Νίκος Μ. Χατζηγιαννάκης

Η γλώσσα Python σε βάθος

Περιλαμβάνει εισαγωγή
στην επιστήμη των υπολογιστών
και τον προγραμματισμό

6η εργασία – Παίζουμε τριανταμία;

ΚΛΕΙΔΑΡΙΘΜΟΣ

6η Εργασία - Παίζουμε τριανταμία;

Σκεπτικό

Το παιχνίδι θα παίζεται με μια στοίβα η οποία μπορεί να περιέχει κάρτες από μία ή περισσότερες τράπουλες. Θα χρησιμοποιήσουμε αντικειμενοστρεφή λογική και σχεδίαση για την υλοποίηση του προγράμματος. Έτσι, κάθε κάρτα θα είναι ένα αντικείμενο μιας κλάσης το οποίο θα διαθέτει ιδιότητες όπως τη φιγούρα του (π.χ. Βαλές, Δέκα, Ρήγας, Άσσος, κ.λπ.), το είδος του (Σπαθί ♣, Καρό, Μπαστούνι♠ ή Κούπα♥), καθώς και την αξία του σε πόντους (1,2,...,10). Κάθε τράπουλα θα είναι και αυτή ένα αντικείμενο μιας κλάσης με ιδιότητες το χρώμα της, καθώς και τις κάρτες που περιέχει. Αρχικά κάθε αντικείμενο-τράπουλα θα μπορεί να περιέχει ένα ή και περισσότερα σετ από 52 αντικείμενα διαφορετικές κάρτες, όπως και μια ή περισσότερες πραγματικές τράπουλες.

Αρχικά στη στοίβα προστίθενται κάρτες από μία ή περισσότερες τράπουλες. Στην πραγματικότητα η στοίβα είναι και αυτή ένα αντικείμενο-τράπουλα. Στη συνέχεια τα περιεχόμενα της στοίβας ανακατεύονται ώστε η σειρά τους να είναι τυχαία.

Όταν αρχίζει να παίζεται το παιχνίδι, τόσο η «μάνα» όσο και ο παίκτης τραβάνε κάρτες από τη κορυφή της στοίβας, με την λογική που αναφέρεται στην εκφώνηση. Κάθε κάρτα που χρησιμοποιείται αφαιρείται από τη στοίβα. Όταν οι κάρτες που παραμένουν στη στοίβα λιγοστέψουν τόσο ώστε να μην μπορεί να συνεχιστεί το παιχνίδι, ο παίκτης ερωτάται αν θέλει να προστεθεί ακόμα μια τράπουλα στη στοίβα για να συνεχίσει ή αν επιθυμεί να σταματήσει.

Η «Μάνα» και ο παίκτης θα είναι και αυτά αντικείμενα αντίστοιχων κλάσεων που θα διαθέτουν ιδιότητες και μεθόδους.

Βήματα

❶ Το πρώτο βήμα είναι να ορίσουμε τις απαραίτητες κλάσεις από τις οποίες θα δημιουργηθούν τα αντικείμενα με τα οποία θα δομήσουμε και θα διαχειριστούμε το πρόγραμμά μας.

❷ Ορίζουμε την κλάση karta από την οποία θα δημιουργούμε αντικείμενα-κάρτες από τα οποία θα αποτελούνται οι τράπουλες και η στοίβα με την οποία θα παίζουμε το παιχνίδι.

- Η μέθοδος δόμησης __init__() της κλάσης δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου τη φιγούρα (Βαλές, Δέκα, Άσσος, κ.λπ.), το είδος (Κούπα, Καρό, κ.λπ.) και την τιμή (δηλαδή τους πόντους κάθε κάρτας).
- Η μέθοδος display() εμφανίζει τη φιγούρα και το είδος της κάρτας. Η μέθοδος pontoi() επιστρέφει την αξία σε πόντους της κάρτας.
- Η μέθοδος __gt__() υπερφορτώνει τον τελεστή > για την κλάση karta, ώστε όταν συγκρίνονται δυο κάρτες να θεωρείται μεγαλύτερη αυτή που έχει μεγαλύτερη αξία πόντων.

❸ Ορίζουμε την κλάση trapoula από την οποία θα δημιουργούμε αντικείμενα-τράπουλες από τα οποία θα αποτελείται η στοίβα με την οποία θα παίζουμε το παιχνίδι.

- Η μέθοδος δόμησης της κλάσης δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου το χρώμα και μια λίστα με τις κάρτες που θα περιέχει η τράπουλα. Η παράμετρος ar της μεθόδου δόμησης καθορίζει το πλήθος των «τραπουλών» (σετ των 52 καρτών της τράπουλας) που θα περιέχει κάθε αντικείμενο της κλάσης, ενώ η παράμετρος x το χρώμα της.
- Η μέθοδος display() εμφανίζει όλες τις κάρτες που περιέχει μια τράπουλα.

