

Εργασία για το Facility Game

Μάθημα: Δομές Δεδομένων 2017 - 2018

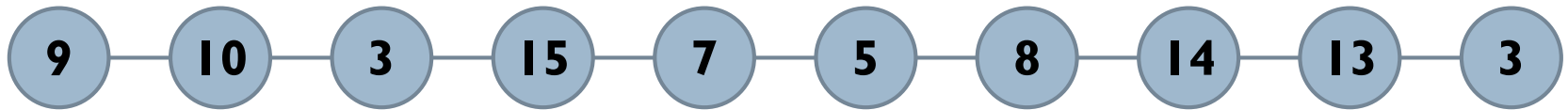


Σωτήρης Γυφτόπουλος

Κανόνες του Facility Game

- ▶ Στο Facility Game υπάρχει ένα σύνολο κόμβων που συνδέονται «σειριακά» και κάθε κόμβος έχει έναν αριθμό – αξία (value).

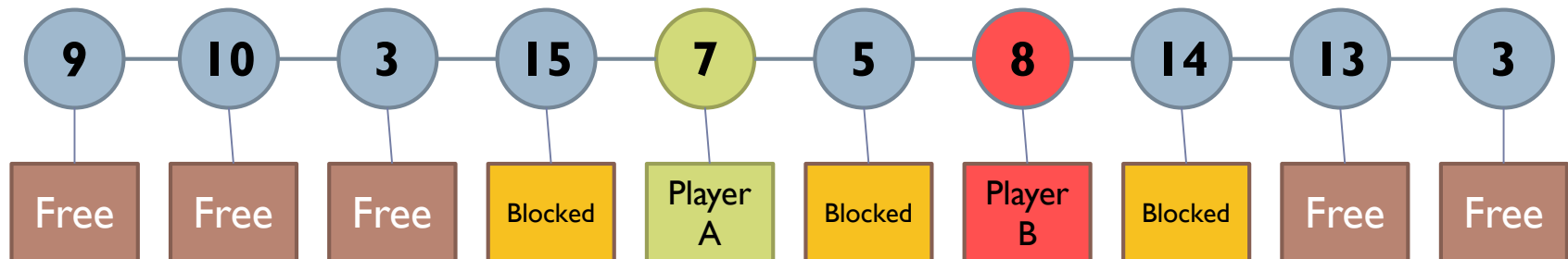
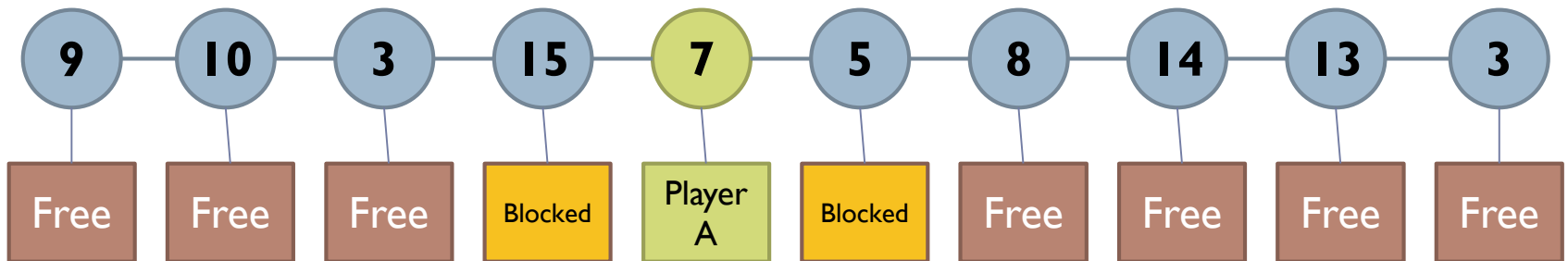
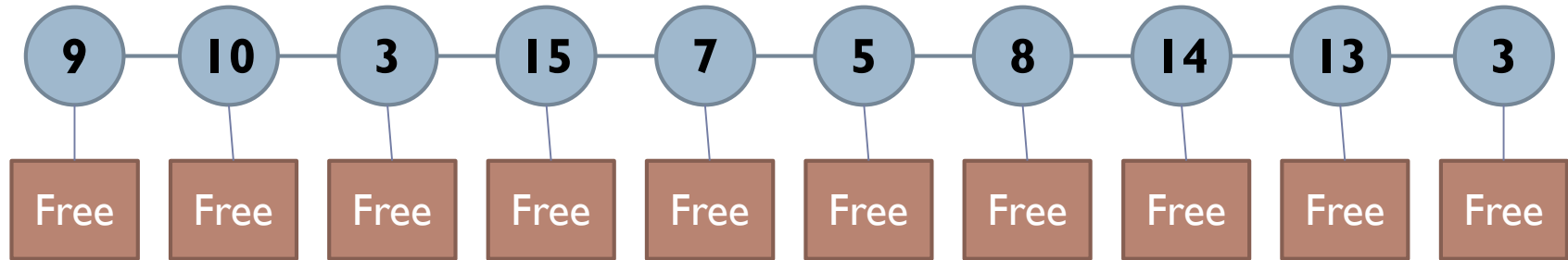
- ▶ π.χ.



