

Zdrojový kód v jazyku Python

Projekt v jazyku Python

Základné informácie o použitých vývojových prostriedkoch

Pycharm – IDE - Integrated development environment - integrované vývojové prostredie, je distribuované s JRE (Java Runtime Environment), takže nie je nutné inštalovať Javu pre spustenie programu Pycharm

Edície Pycharm

Professional - je k dispozícii ako platený profesionálny nástroj - podporuje viac programovacích jazykov, vývoj pre WEB, pre databázy – SQL atď.

Community, Educational - sú k dispozícii zadarmo (opensource) pre prácu, napr. Python, XML, JSON – pre vedecké účely

Python - programovací jazyk existujúci vo verziách:

Verzia 2 - 2.6 a 2.7

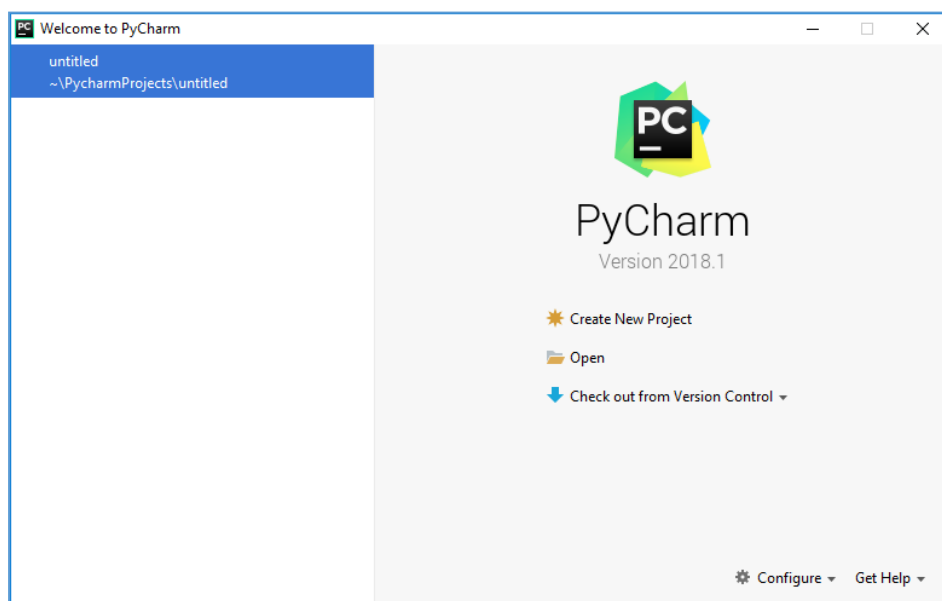
Verzia 3 - od 3.4 po 3.7

V uvedenom projekte boli použité verzie

Pycharm 2018.1 (Community edition) a Python 3.6.5

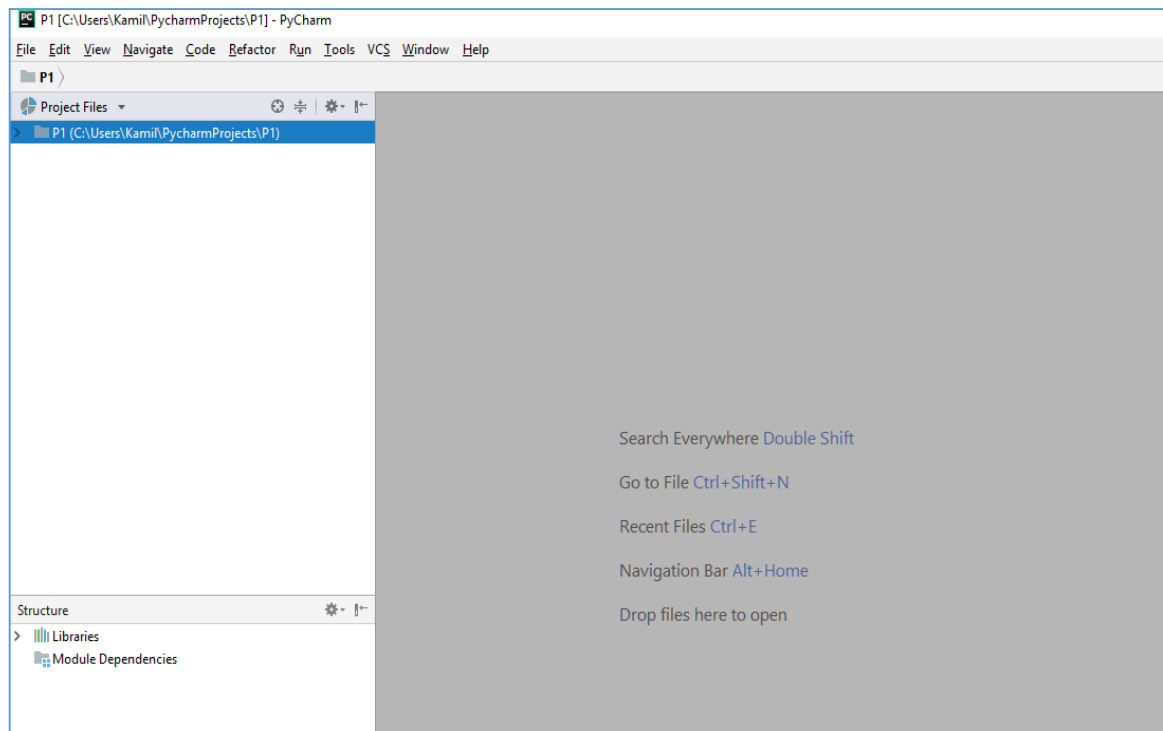
Spustenie a tvorba projektu

Projekt:

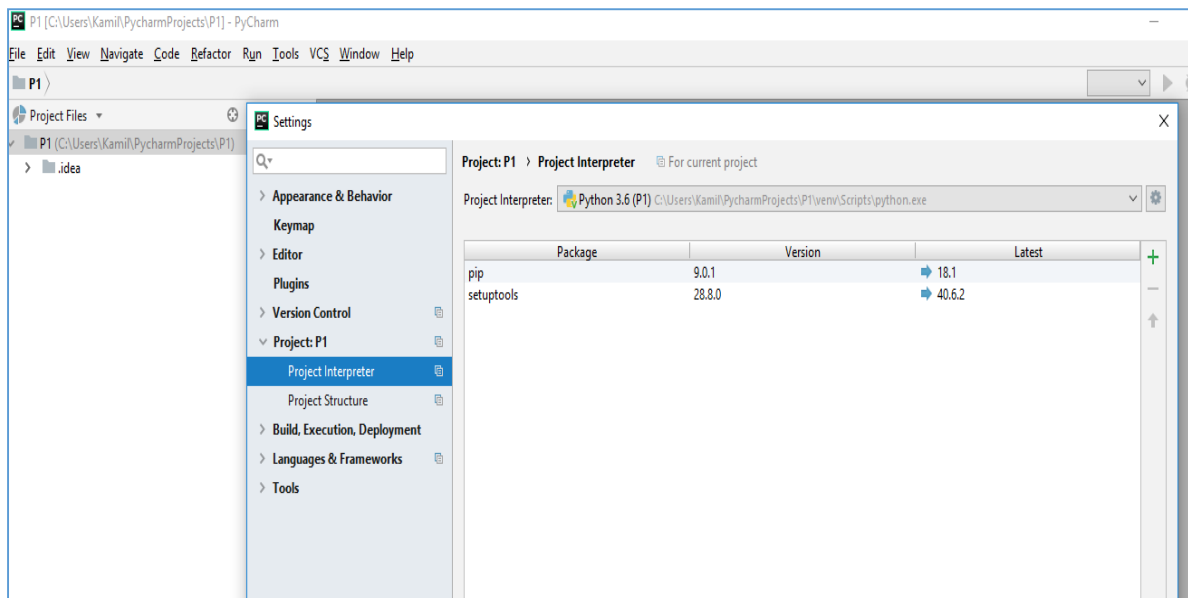


Ak potrebujeme vytvoriť nový projekt vyberieme Create New Project; ak chceme otvoriť existujúci – vyberieme možnosť Open.

