisuje, **presne** požadované meno a parametre a vracajú výsledok cez `return`?
- Je hlavný program pod `if __name__ == "__main__":` (ak úloha testuje funkcie)?
- Sú odstránené pomocné výpisy?
- Je program čitateľný – výstižné mená premenných, komentáre pri zložitejších miestach?

Úlohy na precvičenie

1. Napíš funkciu, ktorá overí platnosť čísla ISBN-13 (váhy sa striedajú 1 a 3, súčet musí byť deliteľný 10).
2. Vygeneruj 20 náhodných čísel od 1 do 50 a vypíš tie, ktoré sú väčšie ako priemer, zoradené zostupne.
3. Načítavaj slová, kým sa nezadá prázdny riadok; vypíš najdlhšie slovo a počet slov začínajúcich samohláskou.
4. Zo súboru s riadkami `meno;body` vypíš poradie podľa bodov a mená s rovnakým počtom bodov zorad' abecedne.
5. Nakresli v tkinteri graf výsledkov z úlohy 4 – vodorovné stĺpce s menami vľavo.
6. Prejdi lekcie témy Maturitná príprava na Mente a pri každej úlohe si najprv napíš plán podľa postupu vyššie.

Úlohy k tejto téme nájdeš na learnwithmente.com v časti Informatika → Python.