- ▶ Στο παιχνίδι υπάρχουν δύο παίκτες που παίζουν εναλλάξ και δεσμεύουν σε κάθε γύρω από έναν κόμβο. Η αξία του κόμβου που δεσμεύουν προστίθεται στο score κάθε παίκτη.
- ▶ Δεν επιτρέπεται να υπάρχουν δύο διπλανοί δεσμευμένοι κόμβοι!



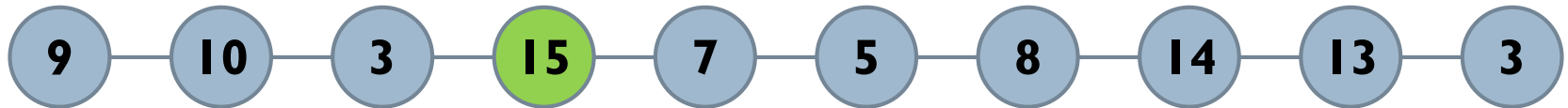
Κανόνες του Facility Game



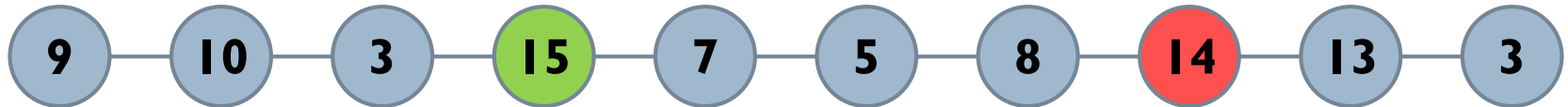
Κανόνες του Facility Game (Normal mode)