Vytvoríme nový projekt, ktorý pomenujeme P1.

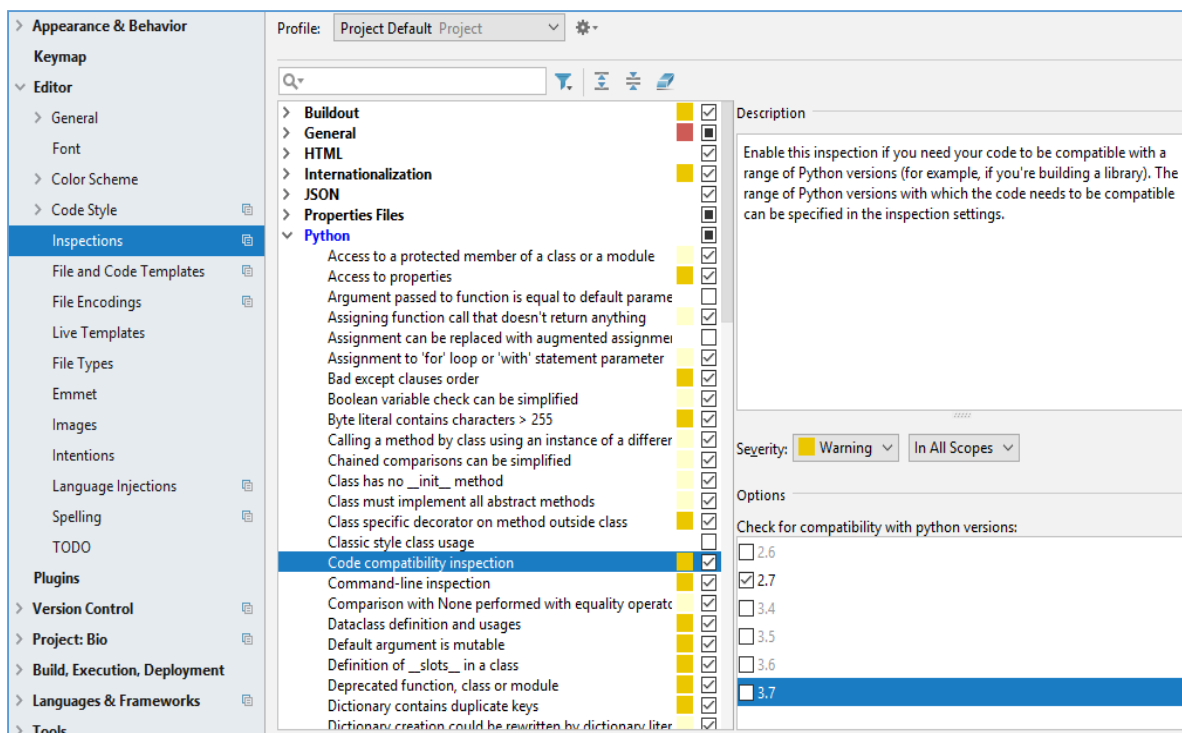


Voľby: File – Settings – Projekt P1 – Projekt Interpreter. Pre Projekt interpreter je nutné pridať cestu k spustiteľnému tvaru Python.exe

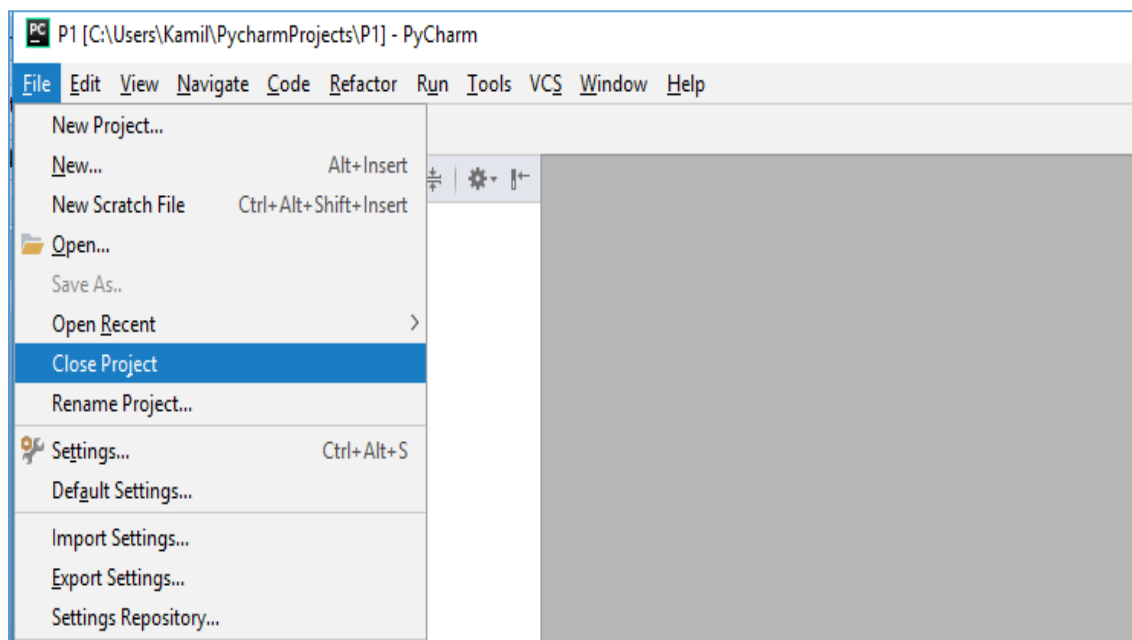


Nastavíme kompatibilitu kódu s použitou verzíou jazyka Python

Voľby: File - Settings – Editor – Inspections – Python – Code compatibility Inspection