- Η μέθοδος suffle() «ανακατεύει» τις κάρτες της στοίβας με κάποιο τυχαίο τρόπο, διαφορετικά θα ήταν πάντα τοποθετημένες με την ίδια σειρά. Η τεχνική βάσει της οποίας γίνεται το ανακάτεμα είναι η εξής: Επιλέγουμε δύο τυχαίες κάρτες από τη στοίβα και τις αντιμεταθέτουμε. Αν αυτό γίνει πολλές φορές οι κάρτες της στοίβας θα ανακατευτούν και θα τοποθετηθούν σε τυχαίες θέσεις. Η επιλογή των τυχαίων καρτών γίνεται με χρήση της συνάρτησης βιβλιοθήκης randint() του αρθρώματος random.
- Η μέθοδος get_one() επιστρέφει την κάρτα που βρίσκεται στην κορυφή ενός αντικειμένου-τράπουλα και ταυτόχρονα την απομακρύνει από τη στοίβα.
- Η μέθοδος __add__() υπερφορτώνει τον τελεστή + για την κλάση trapoula, ώστε όταν προστίθενται δύο τράπουλες να ενώνονται οι κάρτες τους και να συνδυάζεται το χρώμα τους.
- Η μέθοδος __gt__() υπερφορτώνει τον τελεστή > για την κλάση trapoula, ώστε όταν συγκρίνονται δύο τράπουλες να θεωρείται μεγαλύτερη η τράπουλα που έχει περισσότερες κάρτες.
- Η μέθοδος __len__() υπερφορτώνει την ενσωματωμένη συνάρτηση len() ώστε όταν χρησιμοποιείται με όρισμα ένα αντικείμενο-τράπουλα να επιστρέφει το πλήθος των καρτών της.

❹ Ορίζουμε την κλάση mana από την οποία θα δημιουργούμε αντικείμενα-μάνες τα οποία θα διαθέτουν στοίβες αποτελούμενες από μια ή περισσότερες τράπουλες και θα έχουν τη δυνατότητα να παίζουν 31 με κάποιον «παίκτη». από τα οποία θα αποτελείται η στοίβα με την οποία θα παίζουμε το παιχνίδι.

- Η μέθοδος δόμησης __init__() της κλάσης δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου μια λίστα με τις κάρτες που «τραβάει», τον αριθμό της κάρτας που θα κρατήσει «κρυφή», το ποσό που έχει συνολικά κερδίσει ή χάσει στο παιχνίδι, καθώς και μια στοίβα η οποία αποτελείται από μια ή περισσότερες τράπουλες με την οποία θα παίξει το παιχνίδι. Επίσης, η μέθοδος «ανακατεύει» τη στοίβα που δημιούργησε. Η παράμετρος x καθορίζει το χρώμα και η παράμετρος ar το πλήθος των τραπουλών που θα διαθέτει αρχικά η στοίβα των καρτών της «μάνας».
- Η μέθοδος display_all() εμφανίζει όλες τις κάρτες που έχει τραβήξει η «μάνα» καθώς και το σύνολο των πόντων τους.
- Η μέθοδος display_all_but_one() εμφανίζει όλες τις κάρτες που έχει τραβήξει η «μάνα» εκτός από μια που κρατάει «κρυφή» καθώς και το σύνολο των φανερών πόντων της.
- Η μέθοδος vres_kryfi() αποφασίζει και επιστρέφει τον αριθμό της κάρτας που πρέπει να κρατήσει κρυφή η μάνα. Στην περίπτωσή μας πάντα κρατάει ως κρυφή την κάρτα με την μικρότερη αξία.
- Η μέθοδος trava_karta() χρησιμοποιεί τη μέθοδο get_one() της κλάσης trapoula για να «τραβήξει» η μάνα μια κάρτα από τη στοίβα. Η κάρτα αυτή προστίθεται στη λίστα kartes με τις κάρτες που έχει τραβήξει.
- Η μέθοδος dose_karta() χρησιμοποιεί τη μέθοδο get_one() της κλάσης trapoula για να «τραβήξει» μια κάρτα από τη στοίβα την οποία επιστρέφει ως τιμή. Η μέθοδος αυτή χρησιμοποιείται από τον «παίκτη» για να ζητήσει μια κάρτα από τη «μάνα», την οποία θα προσθέσει στις δικές του κάρτες.
- Η μέθοδος pontoi() επιστρέφει το σύνολο των πόντων από τις κάρτες που έχει τραβήξει η «μάνα».
- Η μέθοδος kerdisa() προσθέτει την τιμή της παραμέτρου p στο σύνολο του ποσού που διαθέτει η «μάνα».
- Η μέθοδος exasa() αφαιρεί την τιμή της παραμέτρου p από το σύνολο του ποσού που διαθέτει η «μάνα».
- Η μέθοδος clear_hand() διαγράφει από τη λίστα kartes όλες τις κάρτες που έχει τραβήξει η «μάνα», αφήνοντάς την κενή.