► Παίκτης A και Παίκτης B

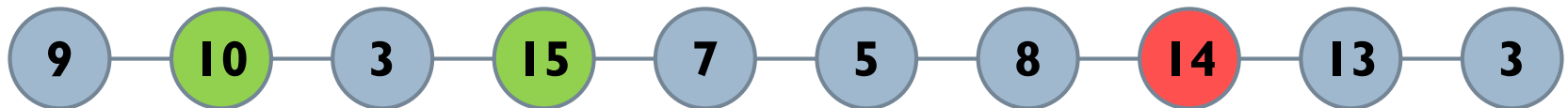
1^{ος} Γύρος: Παίκτης A



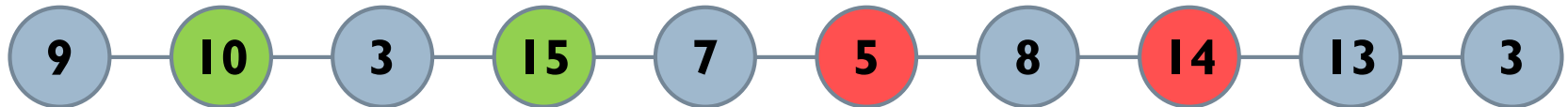
2^{ος} Γύρος: Παίκτης B



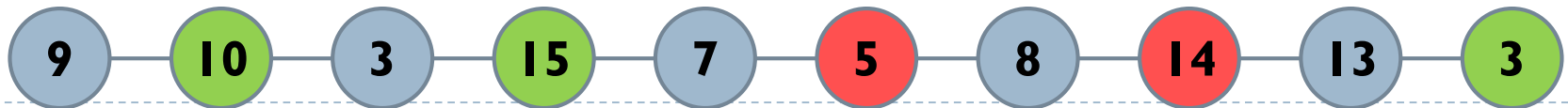
3^{ος} Γύρος: Παίκτης A



4^{ος} Γύρος: Παίκτης B



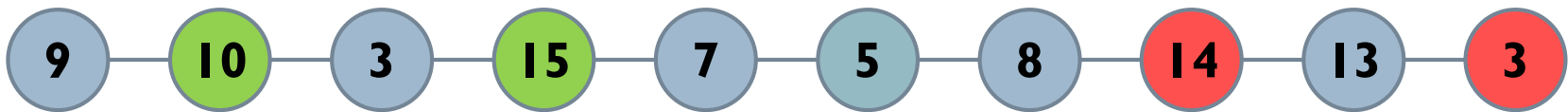
5^{ος} Γύρος: Παίκτης A



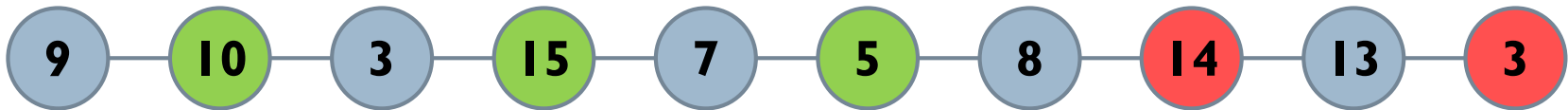
Κανόνες στο Facility Game

- ▶ Αν ένας παίκτης δεσμεύσει τρεις ή περισσότερους «συνεχόμενους» κόμβους τριπλασιάζει τις αξίες αυτών των κόμβων!
- ▶ π.χ.

4^{ος} Γύρος: Παίκτης Β



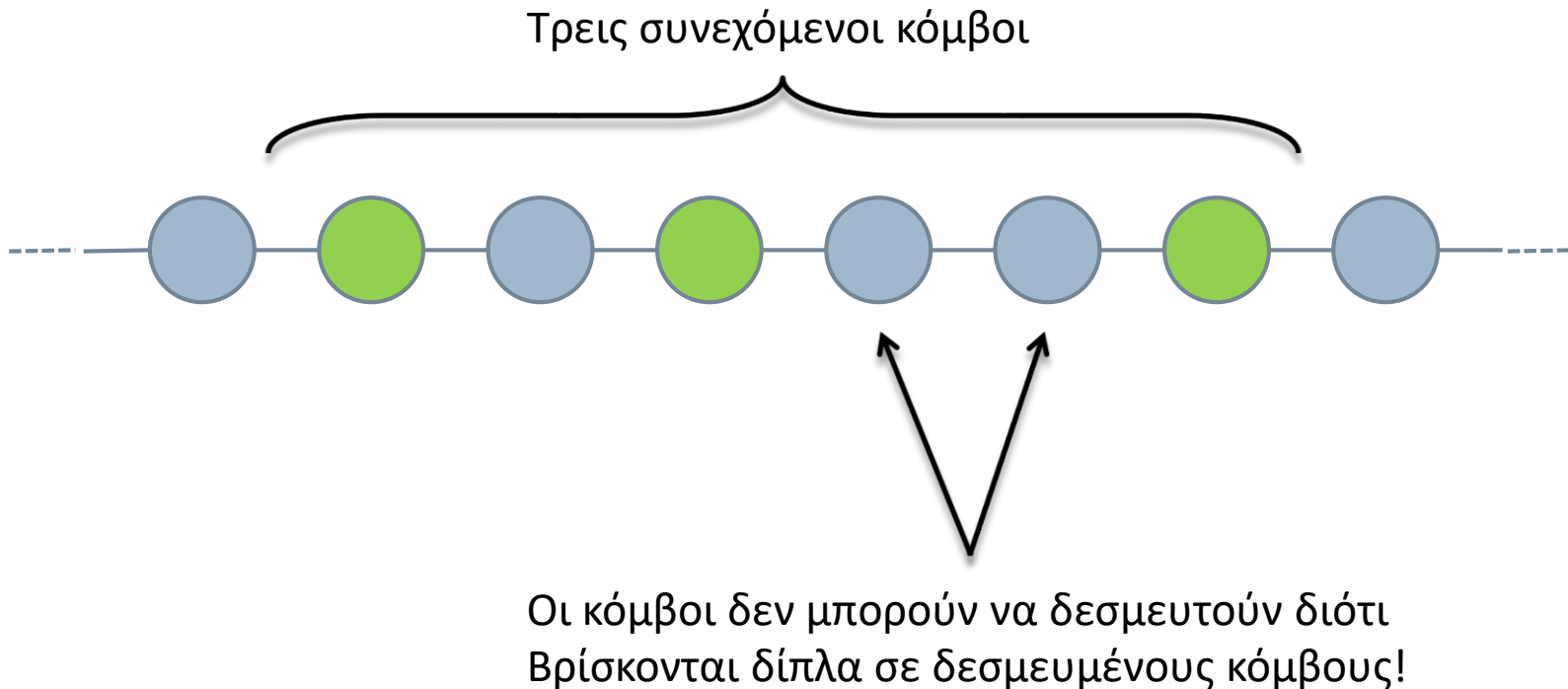
5^{ος} Γύρος: Παίκτης Α



Τρεις συνεχόμενοι κόμβοι

Κανόνες στο Facility Game

- ▶ Μία ακόμα περίπτωση «συνεχόμενων» κόμβων:



- ▶ Καλείστε να υλοποιήσετε τις δικές σας στρατηγικές για να παίξετε στο παιχνίδι.
-



Τύποι παιχνιδιών Facility Game

- ▶ Στο Facility Game υπάρχουν δύο τύποι παιχνιδιών:
 - ▶ NORMAL
 - ▶ NOTRIPLES
- ▶ Στην περίπτωση NORMAL σκοπός κάθε παίκτη είναι να νικήσει τον αντίπαλο επιτυγχάνοντας μεγαλύτερο αποτέλεσμα (score) από αυτόν.
- ▶ Στην περίπτωση NOQUADRUPLES σκοπός του δεύτερου παίκτη είναι να μην επιτρέψει στον πρώτο παίκτη να δεσμεύσει τέσσερις συνεχόμενους κόμβους αλλά να επιτρέπει την δέσμευση τριών συνεχόμενων κόμβων.