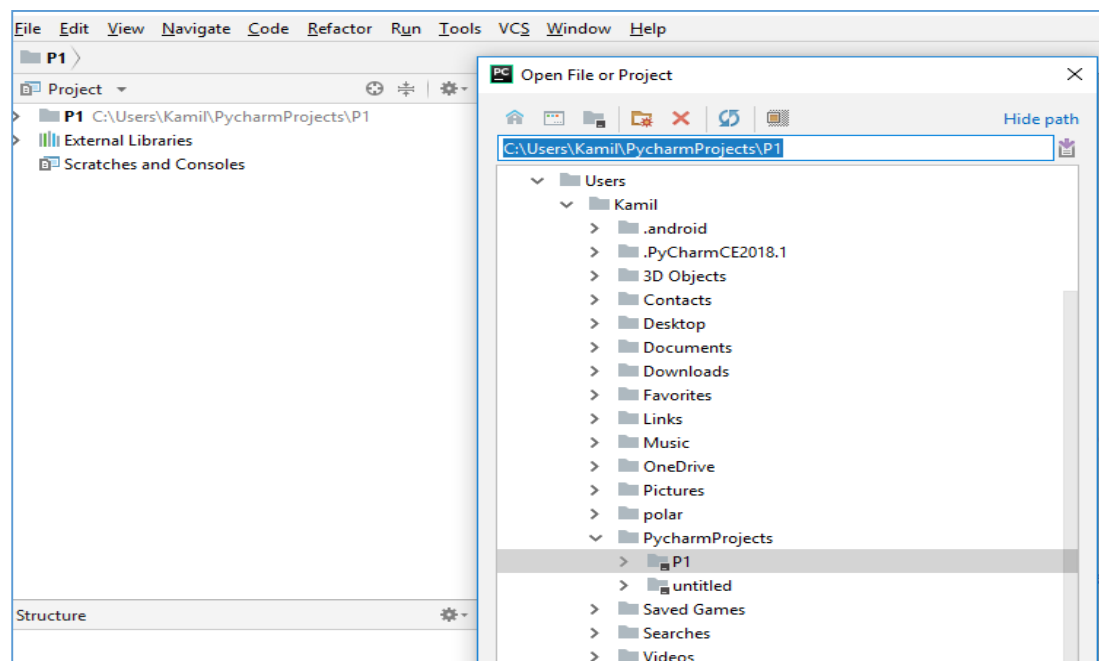
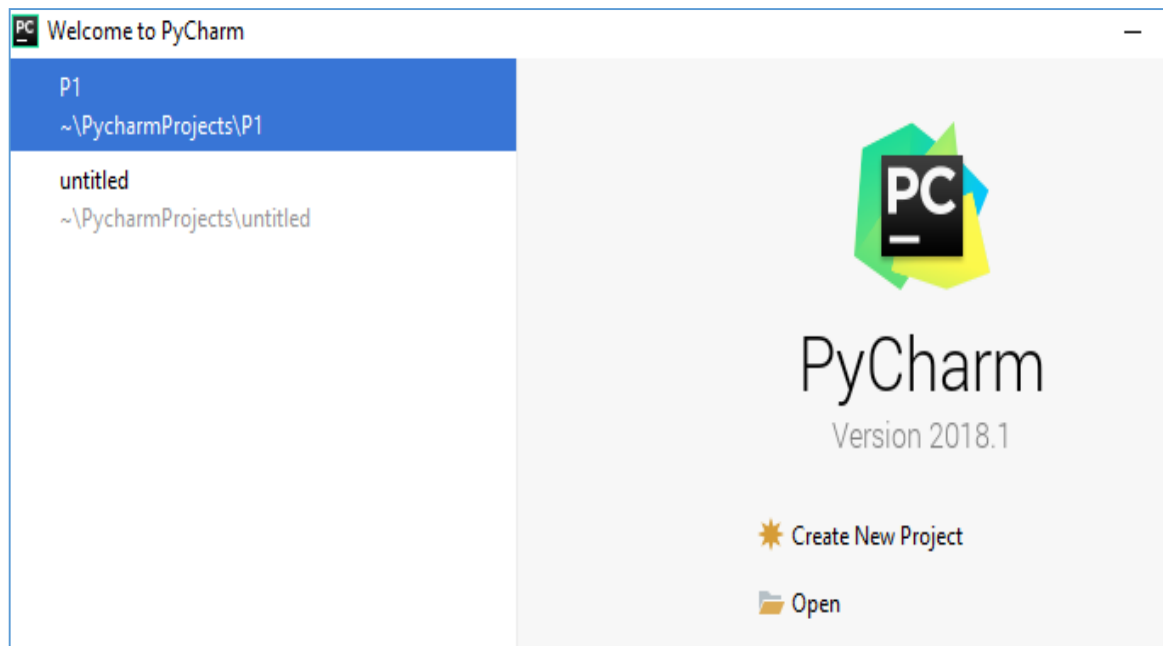


Ukončenie projektu P1



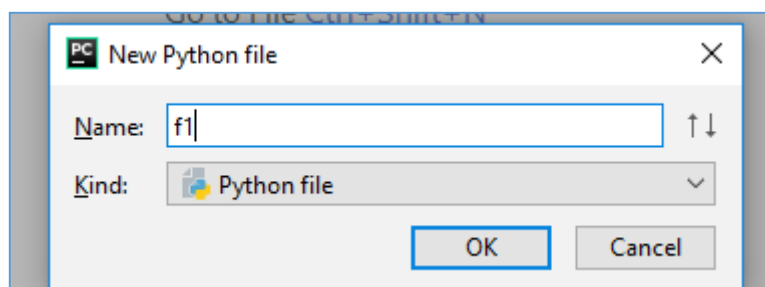
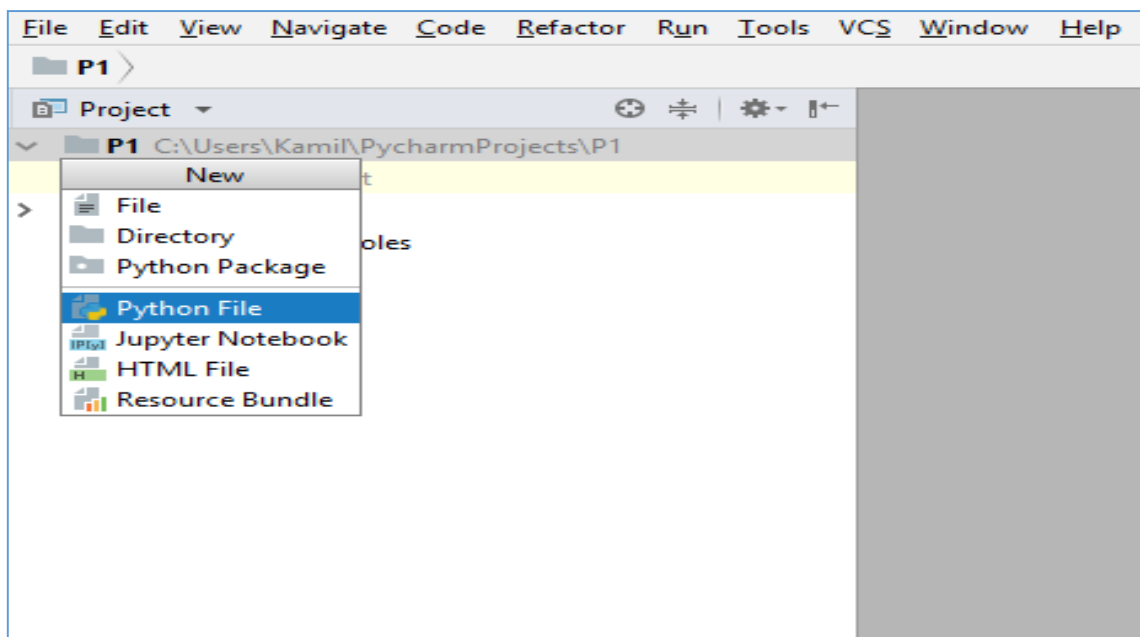
Otvorenie existujúceho projektu P1

Ak chceme otvoriť už existujúci projekt klikneme na P1 v úvodnej obrazovke, alebo vyberieme z menu File – Open.