- Τη καρδιά του προγράμματος αποτελεί η μέθοδος pexe_31() της κλάσης mana η οποία υλοποιεί τη λειτουργία του παιχνιδιού, σύμφωνα με την εκφώνηση. Στη μέθοδο μεταβιβάζεται το αντικείμενο-παίκτης p με τον οποίο θα παίξει η «μάνα».
  - Αρχικά η μάνα τραβάει μία κάρτα από τη στοίβα και την δίνει στον παίχτη ο οποίος ποντάρει το ποσό που επιθυμεί. Η κάρτα αυτή αφαιρείται από τη στοίβα και προστίθεται στις κάρτες του παίκτη.
  - Τώρα ο παίκτης πρέπει να ποντάρει. Αποδεκτά ποντάρισματα είναι αριθμητικές τιμές από το 1 μέχρι το 10.
  - Στη συνέχεια είναι σειρά της «μάνας» να τραβήξει κάρτες. Η «μάνα» τραβάει κάρτες, μέχρι να βγάλει άθροισμα πόντων μεγαλύτερο από 23, ή 14 ή 31 ή να καεί (περισσότερους από 31 πόντους). Στην περίπτωση που συμπληρώσει 31 πόντους κερδίζει το ποντάρισμα και στην περίπτωση που καεί το χάνει.
  - Αν η «μάνα» κερδίσει δείχνει όλες της τις κάρτες.
  - Αν δεν κερδίσει ή δεν χάσει η «μάνα», δείχνει όλες της τις κάρτες εκτός από μια που την κρατάει κρυφή.
  - Τώρα είναι σειρά του παίκτη να τραβήξει κάρτες από τη στοίβα. Τραβάει κάρτες μέχρις ότου αποφασίσει να σταματήσει, να συμπληρώσει 31 πόντους οπότε κερδίζει, ή να καεί οπότε χάνει.
  - Αν κανένας από τους δύο δεν έχει κερδίσει ή δεν έχει καεί, τότε συγκρίνονται οι πόντοι των δύο παικτών και αποφασίζεται ποιος από τους δύο κερδίζει. Σε περίπτωση ισοβαθμίας κερδίζει η «μάνα» και εμφανίζει τις κάρτες της.
  - Εμφανίζεται το συνολικό ποσό που κερδίζει ή χάνει ο παίκτης μέχρι στιγμής και το παιχνίδι συνεχίζεται με τον επόμενο γύρο.
  - Στην περίπτωση που δεν έχουν μείνει αρκετές κάρτες στη στοίβα για έναν νέο γύρο, ζητείται από τον παίκτη να αποφασίσει αν θα προστεθούν στη στοίβα κάρτες από μια ακόμα τράπουλα.
  - Σε αυτή την περίπτωση η στοίβα συμπληρώνεται με τη νέα τράπουλα, ανακατεύεται και το παιχνίδι συνεχίζεται, διαφορετικά το παιχνίδι σταματά.
  - Τέλος, η μέθοδος επιστρέφει ως τιμή το συνολικό ποσό που κέρδισε ή έχασε ο παίκτης.

❺ Ορίζουμε την κλάση paiktis από την οποία θα δημιουργούμε αντικείμενα-παίκτες τα οποία θα παίζουν με μια «μάνα» 31.

- Η μέθοδος δόμησης __init__() της κλάσης δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου μια λίστα με τις κάρτες που «τραβάει» ο παίκτης, το όνομά του καθώς και το ποσό που έχει συνολικά κερδίσει ή χάσει στο παιχνίδι.
- Η μέθοδος display() εμφανίζει όλες τις κάρτες που έχει τραβήξει ο παίκτης.
- Με τη μέθοδο zito_karta() ο παίκτης ζητάει κάρτα από τη μάνα m. Η κάρτα προστίθεται στη λίστα kartes με τις κάρτες που έχει τραβήξει.
- Με τη μέθοδο moy_edosan_karta() ο παίκτης παίρνει μια κάρτα krt που του δίνουν. Η κάρτα προστίθεται στη λίστα kartes με τις κάρτες που έχει τραβήξει.
- Η μέθοδος pontoi() επιστρέφει το σύνολο των πόντων από τις κάρτες που έχει τραβήξει ο παίκτης.
- Η μέθοδος kerdisa() προσθέτει την τιμή της παραμέτρου p στο σύνολο του ποσού που διαθέτει ο παίκτης.
- Η μέθοδος exasa() αφαιρεί την τιμή της παραμέτρου p από το σύνολο του ποσού που διαθέτει ο παίκτης.

- Η μέθοδος clear_hand() διαγράφει από τη λίστα kartes όλες τις κάρτες που έχει τραβήξει ο παίκτης, αφήνοντάς την κενή.

❻ Στο κυρίως πρόγραμμα μένει να υλοποιήσουμε ελάχιστα πράγματα.

- Αρχικά δημιουργείται ένα αντικείμενο-μάνα με μια στοίβα από δύο κόκκινες τράπουλες, το οποίο ανατίθεται στη μεταβλητή m1.
- Στη συνέχεια δημιουργείται ένα αντικείμενο-παίκτης με όνομα "Νίκος" το οποίο ανατίθεται στη μεταβλητή p1.
- Η «μάνα» m1 παίζει τριανταμία με τον παίκτη p1. Το συνολικό ποσό που θα κερδίσει ή θα χάσει ο παίκτης ανατίθεται στη μεταβλητή synolo.
- Τέλος, εμφανίζει τα συνολικά αποτελέσματα του παιχνιδιού.