Στρατηγική Facility Game (NORMAL mode)

- ▶ Η 2^η Εργασία για το μάθημα Δομές Δεδομένων αφορά στην ανάπτυξη και υλοποίηση μιας στρατηγικής για το παιχνίδι Facility Game τύπου NORMAL.
- ▶ Καλείστε να αναπτύξετε μια στρατηγική, να την υλοποιήσετε σε Java και να την ενσωματώσετε σε ένα υπάρχον project.
- ▶ Η στρατηγική σας πρέπει να επιτυγχάνει όσο το δυνατόν καλύτερα αποτελέσματα (scores) και να νικάει τον έτοιμο αντίπαλο (όπως καθορίζεται και από την εκφώνηση) που περιέχεται στο παιχνίδι.



Στρατηγική Facility Game (NOQUADRUPLES mode)

- ▶ Η 2^η Εργασία για το μάθημα Δομές Δεδομένων αφορά στην ανάπτυξη και υλοποίηση μιας στρατηγικής για το παιχνίδι Facility Game τύπου NOQUADRUPLES.
- ▶ Καλείστε να αναπτύξετε μια στρατηγική, να την υλοποιήσετε σε Java και να την ενσωματώσετε σε ένα υπάρχον project.
- ▶ Η στρατηγική σας πρέπει να δεσμεύει κατάλληλους κόμβους ώστε να μην μπορέσει ο πρώτος παίκτης να δεσμεύσει τέσσερις συνεχόμενους κόμβους αλλά να μπορεί να δεσμεύσει τρεις συνεχόμενους κόμβους.



Βήματα για την εκτέλεση του Facility
Game και την υλοποίηση στρατηγικής

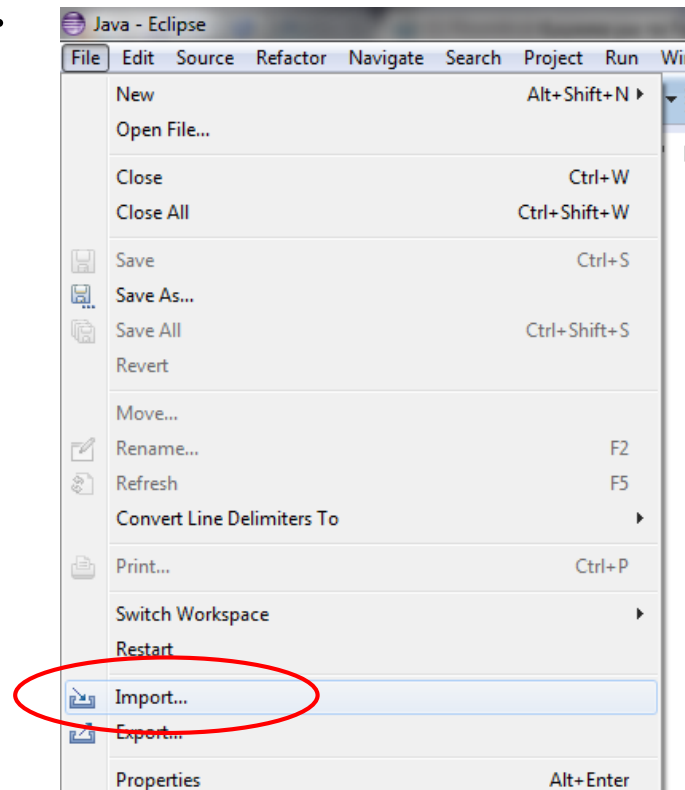
Βήματα για την εκτέλεση του Facility Game και την υλοποίηση στρατηγικής

1. Εισαγωγή του Facility Game στο Eclipse
2. Δημιουργία νέου παίκτη με τη στρατηγική σας ή τροποποίηση υπάρχοντος παίκτη.
3. Εκτέλεση παιχνιδιού με τις κατάλληλες παραμέτρους (είτε **τοπικά** για δοκιμές στον υπολογιστή σας είτε στον **server** όπου καταχωρούνται και τα αποτελέσματα)