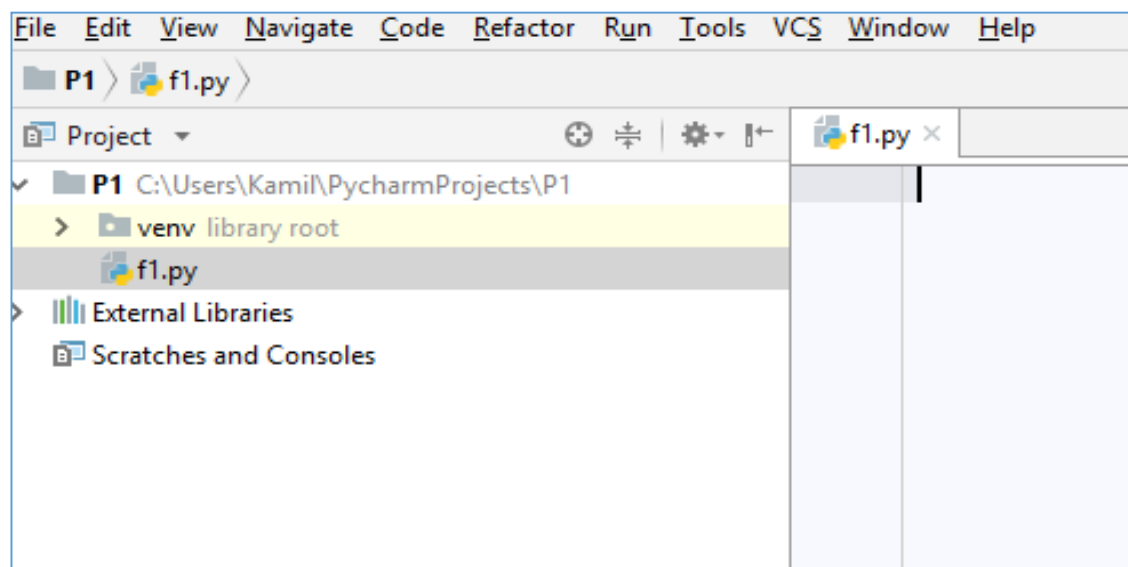


Pridanie súboru do projektu:

Voľba: File - New - Python File.

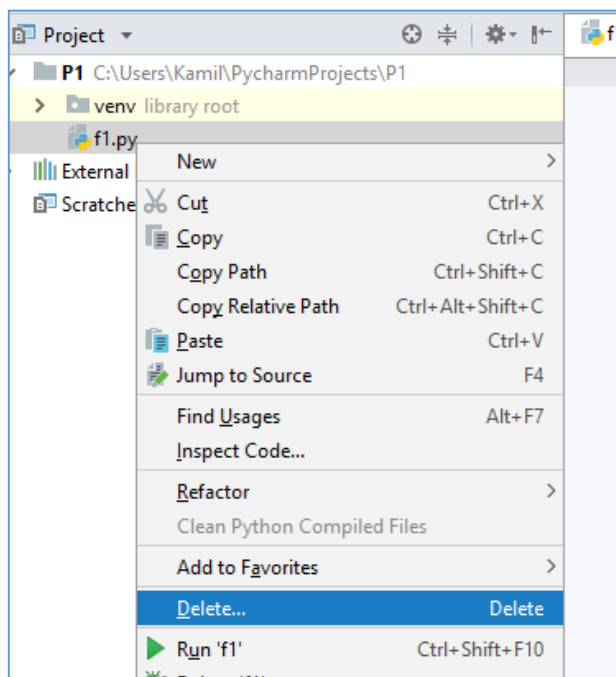


Stav po pridání



Vymazanie súboru f1

Ak chceme odstrániť súbor f1 z projektu – pravým tlačidlom myši vyvoláme kontextové menu a použijeme voľbu Delete.

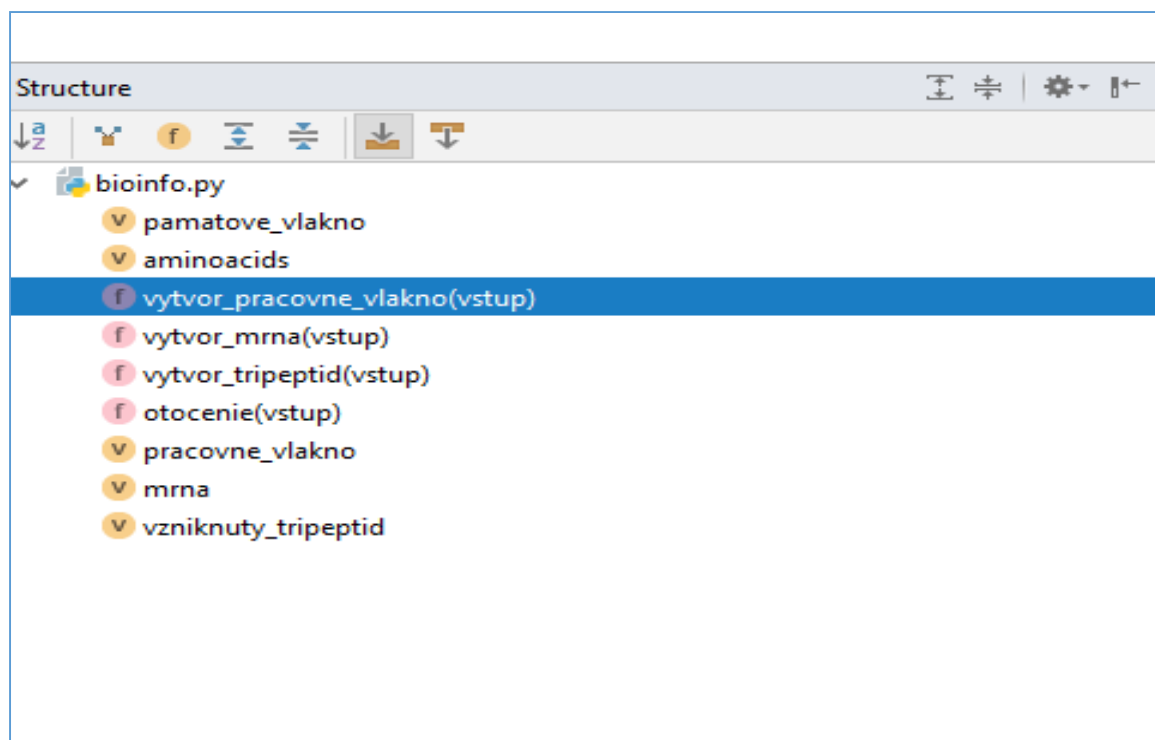


Spustenie aplikácie:

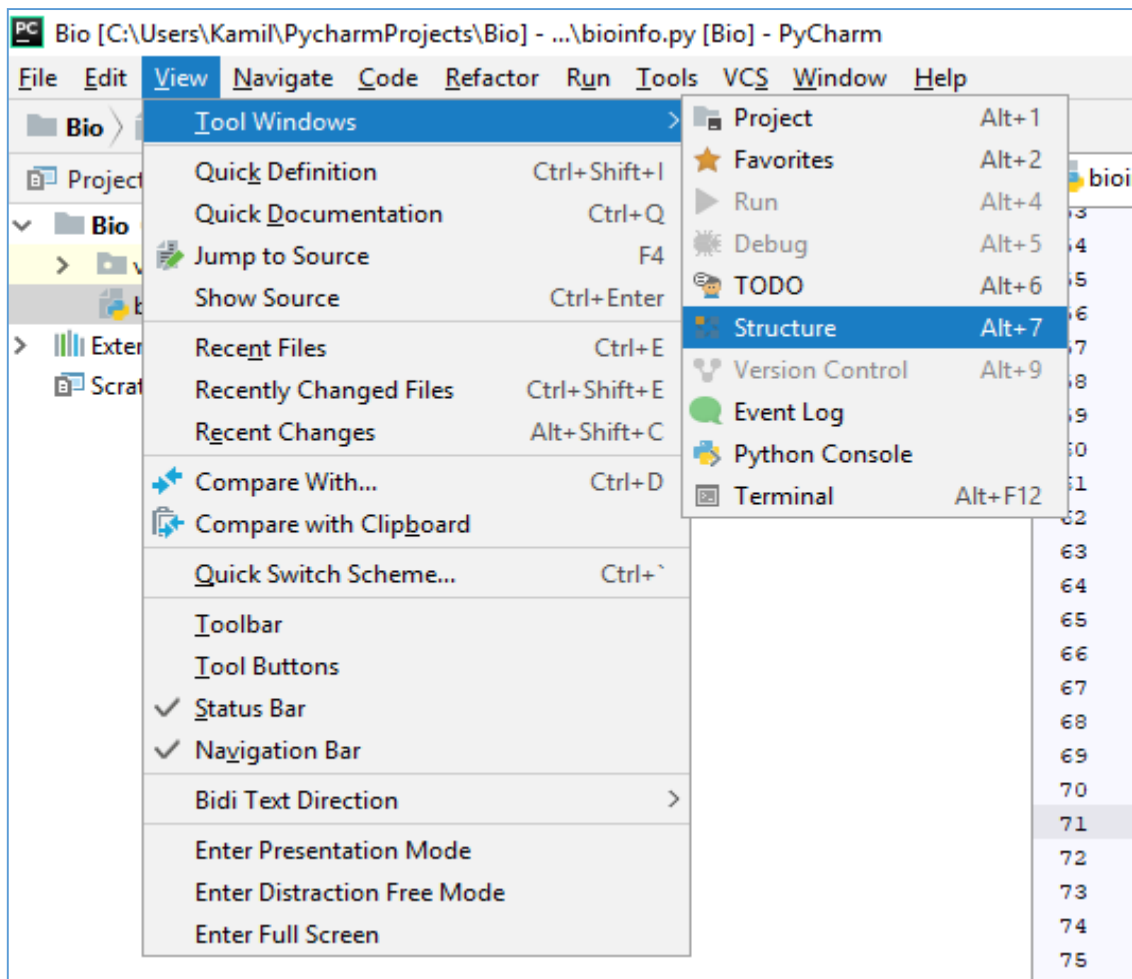
Vytvoríme nový projekt Bio. Pridáme doň súbor bioinfo. Bioinfo je súbor, ktorý má za úlohu: priradiť k zadanému vláknu DNA komplementárne vlákno, dopísať sekvenciu mRNA, zapísať vzniknutú sekvenciu aminokyselín.

```
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Bio > bioinfo.py >
Project C:\Users\Kamil\PycharmProjects\Bio
venv library root
bioinfo.py
External Libraries
Scratches and Consoles
1 pamatove_vlakno = input("Zadaj pamätové vlákno DNA:")
2
3 aminoacids = {
4     "UUU": "phe",
5     "UUC": "phe",
6     "UUA": "leu",
7     "UUG": "leu",
8     "UCU": "ser",
9     "UCC": "ser",
10    "UCA": "ser",
11    "UCG": "ser",
12    "UAU": "tyr",
13    "UAC": "tyr",
```

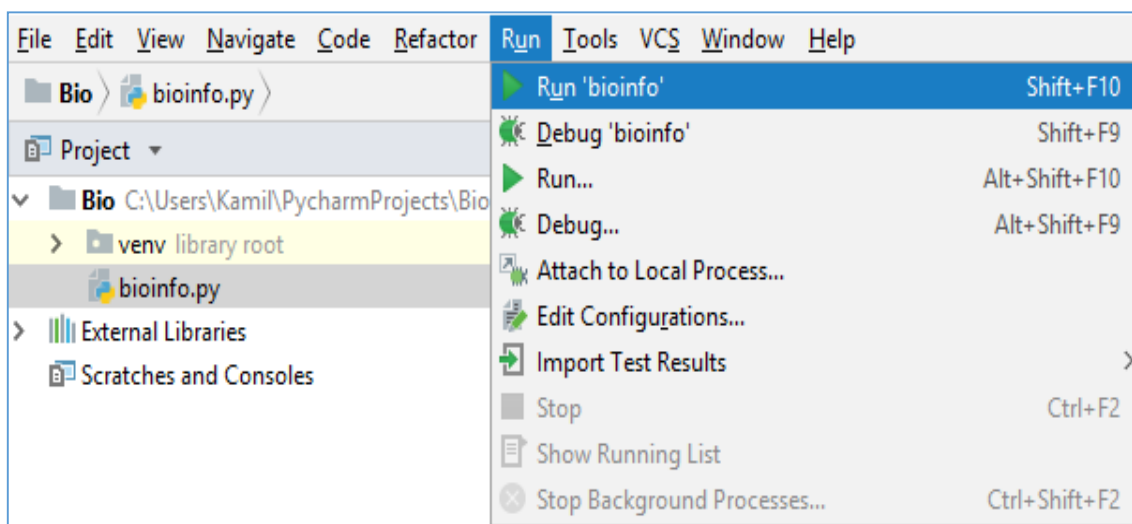
Štruktúra súboru bioinfo.py



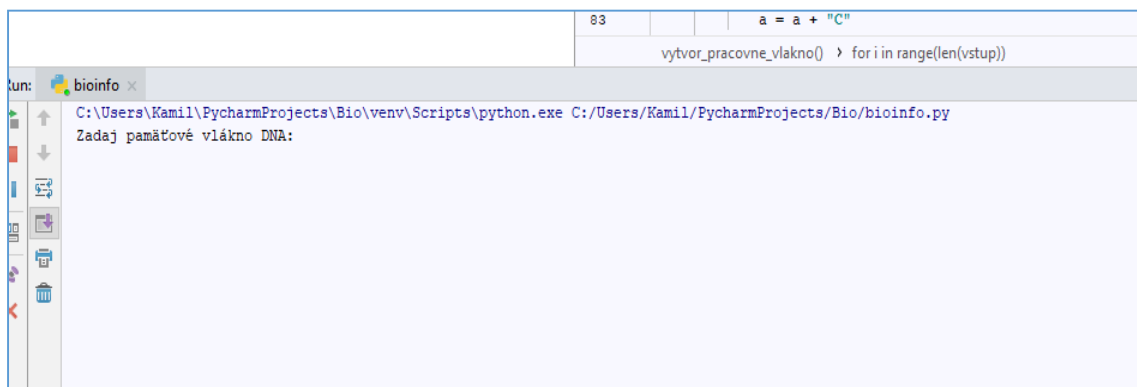
Túto štruktúru vieme získať použitím volieb: View - Tool Windows - Structure.