Κώδικας

e6.py

```
import random

class karta:
    def __init__(self,f,e,t):
        self.figoura=f
        self.eidos=e
        self.timi=t

    def display(self):
        print(self.figoura,self.eidos)

    def pontoi(self):
        return self.timi

    def __gt__(self, o):
        if self.timi>o.timi:
            return True
        else:
            return False

class trapoula:
    def __init__(self,x,ar):
        self.xroma=x
        self.kartes=[]
        fig=('Άσσος','Δύο','Τρία','Τέσσερα','Πέντε','Έξι','Επτά','Οκτώ',
            'Εννέα','Δέκα','Βαλές','Ντάμα','Ρήγας')
        eidi=('Καρό','Κούπα','Μπαστούνι','Σπαθί')

        for i in range(13):
            for j in range(4):
                if i<9:
                    axia=i+1
                else:
                    axia=10
                k=karta(fig[i],eidi[j],axia)
                self.kartes.append(k)
        self.kartes=self.kartes*2
        print('Μόλις δημιούργησα μια {} τράπουλα με {} κάρτες'.format(x,len(self.kartes)))

    def display(self):
        print('Τράπουλα :',self.xroma,'  Κάρτες :',len(self.kartes))
        print('=================================================')
        for i in self.kartes:
            i.display()
```

❷ Η μέθοδος δόμησης της κλάσης **karta** δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου τη φιγούρα (Βαλές, Δέκα, Άσσος, κ.λπ.), το είδος (Κούπα, Καρό, κ.λπ.) και την τιμή (δηλαδή τους πόντους κάθε κάρτας).

❷ Η μέθοδος **display()** εμφανίζει τη φιγούρα και το είδος της κάρτας.

❷ Η μέθοδος **pontoi()** επιστρέφει την αξία σε πόντους της κάρτας.

❷ Η μέθοδος **__gt__()** υπερφορτώνει τον τελεστή **>** για την κλάση **karta**, ώστε να θεωρείται μεγαλύτερη μια κάρτα όταν έχει μεγαλύτερη αξία πόντων.

❸ Η μέθοδος δόμησης της κλάσης **trapoula** δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου το χρώμα και μια λίστα με τις κάρτες που θα περιέχει η τράπουλα. Η παράμετρος **ar** της μεθόδου δόμησης καθορίζει το πλήθος των «τραπουλών» (σετ των 52 καρτών της τράπουλας) που θα περιέχει κάθε αντικείμενο της κλάσης και η παράμετρος **x** το χρώμα της.

❸ Η στη μεταβλητή στιγμιότυπου **karta** έχει ανατεθεί μια κενή λίστα. Σε αυτήν προστίθενται 52 αντικείμενα-κάρτες με όλους τους συνδυασμούς που διαθέτει μια κανονική τράπουλα.

❸ Η μέθοδος **display()** εμφανίζει όλες τις κάρτες που περιέχει μια τράπουλα.

```
    def suffle(self):
        for i in range(5*len(self.kartes)):
            a=random.randint(0,len(self.kartes)-1)
            b=random.randint(0,len(self.kartes)-1)
            self.kartes[a],self.kartes[b]=self.kartes[b],self.kartes[a]
        print('Μόλις ανακάτεψα την {} τράπουλα με {} κάρτες'.format(self.xroma,len(self)))

    def get_one(self):
        krt=self.kartes.pop(0)
        return krt

    def __add__(self, o):
        nt=trapoula(self.xroma+' & '+o.xroma)
        nt.kartes=self.kartes+o.kartes
        return nt

    def __gt__(self, o):
        if len(self.kartes)>len(o.kartes):
            return True
        else:
            return False

    def __len__(self):
        return len(self.kartes)

class mana:
    def __init__(self,x,ar):
        self.kartes=[]
        self.kryfi_karta=None
        self.poso=0
        self.stoiva=trapoula(x,ar)
        self.stoiva.suffle()

    def display_all(self):
        print('\nΚάρτες Μάνας')
        print('================')
        s=0
        for k in self.kartes:
            k.display()
            s=s+k.pontoi()
        print('=================')
        print('Σύνολο πόντων:',s)

    def display_all_but_one(self,kryfi):
        print('\nΦανερές Κάρτες Μάνας')
        print('====================')
        s=0
        for i in range(len(self.kartes)):
            if i!=kryfi:
                self.kartes[i].display()
                s=s+self.kartes[i].pontoi()
            else:
                print('??????????????')
        print('====================')
        print('Σύνολο φανερών πόντων:',s)

    def vres_kryfi(self):
        maxpos=self.kartes.index(max(self.kartes))
        minpos=self.kartes.index(min(self.kartes))
        return minpos
```

❸ Η μέθοδος **suffle()** «ανακατεύει» τις κάρτες ενός αντικειμένου τράπουλα με τυχαίο τρόπο.

❸ Η μέθοδος **get_one()** επιστρέφει την κάρτα που βρίσκεται στην κορυφή ενός αντικειμένου-τράπουλα και ταυτόχρονα την απομακρύνει από τη στοίβα.

❸ Η μέθοδος **__add__()** υπερφορτώνει τον τελεστή **+** για την κλάση **trapoula**, ώστε όταν προστίθενται δύο τράπουλες να ενώνονται οι κάρτες τους και να συνδυάζεται το χρώμα τους.

