# Περιγραφή Κλάσεων

* **Research.java**

Κλάση που διαμορφώνει τον τρόπο με τον οποίο χειριζόμαστε τα δεδομένα εισόδου όταν εκείνα είναι διακριτά και αρχικοποιεί τις get() συναρτήσεις για να μπορέσουμε να εκμαιεύσουμε δεδομένα εντός του λογισμικού.

* **ContResearch.java**

Κλάση που διαμορφώνει τον τρόπο με τον οποίο χειριζόμαστε τα δεδομένα εισόδου όταν εκείνα είναι συνεχή και αρχικοποιεί τις get() συναρτήσεις για να μπορέσουμε να εκμαιεύσουμε δεδομένα εντός του λογισμικού.

* **MetaStats.java**

Στην MetaStats.java υλοποιούνται όλες οι χρήσιμες στατιστικές συναρτήσεις για την διεξαγωγή της μετα-ανάλυσης κυρίως για τα διακριτά δεδομένα. Πολλές συναρτήσεις όμως είναι κοινές και στους δύο τύπους δεδομένων όπως γίνεται αντιληπτό και στο άνωθεν flowchart.Οι Codom-, Dom- και Rec-OddsRatio() χρησιμοποιούνται για την εύρεση του odds ratio με βάση το συνεπικρατές, το επικρατές και το υπολειπόμενο μοντέλο αντίστοιχα. Οι Codom-, Dom- και Rec-FixedInvVariance() χρησιμοποιούνται για τον υπολογισμό της αντίστροφης διακύμανσης στο μοντέλο σταθερών επιδράσεων(Fixed Effects Μοντέλο) για κάθε επιμέρους μοντέλο(όπως αναλύθηκαν παραπάνω) ξεχωριστά. Η WeightedMean() υπολογιζει το σταθμισμένο μέσο των ερευνών, ενώ η StdErrorCombined() το συνολικο τυπικό σφάλμα Standard Error που προκύπτει άπο τις έρευνες. Έπειτα μέσω της Limits εντοπίζονται τα όρια για διάστημα εμπιστοσύνης 95\% και μέσω της Zeta υπολογίζουμε τον αριθμό που εισέρχεται στη συνάρτηση Φ(Z). Τέλος, μέσω της Pvalue() υπολογίζεται το τελικό p-value που θα δωθεί στο χρήστη. Οι συναρτήσεις CochransQ() , {TauSquared() και RandomInvVariance() χρησιμοποιούνται συνδυαστικά με τις προηγούμενες συναρτήσεις, όταν ο χρήστης επιλέγει να εκτελέσει τους υπολογισμούς του με το Random Effects Model. Οι τρεις αυτές συναρτήσεις υπολογίζουν το Q του Cochran, το tau2 και τέλος χρησιμοποιόντας αυτά με την RandomInvVariance() προσθέτουν την επιπλέον διακύμανση στη FixedInvVariance().

* **Gaussian.java**

Open source κλάση την οποία χρησιμοποιούμε για να βρούμε την τιμή Φ(Ζ).

* **ContMetaStats.java**

Η μοναδική διαφορά που υπάρχει εν συγκρίσει με την απλή MetaStats είναι πως εδώ χειριζόμαστε συνεχή δεδομένα. Μέσω των Codom-, Dom- και Rec-EffSize() υπολογίζουμε το effect size για κάθε ένα εκ των μοντέλων (χρησιμοποιείται αντι για την odds ratio ) και εν συνεχεία υπολογίζουμε το την αντίστροφη διακύμανση με τις υπόλοιπες τρεις συναρτήσεις. Τα αποτελέσματα που προκύπτουν χρησιμοποιούνται από τις συναρτήσεις της MetaStats, δηλαδή για τις υπόλοιπες μετρικές δεν υπάρχει καμία περαιτέρω αλλαγή.

* **thread\_research.java**

Κλάση για τον χειρισμό των threads όσον αφορά το τι να εκτελέσουν για διακριτά δεδομένα.

* **thread\_cont\_research.java**

Κλάση για τον χειρισμό των threads όσον αφορά το τι να εκτελέσουν για συνεχή δεδομένα.

* **PreferenceHandling.java**

Αρχικά μέσω της FileReading() διαβάζουμε το αρχείο με τις οδηγίες που μας δίνει ο χρήστης. Αφού αποθηκεύουμε τις επιλογές περνάμε στην Choice() όπου ανοίγουμε ένα αρχείο output και ανάλογα με τις επιλογές που έχουμε διαβάσει από την FileReading() εκτελούμε το αντίστοιχο σετ εντολών και γράφουμε τα αποτελέσματα στο output αρχείο. Όλοι οι πιθανοί συνδυασμοί που μπορούν να προκύψουν αναλύονται στο παραπάνω διάγραμμα ροής.