Spustenie programu: použijeme voľbu Run bioinfo.



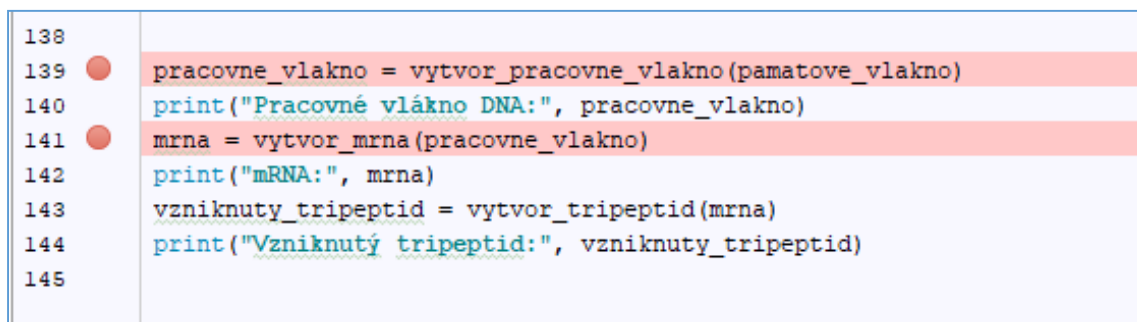
Komunikácia s programom.



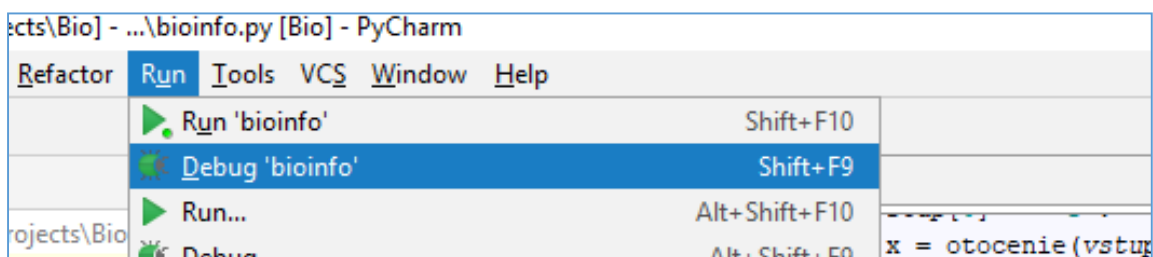
Ladenie programu:

Nastavenie stopadries.

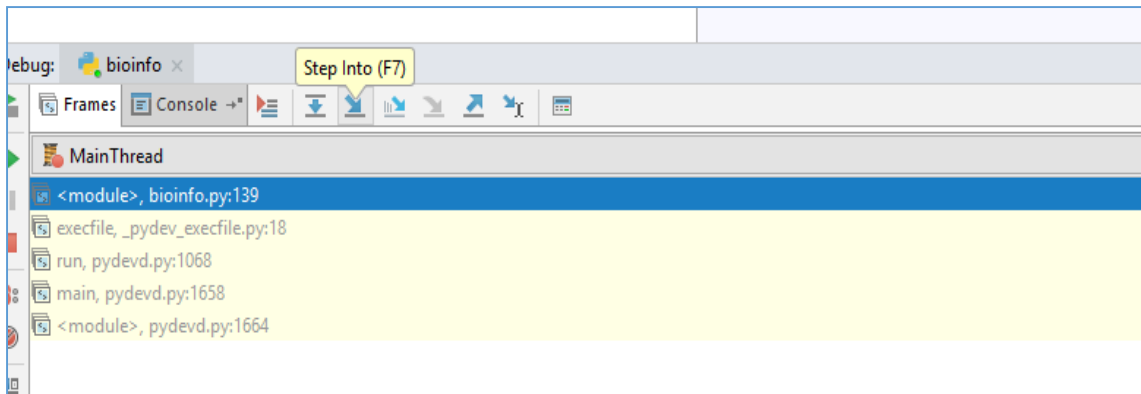
Stopadresy nastavíme kliknutím napravo od čísla riadku v okne zdrojového kódu.



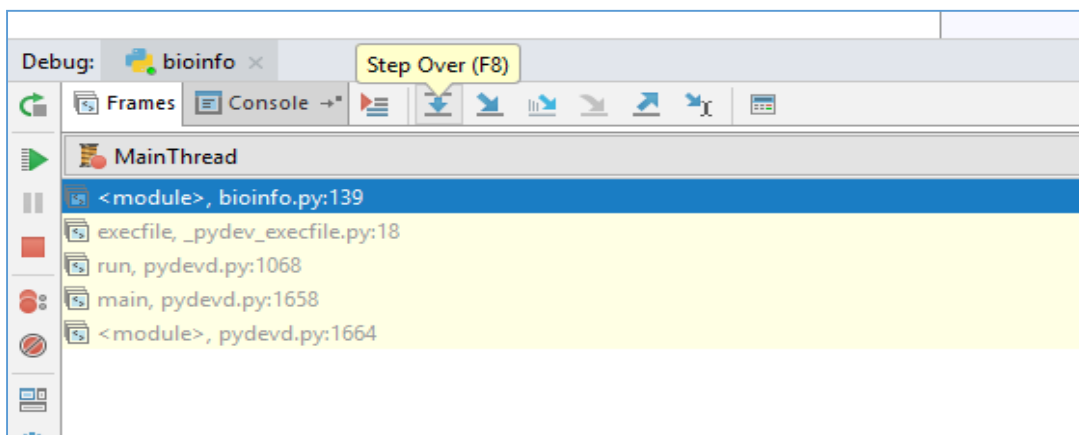
Štart ladenia



Vstup do ladenej funkcie : kláves F7 alebo zvýraznený symbol.



Pokračovanie bez vstupu do ladenej funkcie - kláves F8 alebo zvýraznený symbol.