❸ Η μέθοδος **__gt__()** υπερφορτώνει τον τελεστή **>** για την κλάση **trapoula**, ώστε να θεωρείται μεγαλύτερη μια τράπουλα όταν έχει περισσότερες κάρτες.

❸ Η μέθοδος **__len__()** υπερφορτώνει την ενσωματωμένη συνάρτηση **len()** ώστε όταν χρησιμοποιείται με όρισμα ένα αντικείμενο-τράπουλα να επιστρέφει το πλήθος των καρτών της.

❹ Η μέθοδος δόμησης της κλάσης **mana** δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου μια λίστα με τις κάρτες που «τραβάει» τον αριθμό της κάρτας που θα κρατήσει «κρυφή» το ποσό που έχει συνολικά κερδίσει ή χάσει στο παιχνίδι καθώς και μια στοίβα η οποία αποτελείται από μια ή περισσότερες τράπουλες με την οποία θα παίξει το παιχνίδι. Επίσης, η μέθοδος «ανακατεύει» τη στοίβα που δημιούργησε. Η παράμετρος **x** καθορίζει το χρώμα και η παράμετρος **ar** το πλήθος των τραπουλών που θα διαθέτει η στοίβα των καρτών της «μάνας».

❹ Η μέθοδος **display_all()** εμφανίζει όλες τις κάρτες που έχει τραβήξει η «μάνα» καθώς και το σύνολο των πόντων τους.

❹ Η μέθοδος **display_all_but_one()** εμφανίζει όλες τις κάρτες που έχει τραβήξει η «μάνα» εκτός από μια που κρατάει «κρυφή» καθώς και το σύνολο των φανερών πόντων της.

❹ Η μέθοδος **vres_kryfi()** επιστρέφει τον αριθμό της κάρτας που πρέπει να κρατήσει κρυφή η μάνα. Στην περίπτωσή μας πάντα κρατάει ως κρυφή την κάρτα με την μικρότερη αξία.

```
def trava_karta(self):
    krt=self.stoiva.get_one()
    self.kartes.append(krt)

def dose_karta(self):
    krt=self.stoiva.get_one()
    return krt

def pontoi(self):
    s=0
    for i in self.kartes:
        s=s+i.pontoi()
    return s

def kerdisa(self,p):
    self.poso=self.poso+p

def exasa(self,p):
    self.poso=self.poso-p

def clear_hand(self):
    self.kartes.clear()

def pexe_31(self,p):
    gyroi=1
    while True:
        print('\n########## ΓΥΡΟΣ Νο {} - Κάρτες στη στοίβα {}
              ##########\n'.format(gyroi,len(self.stoiva)))
        print('Μάνα: Τώρα δίνω στον παικτη {} την πρώτη κάρτα'.format(p.onoma))
        k=self.dose_karta()
        k.display()
        p.moy_edosan_karta(k)
        print('Σύνολο πόντων:',p.pontoi())
        while True:
            bet_str=input('{} δώσε το ποντάρισμά σου (1~10 euro) -> '.format(p.onoma))
            if bet_str.isnumeric():
                bet=int(bet_str)
                if bet>=1 and bet<=10:
                    break
        print('\nΤώρα τραβάει η μάνα')


        while self.pontoi()<=23:
            self.trava_karta()
            if self.pontoi()==14: break

        if self.pontoi()==31:
            print('Τριανταμία ... σε κέρδισα')
            self.display_all()
            self.kerdisa(bet)
            p.exasa(bet)

        elif self.pontoi()>31:
            print('Οπς ... κάηκα με',self.pontoi(),'πόντους')
            self.exasa(bet)
            p.kerdisa(bet)
        else:
            print('Μάνα: Τράβηξα {} κάρτες'.format(len(self.kartes)))
            self.display_all_but_one(self.vres_kryfi())
            print('\nΤώρα δίνω κάρτες στον παικτη {}'.format(p.onoma))
```

❹ Η μέθοδος **trava_karta()** χρησιμοποιεί τη μέθοδο **get_one()** της κλάσης **trapoula** για να «τραβήξει» η μάνα μια κάρτα από τη στοίβα. Η κάρτα αυτή προστίθεται στη λίστα **kartes** με τις κάρτες που έχει τραβήξει ένα αντικείμενο-μάνα.

❹ Η μέθοδος **dose_karta()** χρησιμοποιεί τη μέθοδο **get_one()** της κλάσης **trapoula** για να «τραβήξει» μια κάρτα από τη στοίβα την οποία επιστρέφει ως τιμή.

❹ Η μέθοδος **pontoi()** επιστρέφει το σύνολο των πόντων από τις κάρτες που έχει τραβήξει η «μάνα».

❹ Η μέθοδος **kerdisa()** προσθέτει την τιμή της παραμέτρου **p** στο σύνολο του ποσού που διαθέτει η «μάνα».

❹ Η μέθοδος **exasa()** αφαιρεί την τιμή της παραμέτρου **p** από το σύνολο του ποσού που διαθέτει η «μάνα».