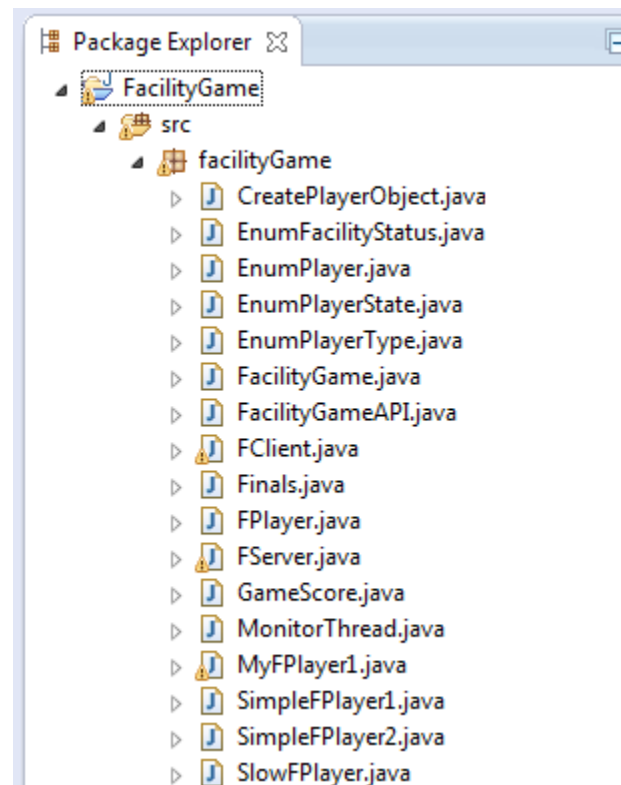
Εισαγωγή του Facility Game

- ▶ Από το μενού File → Import ... και ακολουθούμε τα βήματα για την εισαγωγή του project που κατεβάσατε από το eclass.



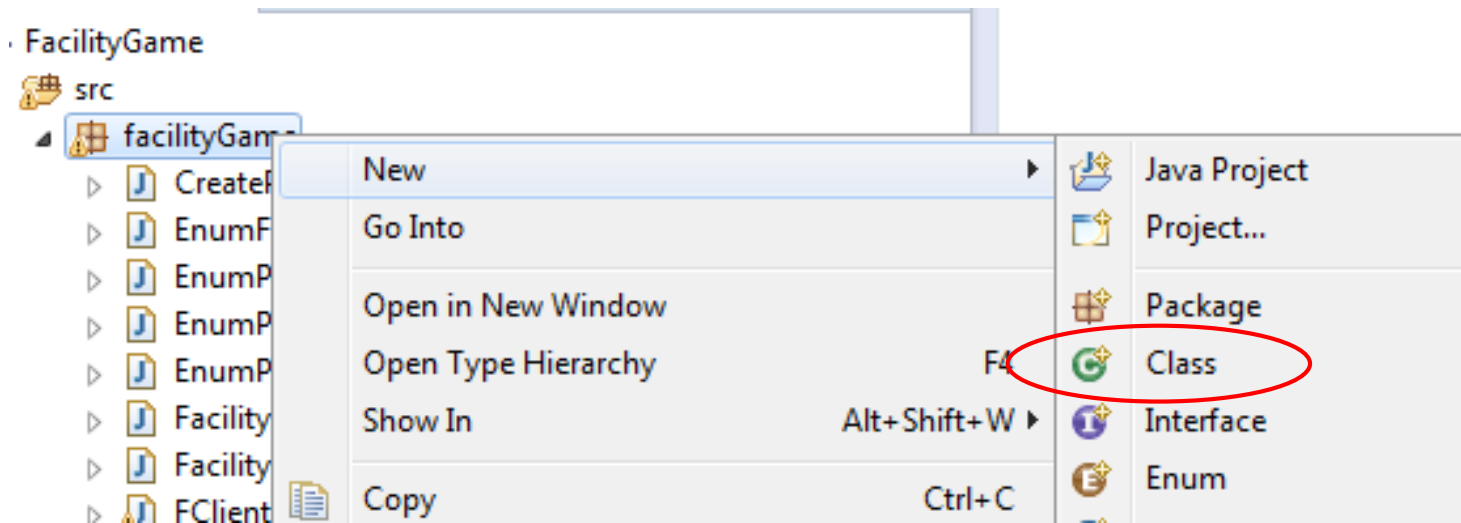
Δημιουργία νέου παίκτη

- ▶ Αφού ανοίξουμε το project FacilityGame, ανοίγουμε τον φάκελο του πηγαίου κώδικα (src) και ανοίγουμε το πακέτο του facilityGame.