Štart programu z príkazového riadku

```
C:\Users\Kamil\PycharmProjects\Bio>python bioinfo.py
Zadaj pamäťové vlákno DNA:GGG
```

RIEŠENIE PRÍKLADU 1 V JAZYKU PYTHON

```
znaky=["3","5","A","T","G","C"]
```

#pracovne vlakno

```
def vytvor_pracovne_vlakno(vstup):
```

```
    a = ""
```

```
    for i in range(len(vstup)):
```

```
        if vstup[i] == "5":
```

```
            a = a + "3"
```

```
        elif vstup[i] == "3":
```

```
            a = a + "5"
```

```
        elif vstup[i] == "A":
```

```
            a = a + "T"
```

```

    elif vstup[i] == "T":
        a = a + "A"
    elif vstup[i] == "G":
        a = a + "C"
    elif vstup[i] == "C":
        a = a + "G"
    else:
        a = a + vstup[i]
    return a

kodujuce_vlakno = input("Zadaj kódujúce vlákno:")
i = len(kodujuce_vlakno)

if i < 5:
    print("Prliš krátky reťazec")
    exit(0)

if (kodujuce_vlakno[0] != '3') and (kodujuce_vlakno[0] != '5'):
    print("Nesprávny prvý znak")
    exit(0)
if (kodujuce_vlakno[i-1] != '3') and (kodujuce_vlakno[i-1] != '5'):
    print("Nesprávny posledný znak")
    exit(0)

if kodujuce_vlakno[0] == kodujuce_vlakno[i-1]:
    print("Nepripustná zhoda medzi prvým a posledným znakom")
    exit(0)

for j in range (0,i):
    if kodujuce_vlakno[j] not in znaky:
        print("Nesprávny znak v retazci: "+kodujuce_vlakno[j])
        exit(0)

pracovne_vlakno = vytvor_pracovne_vlakno(kodujuce_vlakno)
print("Pracovné vlákno:", pracovne_vlakno)

```

RIEŠENIE PRÍKLADOV 2 A 3 V JAZYKU PYTHON

```

znaky=["3","5","A","T","G","C"]

#pracovne vlakno
def vytvor_pracovne_vlakno(vstup):
    a = ""
    for i in range(len(vstup)):
        if vstup[i] == "5":
            a = a + "3"
        elif vstup[i] == "3":
            a = a + "5"

```

```
    elif vstup[i] == "A":
        a = a + "T"
    elif vstup[i] == "T":
        a = a + "A"
    elif vstup[i] == "G":
        a = a + "C"
    elif vstup[i] == "C":
        a = a + "G"
    else:
        a = a + vstup[i]
return a
```

```
def vytvor_mrna(vstup):
    b = ""
    for i in range(len(vstup)):
        if vstup[i] == "5":
            b = b + "3"
        elif vstup[i] == "3":
            b = b + "5"
        elif vstup[i] == "A":
            b = b + "U"
        elif vstup[i] == "T":
            b = b + "A"
        elif vstup[i] == "G":
            b = b + "C"
        elif vstup[i] == "C":
            b = b + "G"
        else:
            b = b + vstup[i]
    return b
```

```
kodujuce_vlakno = input("Zadaj kódujúce vlákno:")
i = len(kodujuce_vlakno)
```

```
if i < 5:
    print("Prliš krátky reťazec")
    exit(0)
```

```
if (kodujuce_vlakno[0] != '3') and (kodujuce_vlakno[0] != '5'):
    print("Nesprávny prvý znak")
    exit(0)
```

```
if (kodujuce_vlakno[i-1] != '3') and (kodujuce_vlakno[i-1] != '5'):
    print("Nesprávny posledný znak")
    exit(0)
```

```
if kodujuce_vlakno[0] == kodujuce_vlakno[i-1]:
    print("Neprípustná zhoda medzi prvým a posledným znakom")
    exit(0)
```

```
for j in range(0,i):
```

```
if kodujuce_vlakno[j] not in znaky:
    print("Nesprávny znak v retazci: "+kodujuce_vlakno[j])
    exit(0)

pracovne_vlakno = vytvor_pracovne_vlakno(kodujuce_vlakno)
print("Pracovné vlákno:", pracovne_vlakno)

mrna = vytvor_mrna(pracovne_vlakno)
print("mRNA:", mrna)
```

RIEŠENIE PRÍKLADU 4 V JAZYKU PYTHON

```
aminoacids = {
    "UUU": "phe",
    "UUC": "phe",
    "UUA": "leu",
    "UUG": "leu",
    "UCU": "ser",
    "UCC": "ser",
    "UCA": "ser",
    "UCG": "ser",
    "UAU": "tyr",
    "UAC": "tyr",
    "UAA": "stop",
    "UAG": "stop",
    "UGU": "cys",
    "UGC": "cys",
    "UGA": "stop",
    "UGG": "trp",
    "CUU": "leu",
    "CUC": "leu",
    "CUA": "leu",
    "CUG": "leu",
    "CCU": "pro",
    "CCC": "pro",
    "CCA": "pro",
    "CCG": "pro",
    "CAU": "his",
    "CAC": "his",
    "CAA": "glu",
    "CAG": "glu",
    "CGU": "arg",
    "CGC": "arg",
    "CGA": "arg",
    "CGG": "arg",
    "AUU": "ile",
    "AUC": "ile",
    "AUA": "ile",
    "AUG": "met",
```

```

"ACU": "thr",
"ACC": "thr",
"ACA": "thr",
"ACG": "thr",
"AAU": "asn",
"AAC": "asn",
"AAA": "lys",
"AAG": "lys",
"AGU": "ser",
"AGC": "ser",
"AGA": "arg",
"AGG": "arg",
"GUU": "val",
"GUC": "val",
"GUA": "val",
"GUG": "val",
"GCU": "ala",
"GCC": "ala",
"GCA": "ala",
"GCG": "ala",
"GAU": "asp",
"GAC": "asp",
"GAA": "glu",
"GAG": "glu",
"GGU": "gly",
"GGC": "gly",
"GGA": "gly",
"GGG": "gly"
}

```

```

znaky=["3","5","A","U","G","C"]
inic_aacid = "met"

```

```

def vytvor_peptid(vstup):
    je_inic = 0
    je_stop = 0
    je_pomlcka = 0

    triplet = ""
    aminoacid = ""
    aacid = ""
    horny_index = len(vstup)-1

    i = 1
    while i < horny_index-2:
        triplet = vstup[i] + vstup[i + 1] + vstup[i + 2]
        aacid = aminoacids[triplet]
        if aacid == "stop":
            je_stop = 1
            break
        else:

```

```

    if aacid == inic_aacid:
        je_inic = 1

    if je_inic == 1:
        if je_pomlcka == 0:
            aminoacid = aminoacid + aacid
            je_pomlcka = 1
        else:
            aminoacid = aminoacid + "-" + aacid
    i = i + 3

    if je_inic == 0:
        aminoacid = "Inicializačný kodon nebol nájdený"

    if je_stop == 0:
        aminoacid = "Stop kodon nebol nájdený"

    #aminoacid = aminoacid[1:]
    return aminoacid

kodujuce_vlakno = input("Zadaj kódujúce vlákno:")
i = len(kodujuce_vlakno)

if i < 5:
    print("Prliš krátky reťazec")
    exit(0)

if (kodujuce_vlakno[0] != '3') and (kodujuce_vlakno[0] != '5'):
    print("Nesprávny prvý znak")
    exit(0)
if (kodujuce_vlakno[i-1] != '3') and (kodujuce_vlakno[i-1] != '5'):
    print("Nesprávny posledný znak")
    exit(0)

if kodujuce_vlakno[0] == kodujuce_vlakno[i-1]:
    print("Neprípustná zhoda medzi prvým a posledným znakom")
    exit(0)

for j in range (0,i):
    if kodujuce_vlakno[j] not in znaky:
        print("Nesprávny znak v retazci: "+kodujuce_vlakno[j])
        exit(0)

vzniknuty_peptid=vytvor_peptid(kodujuce_vlakno)
print("Vzniknutá aminokyselina/ peptid",vzniknuty_peptid)