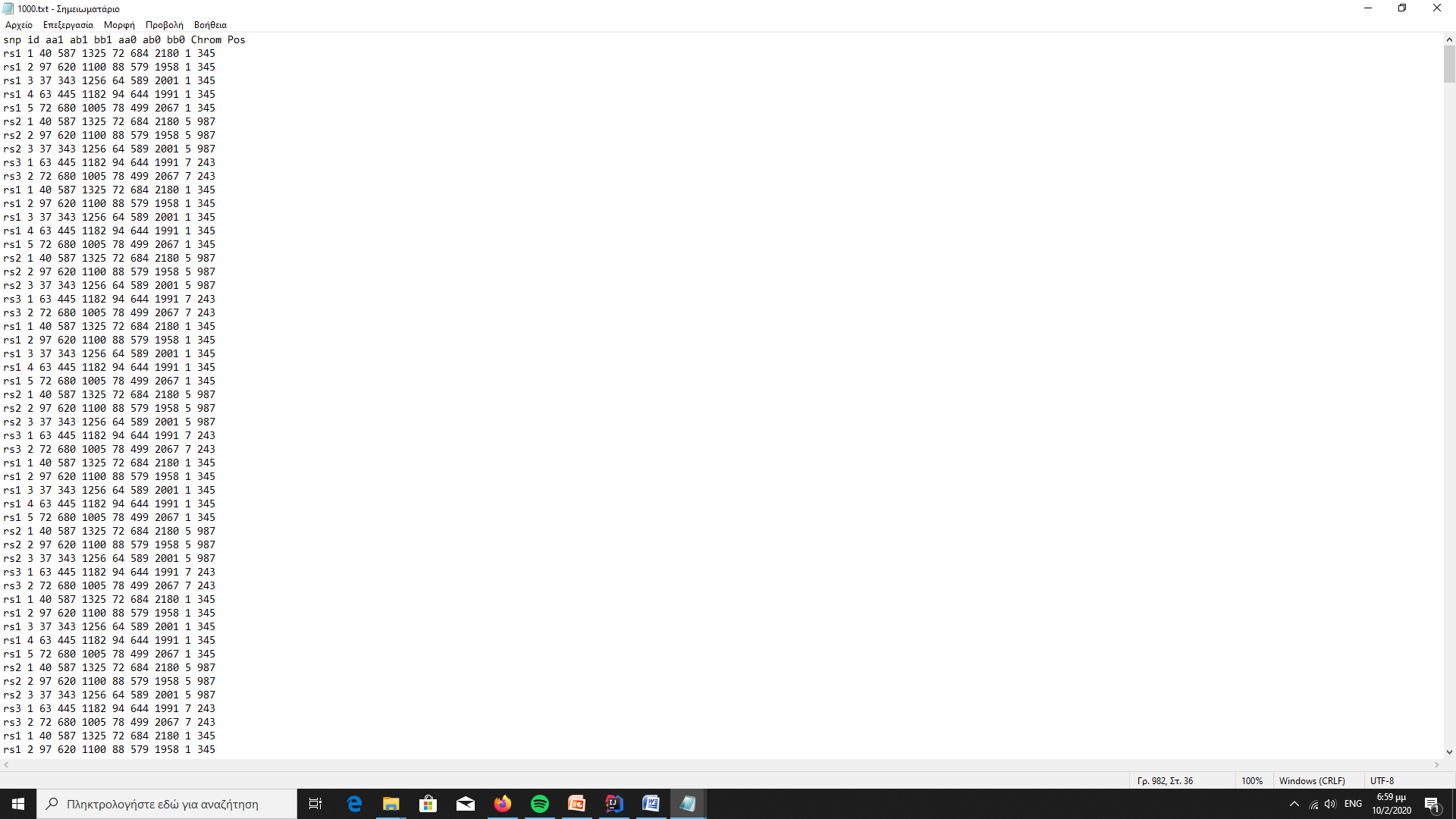
* **DataRead.java**

Εντός της παρούσας κλάσης γίνεται ορισμός λιστών για την αποθήκευση των δεδομένων που εισάγονται από τους χρήστες. Η κλάση αυτή περιέχει δύο συναρτήσεις, την DiscreteDataRead()και την ContinuousDataRead(), οι οποίες χρησιμοποιούνται για διάβασμα σταθερών ή συνεχών δεδομένων αντίστοιχα και για την αποθήκευση των δεδομένων αυτών για μελλοντική χρήση από τις υπόλοιπες κλάσεις του λογισμικού. Η επιλογή μεταξύ των δύο συναρτήσεων, γίνεται αυτόματα από το λογισμικό ανάλογα με την επιλογή που έχει διαβαστεί από το αρχείο επιλογών του χρήστη.