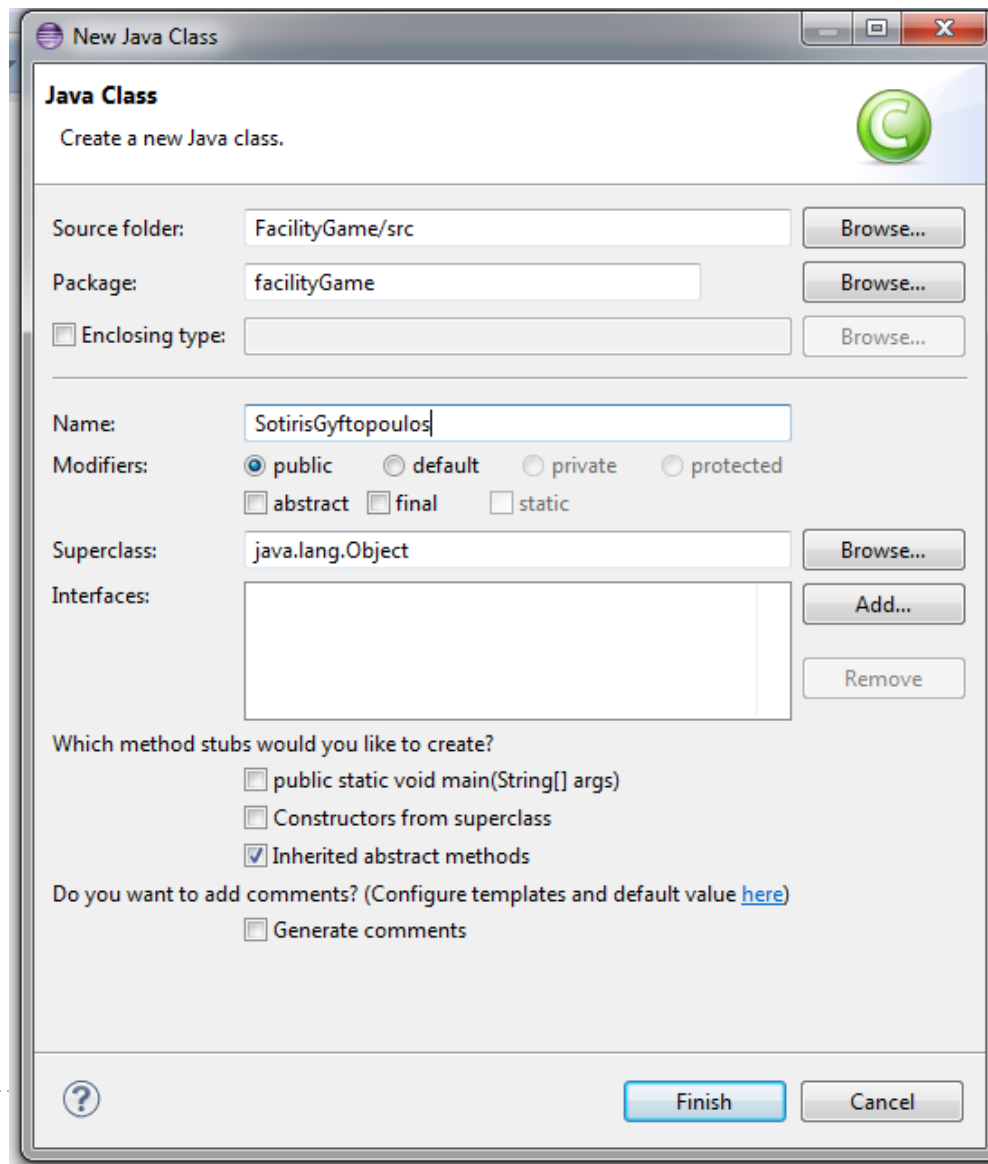


Δημιουργία νέου παίκτη

- ▶ Δημιουργούμε μια νέα κλάση στο πακέτο και δίνουμε ένα όνομα στον παίκτη – κλάση π.χ. SotirisGyftopoulos



Δημιουργία νέου παίκτη



New Java Class

Create a new Java class.

Source folder: FacilityGame/src Browse...

Package: facilityGame Browse...

☐ Enclosing type: Browse...

Name: SotirisGyftopoulos

Modifiers: ☒ public ☐ default ☐ private ☐ protected
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: Add...
Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)
☐ Constructors from superclass
☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))
☐ Generate comments

? Finish Cancel

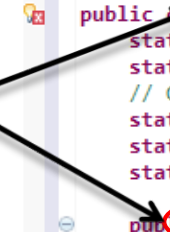
Δημιουργία νέου παίκτη

- ▶ Στη νέα κλάση που δημιουργήσαμε αντιγράφουμε τα περιεχόμενα από κάποιον υπάρχοντα παίκτη π.χ. `FPlayerMy1.java`.
- ▶ ΠΡΟΣΟΧΗ! Κατά την αντιγραφή των περιεχομένων αντιγράφονται και τα ονόματα των δηλώσεων της κλάσης και της συνάρτησης δημιουργίας! Πρέπει να τα διορθώσουμε και να δώσουμε το σωστό όνομα όπως το δηλώσαμε στην δημιουργία της κλάσης!