❹ Η μέθοδος **clear_hand()** διαγράφει από τη λίστα **kartes** όλες τις κάρτες που έχει τραβήξει η «μάνα», αφήνοντάς την κενή.

❹ Η μέθοδος **pexe_31()** δέχεται ως παράμετρο ένα αντικείμενο-παίκτη **p**, με τον οποίο «παίζει» τριανταμία.

❹ Η «μάνα» τραβάει μια κάρτα **k**, την εμφανίζει και τη δίνει στον παίκτη **p**.

❹ Ζητείται από τον παίκτη να ποντάρει. Αποδεκτές είναι μόνο αριθμητικές τιμές από 1 μέχρι 10.

❹ «Τραβάει» η μάνα κάρτες από τη στοίβα μέχρι να αποκτήσει σύνολο πόντων >23. Σταματάει επίσης όταν έχει βγάλει σύνολο πόντων 14.

❹ Αν τελικά η «μάνα» έβγαλε 31, δείχνει όλες της τις κάρτες, κερδίζει το ποντάρισμα **bet** ενώ ο παίκτης **p** το χάνει.

❹ Αν η «μάνα» έβγαλε >31 τότε κάηκε, χάνει το ποντάρισμα **bet** ενώ ο παίκτης **p** το κερδίζει.

❹ Διαφορετικά, εμφανίζει τις κάρτες που τράβηξε, εκτός από μια «κρυφή» κάρτα.

```
            while True:
                k=p.zito_karta(self)
                k.display()

                print('Σύνολο πόντων:',p.pontoi())

                if p.pontoi()>=31: break
                ans=input('{} πάτησε ENTER για επόμενη κάρτα - οτιδήποτε άλλο για
                          σταμάτημα  >'.format(p.onoma))
                if ans!='': break
            if p.pontoi()==31:
                print('\nΜΠΡΑΒΟ ΤΡΙΑΝΤΑΜΙΑ κέρδισες!!!')
                self.exasa(bet)
                p.kerdisa(bet)
            elif p.pontoi()>31:
                print('\nΔυστυχώς  κάηκες με',p.pontoi())
                self.display_all()
                self.kerdisa(bet)
                p.exasa(bet)
            elif self.pontoi()==14:
                print('\nΔυστυχώς σε κέρδισα γιατί έχω 14')
                self.display_all()
                self.kerdisa(bet)
                p.exasa(bet)
            elif p.pontoi()==14:
                print('\nΜπράβο με κέρδισες έχεις 14')
                self.exasa(bet)
                p.kerdisa(bet)
            elif p.pontoi()>self.pontoi():
                print('\nΜπράβο με κέρδισες έχεις {} και έχω {}
                         πόντους'.format(p.pontoi(),self.pontoi()))
                self.exasa(bet)
                p.kerdisa(bet)
            elif p.pontoi()==self.pontoi():
                print('\nΈχουμε και οι δύο {} πόντους ... αλλά επειδή είμαι
                        η μάνα σε κερδίζω :-)'.format(self.pontoi()))
                self.display_all()
                self.kerdisa(bet)
                p.exasa(bet)
            else:
                print('\nΔυστυχώς σε κέρδισα έχεις {} και έχω {}
                        πόντους'.format(p.pontoi(),self.pontoi()))
                self.display_all()
                self.kerdisa(bet)
                p.exasa(bet)
        if (p.poso>0):
            print('\n****** {} μέχρι τώρα κερδίζεις {} euro '.format(p.onoma,p.poso))
        elif (p.poso<0):
            print('\n****** {} μέχρι τώρα χάνεις {} euro '.format(p.onoma,-p.poso))
        else:
            print('\n****** {} μέχρι τώρα δεν κερδίζεις ούτε χάνεις ******')
        gyroi=gyroi+1
        if len(self.stoiva)<=10:
            print('Μάνα: Δεν μπορεί να γίνει άλλος γύρος παρέμειναν μόνο {}
                    κάρτες στη στοίβα'.format(len(self.stoiva)))
            ans=input('Θέλετε να προστεθεί και άλλη τράπουλα στη στοίβα Ν/Ο >')
            if ans in 'ΝnΝν':
                self.stoiva=self.stoiva+trapoula('NEA',1)
                self.stoiva.suffle()
            else: break
        ans=input('{} πάτησε ENTER για επόμενο γύρο - οτιδήποτε άλλο για
```

❹ Ο παίκτης **p** τώρα ζητάει κάρτες μέχρι να αποφασίσει να σταματήσει, να βγάλει 31 ή να καεί.

❹ Αν ο παίκτης **p** έβγαλε 31, κερδίζει το ποντάρισμα **bet** ενώ ο η «μάνα» το χάνει.

❹ Αν ο παίκτης **p** έβγαλε >31 τότε κάηκε, χάνει το ποντάρισμα **bet** ενώ η «μάνα» το κερδίζει.

❹ Αν η «μάνα» έχει βγάλει 14 και ο παίκτης δεν έχει 31, τότε κερδίζει το ποντάρισμα **bet** ενώ ο παίκτης το χάνει.

❹ Αν ο παίκτης έχει βγάλει 14, τότε κερδίζει το ποντάρισμα **bet** ενώ η «μάνα» το χάνει.