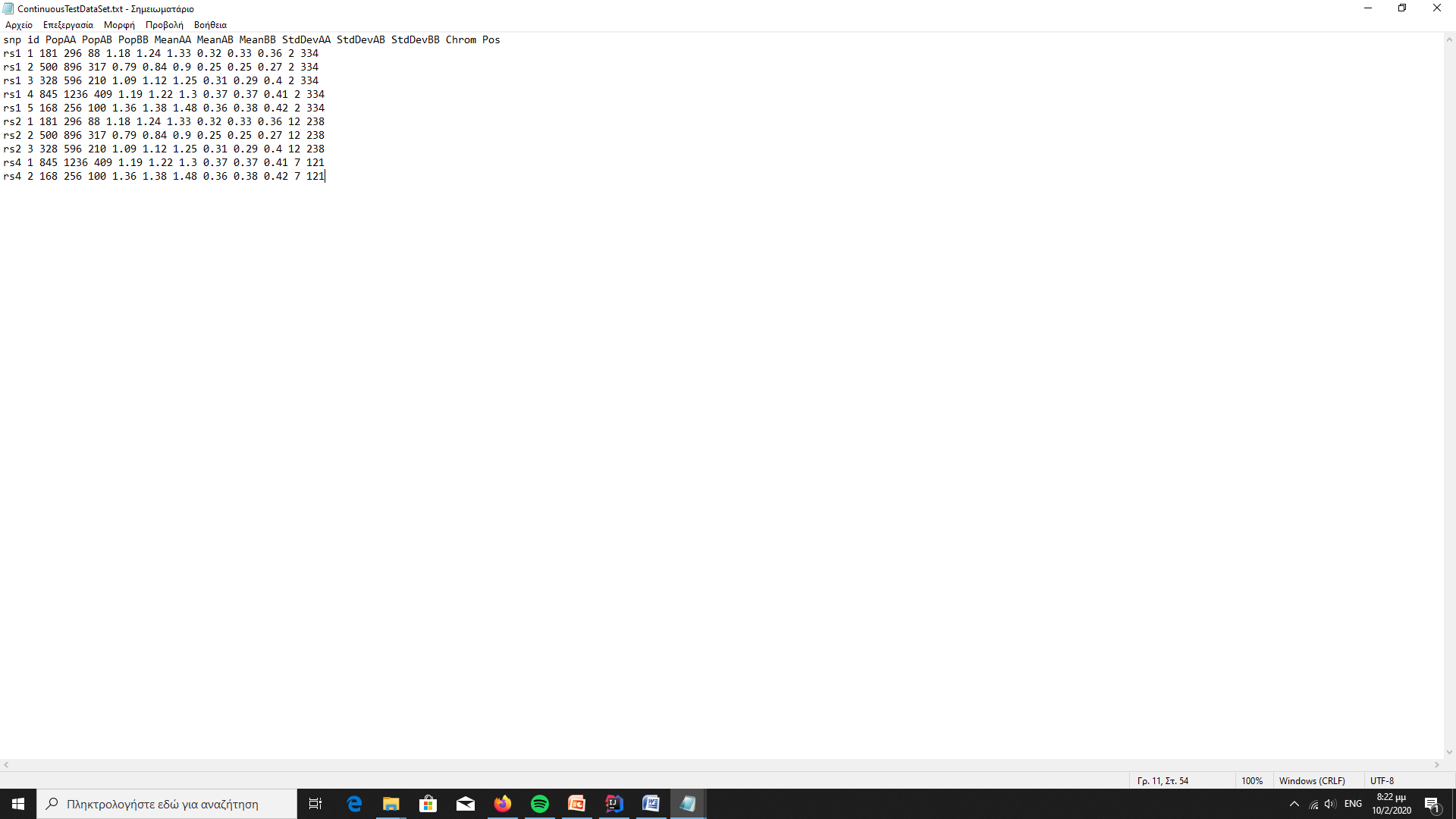
# Δεδομένα και αρχεία εισόδου

Το παρόν λογισμικό είναι ένα command line app σε πρώτη φάση. Για τον λόγο αυτό κατά την εκτέλεση του δέχεται ως πρώτο όρισμα ένα .txt αρχείο με συγκεκριμένη δομή που θα αναλυθεί εν συνεχεία, το οποίο θα περιέχει τις ρυθμίσεις που θέλει να δώσει ο χρήστης έτσι ώστε το πρόγραμμα να εκτελεί τις ανάλογες μεθόδους και ως δεύτερο όρισμα το .txt αρχείο με τα δεδομένα εισόδου τα οποία είναι space-delimited. Για την ορθή λειτουργία του προγράμματος τα αρχεία δεδομένων πρέπει να ακολουθούν την κάτωθι δομή:

* **Διακριτά Δεδομένα**



* **Συνεχή Δεδομένα**



Αντίστοιχα, το αρχείο με τις προτιμήσεις του χρήστη έχει την παρακάτω δομή και αναλόγως τι επιθυμεί να εκτελέσει πληκτρολογεί και το ανάλογο γράμμα.