```

RIEŠENIE PRÍKLADU 5 V JAZYKU PYTHON

```
aminoacids = {  
  "UUU": "phe",  
  "UUC": "phe",  
  "UUA": "leu",  
  "UUG": "leu",  
  "UCU": "ser",  
  "UCC": "ser",  
  "UCA": "ser",  
  "UCG": "ser",  
  "UAU": "tyr",  
  "UAC": "tyr",  
  "UAA": "stop",  
  "UAG": "stop",  
  "UGU": "cys",  
  "UGC": "cys",  
  "UGA": "stop",  
  "UGG": "trp",  
  "CUU": "leu",  
  "CUC": "leu",  
  "CUA": "leu",  
  "CUG": "leu",  
  "CCU": "pro",  
  "CCC": "pro",  
  "CCA": "pro",  
  "CCG": "pro",  
  "CAU": "his",  
  "CAC": "his",  
  "CAA": "glu",  
  "CAG": "glu",  
  "CGU": "arg",  
  "CGC": "arg",  
  "CGA": "arg",  
  "CGG": "arg",  
  "AUU": "ile",  
  "AUC": "ile",  
  "AUA": "ile",  
  "AUG": "met",  
  "ACU": "thr",  
  "ACC": "thr",  
  "ACA": "thr",  
  "ACG": "thr",  
  "AAU": "asn",  
  "AAC": "asn",  
  "AAA": "lys",  
  "AAG": "lys",  
  "AGU": "ser",  
  "AGC": "ser",  
  "AGA": "arg",  
  "AGG": "arg",  
  "GUU": "val",  
  "GUC": "val",  
  "GUA": "val",  
}
```



```
"GUG": "val",
"GCU": "ala",
"GCC": "ala",
"GCA": "ala",
"GCG": "ala",
"GAU": "asp",
"GAC": "asp",
"GAA": "glu",
"GAG": "glu",
"GGU": "gly",
"GGC": "gly",
"GGA": "gly",
"GGG": "gly"
}
```

```
znaky=["3","5","A","T","G","C"]
inic_aacid = "met"
```

#pracovne vlakno

```
def vytvor_pracovne_vlakno(vstup):
```

```
    a = ""
    for i in range(len(vstup)):
        if vstup[i] == "5":
            a = a + "3"
        elif vstup[i] == "3":
            a = a + "5"
        elif vstup[i] == "A":
            a = a + "T"
        elif vstup[i] == "T":
            a = a + "A"
        elif vstup[i] == "G":
            a = a + "C"
        elif vstup[i] == "C":
            a = a + "G"
        else:
            a = a + vstup[i]
    return a
```

```
def vytvor_mrna(vstup):
```

```
    b = ""
    for i in range(len(vstup)):
        if vstup[i] == "5":
            b = b + "3"
        elif vstup[i] == "3":
            b = b + "5"
        elif vstup[i] == "A":
            b = b + "U"
        elif vstup[i] == "T":
            b = b + "A"
        elif vstup[i] == "G":
            b = b + "C"
```

```

    elif vstup[i] == "C":
        b = b + "G"
    else:
        b = b + vstup[i]
    return b

```

```

def vytvor_tripectid(vstup):

```

```

    if vstup[0] == "3":
        x = otocenie(vstup)
        print("otoceny:", x)
    else:
        # x = vstup[2:-2]
        x = vstup[1]+vstup[2]+vstup[3]

```

```

    triplet = ""
    aminoacid = ""
    i = 0
    while i < len(x):
        triplet = x[i] + x[i + 1] + x[i + 2]
        kodon = aminoacids[triplet]
        if kodon == "stop":
            break
        else:
            aminoacid = aminoacid + "-" + kodon
        i = i + 3
    aminoacid = aminoacid[1:]
    return aminoacid

```

```

def vytvor_peptid(vstup):

```

```

    je_inic = 0
    je_stop = 0
    je_pomlcka = 0

    triplet = ""
    aminoacid = ""
    aacid = ""
    horny_index = len(vstup)-1

    i = 1
    while i < horny_index-2:
        triplet = vstup[i] + vstup[i + 1] + vstup[i + 2]
        aacid = aminoacids[triplet]
        if aacid == "stop":
            je_stop = 1
            break
        else:
            if aacid == inic_aacid:
                je_inic = 1

            if je_inic == 1:
                if je_pomlcka == 0:

```

```

        aminoacid = aminoacid + aacid
        je_pomlcka = 1
    else:
        aminoacid = aminoacid + "-" + aacid
    i = i + 3

if je_inic == 0:
    aminoacid = "Inicializačný kodon nebol nájdený"

if je_stop == 0:
    aminoacid = "Stop kodon nebol nájdený"

    #aminoacid = aminoacid[1:]
return aminoacid

def otocenie(vstup):
    otoceny = ""
    if vstup[0] != '3':
        return vstup
    l = len(vstup)-1
    for i in range(l,-1,-1 ):
        otoceny = otoceny+vstup[i]
        # znak = vstup[i]
        # otoceny = otoceny+znak

    return otoceny

kodujuce_vlakno = input("Zadaj kódujúce vlákno:")
i = len(kodujuce_vlakno)

if i < 5:
    print("Prliš krátky reťazec")
    exit(0)

if (kodujuce_vlakno[0] != '3') and (kodujuce_vlakno[0] != '5'):
    print("Nesprávny prvý znak")
    exit(0)
if (kodujuce_vlakno[i-1] != '3') and (kodujuce_vlakno[i-1] != '5'):
    print("Nesprávny posledný znak")
    exit(0)

if kodujuce_vlakno[0] == kodujuce_vlakno[i-1]:
    print("Neprípustná zhoda medzi prvým a posledným znakom")
    exit(0)

for j in range (0,i):
    if kodujuce_vlakno[j] not in znaky:
        print("Nesprávny znak v reťazci: "+kodujuce_vlakno[j])
        exit(0)

```

```
pracovne_vlakno = vytvor_pracovne_vlakno(kodujuce_vlakno)
print("Pracovné vlákno DNA:", pracovne_vlakno)

mrna = vytvor_mrna(pracovne_vlakno)
print("mRNA:", mrna)

otocene = otocenie(mrna)
print("Správne orientované vlákno", otocene)

vzniknuty_peptid=vytvor_peptid(mrna)
print("Vzniknutá aminokyselina/ peptid",vzniknuty_peptid)
```