Δημιουργία νέου παίκτη

Πρέπει να διορθώσουμε τα ονόματα που αντιγράφηκαν και να βάλουμε αυτά που ορίσαμε στην κλάση π.χ. SotirisGyftopoulos



```
package facilityGame;

import java.util.Vector;

public class MyFPlayer1 extends FPlayer {
    static String playerName = "MyFPplayer1";
    static String version = "1.0";
    // Give your personal information in Greek
    static int afm = 11111; // AFM should be in form 5XXXX
    static String firstname = "John";
    static String lastname = "Doe";

    public MyFPlayer1(EnumPlayer player) {
        super(player, playerName, version, afm, firstname, lastname);
    }

    public int nextMove(FacilityGameAPI game) {
        int move = -1;

        // Select which free location (move = location) to occupy

        // Trivial choice: Select the first free location
        int n = game.getN();
        for (int i = 0; i < n; i++) {
            if (game.getStatus(i) == EnumFacilityStatus.FREE) {
                // Select location i
                move = i;
                // The values of the locations are stored in the array
                // game.value
                // int locationValue = game.value[i];

                break; // Abandon the loop
            }
        }

        // Obtain some information ...
    }
}
```

Δημιουργία νέου παίκτη

- ▶ Στη συνέχεια πρέπει να δώσουμε τα στοιχεία μας στις μεταβλητές που ορίζονται μέσα στην κλάση. Τα στοιχεία αυτά είναι:
 1. Το όνομα που θα χρησιμοποιούμε στο παιχνίδι για τον παίκτη μας (playerName)
 2. Η έκδοση της στρατηγικής μας, σε περίπτωση που στο μέλλον θέλουμε να κάνουμε βελτιώσεις και να πετύχουμε καλύτερα αποτελέσματα (version)
 3. Τον αριθμό μητρώου του φοιτητή (afm)
 4. Το όνομα του φοιτητή (firstname)
 5. Το επίθετο του φοιτητή (lastname)
- ▶ Η αλλαγή των στοιχείων είναι υποχρεωτική για τη σωστή εκτέλεση του παιχνιδιού!



Δημιουργία νέου παίκτη

```
static String playerName = "Sotiris_Gyftopoulos_Player";  
static String version = "1.0";  
// Give your personal information in Greek  
static int afm = 55555; // AFM should be in form 5XXXX  
static String firstname = "Sotiris";  
static String lastname = "Gyftopoulos";
```



Δήλωση του παίκτη μας στο παιχνίδι

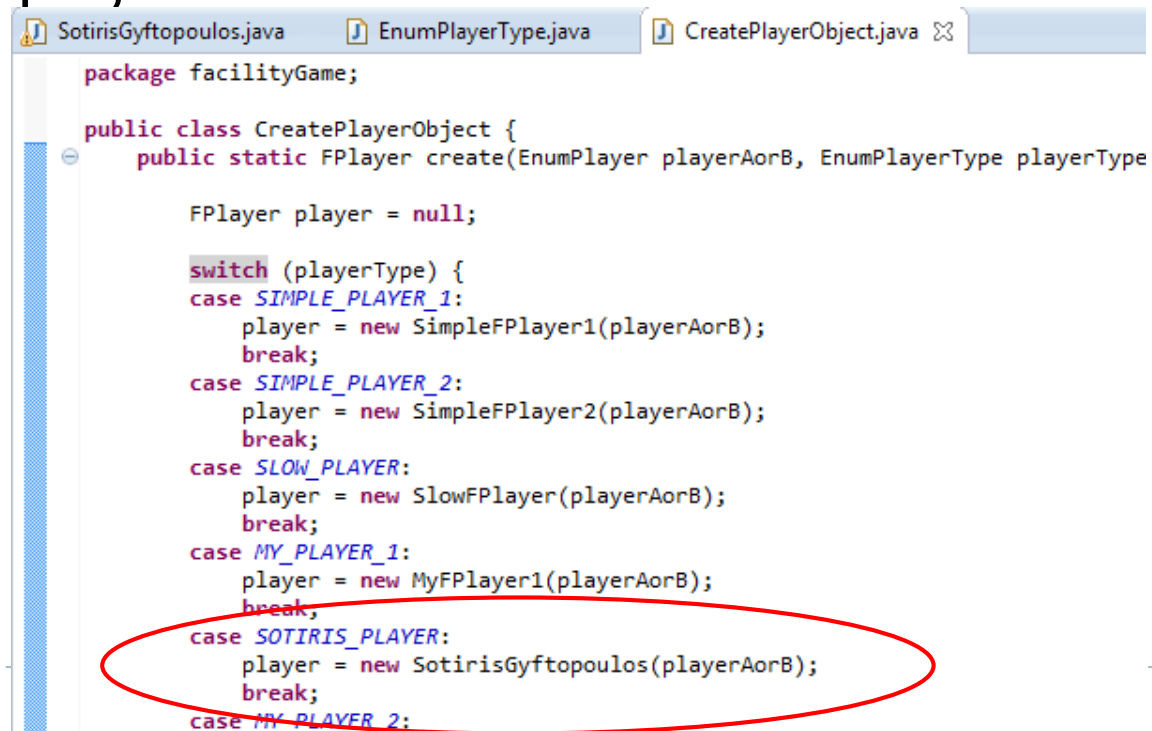
- ▶ Στη συνέχεια πρέπει να κάνουμε τις απαραίτητες αλλαγές σε κάποια αρχεία για να δηλώσουμε τον παίκτη μας.
- ▶ Ανοίγουμε το αρχείο EnumPlayerType.java και δημιουργούμε μια νέα καταχώρηση για τον παίκτη μας δηλώνοντας ένα νέο όνομα για αυτόν.

```
package facilityGame;

public enum EnumPlayerType {
    FPLAYER_SIMPLE_1, // Reserved,
    FPLAYER_201112, // Reserved, d
    FPLAYER_201213A, // Reserved, c
    FPLAYER_201213B, // Reserved, c
    SIMPLE_PLAYER_1,
    SIMPLE_PLAYER_2,
    SLOW_PLAYER,
    MY_PLAYER_1,
    MY_PLAYER_2,
    SOTIRIS_PLAYER,
    // συμπληρώστε, διορθώστε κτλ.
}
```