❹ Αν ο παίκτης έχει βγάλει περισσότερους πόντους, τότε κερδίζει το ποντάρισμα **bet** ενώ η «μάνα» το χάνει.

❹ Αν ο παίκτης και «μάνα» έχουν ίδιους πόντους, τότε κερδίζει το ποντάρισμα **bet** η «μάνα», ενώ ο παίκτης το χάνει.

❹ Αν ο παίκτης έχει λιγότερους πόντους, τότε κερδίζει το ποντάρισμα **bet** η «μάνα», ενώ ο παίκτης το χάνει.

❹ Εμφανίζεται το συνολικό ποσό που κερδίζει ή χάνει μέχρι στιγμής ο παίκτης **p**.

❹ Αν οι υπόλοιπες κάρτες στη στοίβα είναι λίγες, ερωτάται ο παίκτης και ανάλογα προστίθεται μια ακόμα τράπουλα στη στοίβα της μάνας.

```
                        σταμάτημα  >'.format(p.onoma))
            if ans!='': break
            self.clear_hand()
            p.clear_hand()
        return p.poso

class paiktis:
    def __init__(self,o):
        self.kartes=[]
        self.onoma=o
        self.poso=0

    def display(self):
        print('\nΚάρτες παίκτη:',self.onoma)
        print('====================')
        for k in self.kartes:
            k.display()

    def zito_karta(self,m):
        krt=m.dose_karta()
        self.kartes.append(krt)
        return krt

    def moy_edosan_karta(self,krt):
        self.kartes.append(krt)

    def pontoi(self):
        s=0
        for i in self.kartes:
            s=s+i.pontoi()
        return s

    def kerdisa(self,p):
        self.poso=self.poso+p

    def exasa(self,p):
        self.poso=self.poso-p

    def clear_hand(self):
        self.kartes.clear()

# Κυρίως πρόγραμμα
m1=mana('Κόκκινη',2)
p1=paiktis('Νίκος')
synolo=m1.pexe_31(p1)
# Εμφάνιση αποτελέσματος
if (synolo>0):
    print('\n****** {} κέρδισες συνολικά {} euro '.format(p1.onoma,synolo))
elif (synolo<0):
    print('\n****** {} έχασες συνολικά {} euro '.format(p1.onoma,-synolo))
else:
    print('\nΟΥΤΕ ΓΑΤΑ ΟΥΤΕ ΖΗΜΙΑ {} ΔΕΝ ΚΕΡΔΙΣΕΣ
          ΑΛΛΑ ΔΕΝ ΕΧΑΣΕΣ ΚΙΟΛΑΣ \n'.format(p1.onoma))
print('##### Ελπίζουμε να σε ξαναδούμε στο καζίνο μας! #####')
```

❹ Αν ο παίκτης επιλέξει να παίξει ακόμα έναν γύρο, τότε διαγράφονται οι κάρτες που έχουν τραβήξει τόσο η "«μάνα» όσο και ο παίκτης και το παιχνίδι συνεχίζεται, διαφορετικά διακόπτεται.

❹ Όταν το παιχνίδι τερματιστεί, η μέθοδος επιστρέφει το συνολικό ποσό που κέρδισε ή έχασε ο παίκτης **p**.

❺ Η μέθοδος δόμησης της κλάσης **paiktis** δημιουργεί ένα στιγμιότυπο (αντικείμενο) της κλάσης με μεταβλητές στιγμιότυπου μια λίστα με τις κάρτες που «τραβάει», το όνομά του καθώς και το ποσό που έχει συνολικά κερδίσει ή χάσει στο παιχνίδι.

❺ Η μέθοδος **display()** εμφανίζει όλες τις κάρτες που έχει τραβήξει ο παίκτης.

❺ Με τη μέθοδο **zito_karta()** ο παίκτης ζητάει κάρτα από τη μάνα m. Η κάρτα προστίθεται στη λίστα **kartes** με τις κάρτες που έχει τραβήξει.

❺ Με τη μέθοδο **moy_edosan_karta()** ο παίκτης παίρνει μια κάρτα **krt** που του δίνουν. Η κάρτα προστίθεται στη λίστα **kartes** με τις κάρτες που έχει τραβήξει.

❺ Η μέθοδος **pontoi()** επιστρέφει το σύνολο των πόντων από τις κάρτες που έχει τραβήξει ο παίκτης.

❺ Η μέθοδος **kerdisa()** προσθέτει την τιμή της παραμέτρου **p** στο σύνολο του ποσού που διαθέτει ο παίκτης.

❺ Η μέθοδος **exasa()** αφαιρεί την τιμή της παραμέτρου **p** από το σύνολο του ποσού που διαθέτει ο παίκτης.

❺ Η μέθοδος **clear_hand()** διαγράφει από τη λίστα **kartes** όλες τις κάρτες που έχει τραβήξει ο παίκτης, αφήνοντάς την κενή.

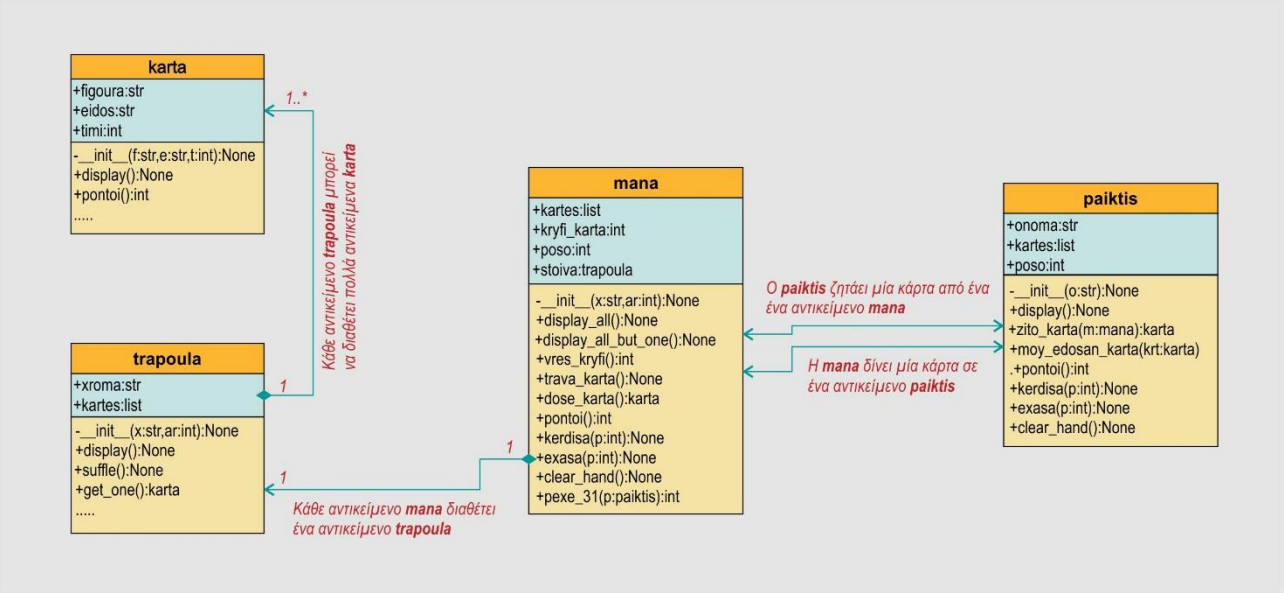
❻ Στη μεταβλητή **m1** ανατίθεται ένα αντικείμενο-μάνα με μια στοίβα από δύο κόκκινες τράπουλες.

❻ Στη μεταβλητή **p1** ανατίθεται ένα αντικείμενο-παίκτης με όνομα "Νίκος".

❻ Η «μάνα» **m1** παίζει τριανταμία με τον παίκτη **p1**. Το συνολικό ποσό που θα κερδίσει ή θα χάσει ο παίκτης ανατίθεται στη μεταβλητή **synolo**.

❻ Εμφανίζει τα συνολικά αποτελέσματα του παιχνιδιού.

Στο σχήμα Ε6.1, της επόμενης σελίδας παρουσιάζεται το UML διάγραμμα των κλάσεων του προγράμματος και αποτυπώνεται η συσχέτιση μεταξύ τους.



Σχήμα Ε6.1 *Συσχέτιση κλάσεων*

Προτάσεις

- Το παραπάνω πρόγραμμα ο κάθε Άσσος έχει πάντα αξία ενός πόντου! Όμως, στην πραγματική τριανταμία ο πρώτος Άσσος μετράει για 1 ή 11, που είναι επιλογή του παίκτη. Σε περίπτωση περισσότερων Άσσων (2,3 ή 4), τότε ο ένας μετράει υποχρεωτικά 11 και οι υπόλοιποι 1. Προσπαθήστε να ενσωματώσετε στο πρόγραμμα αυτή τη δυνατότητα, δεν είναι πολύ εύκολο!
- Το πρόγραμμα επίσης «κρύβει» από τις κάρτες της «μάνας» αυτή με τη μικρότερη αξία. Αυτό δεν είναι πάντα το σωστό στο κανονικό παιχνίδι. Για παράδειγμα αν η «μάνα» έχει 10,10,9 και Άσσο δεν είναι σωστό να κρύψει τον Άσσο, γιατί φανερώνει ότι έχει σίγουρα 30. Καλύτερα θα ήταν να έκρυβε το 9. Με κάποιο τρόπο θα μπορούσε να μπει και αυτή η παράμετρος στο πρόγραμμα.
- Επιπρόσθετα, μπορούμε να βάλουμε και τη δυνατότητα «μπλόφας» από την πλευρά της «μάνας» ώστε να σταματάει ακόμα και αν δεν έχει καλό χαρτί και να φανερώσει τις ανάλογες κάρτες ώστε να μπερδεύει τον παίκτη. Η «μπλόφα» όμως πρέπει να συμβαίνει τυχαία και με κριτήριο τις κάρτες που ήδη διαθέτει.

Έχοντας ως βάση τη στοίβα με τις κάρτες του παραπάνω προγράμματος, μπορούμε να υλοποιήσουμε και άλλα παιχνίδια τράπουλας όπως το blackjack ή το poker.