Δήλωση του παίκτη μας στο παιχνίδι

- ▶ Ανοίγουμε το αρχείο `CreatePlayerObject.java` και στο `switch` που περιέχεται μέσα δημιουργούμε μια νέα καταχώρηση που θα συνδέει το όνομα που δηλώσαμε στο αρχείο `EnumPlayerType.java` με το όνομα της κλάσης του παίκτη μας



```
package facilityGame;

public class CreatePlayerObject {
    public static FPlayer create(EnumPlayer playerAorB, EnumPlayerType playerType) {
        FPlayer player = null;

        switch (playerType) {
            case SIMPLE_PLAYER_1:
                player = new SimpleFPlayer1(playerAorB);
                break;
            case SIMPLE_PLAYER_2:
                player = new SimpleFPlayer2(playerAorB);
                break;
            case SLOW_PLAYER:
                player = new SlowFPlayer(playerAorB);
                break;
            case MY_PLAYER_1:
                player = new MyFPlayer1(playerAorB);
                break;
            case SOTIRIS_PLAYER:
                player = new SotirisGyftopoulos(playerAorB);
                break;
            case MY_PLAYER_2:
                // ...
            break;
        }
    }
}
```

Δημιουργία στρατηγικής

- ▶ Αφού τελειώσαμε τις δηλώσεις για τον παίκτη μας στο παιχνίδι, πρέπει να δημιουργήσουμε τη στρατηγική μας. Η υλοποίηση της στρατηγικής μας γίνεται στη συνάρτηση `nextMove` στην κλάση του παίκτη που δημιουργήσαμε. Η συνάρτηση αυτή επιστρέφει σε κάθε γύρω τον αριθμό του κόμβου που θέλουμε να δεσμεύσουμε αφού κάνουμε τους ελέγχους που θέλουμε.
- ▶ ΠΡΟΣΟΧΗ! Οι αριθμοί των κόμβων ξεκινούν από το 0 και όταν θα κάνουμε μια επιλογή κόμβου θα πρέπει να είμαστε σίγουροι ότι ο κόμβος αυτός πρέπει να είναι ελεύθερος



Δημιουργία στρατηγικής

- ▶ Για την υλοποίηση της στρατηγικής μας έχουμε στη διάθεσή μας μια σειρά από συναρτήσεις

```
// Return the number of nodes in the game
public int getN();

// Return the seed for the pseudorandom number generator
public long getSeed();

// Get the value of a specific node
public int getValue(int i);

// Get an array with the values of all nodes
public int[] getAllValues();

// Get the status of a specific node
public EnumFacilityStatus getStatus(int node);

// Get an array with the status of all nodes
public EnumFacilityStatus[] getAllStatus();

// The number of moves so far
public int getCurMoveIndex();

// Which player made each move?
public Vector<EnumPlayer> getMoveByPlayer();

// Which location/node was chosen at each move?
public Vector<Integer> getMoveLocation();

// Get current score
public GameScore getScore();

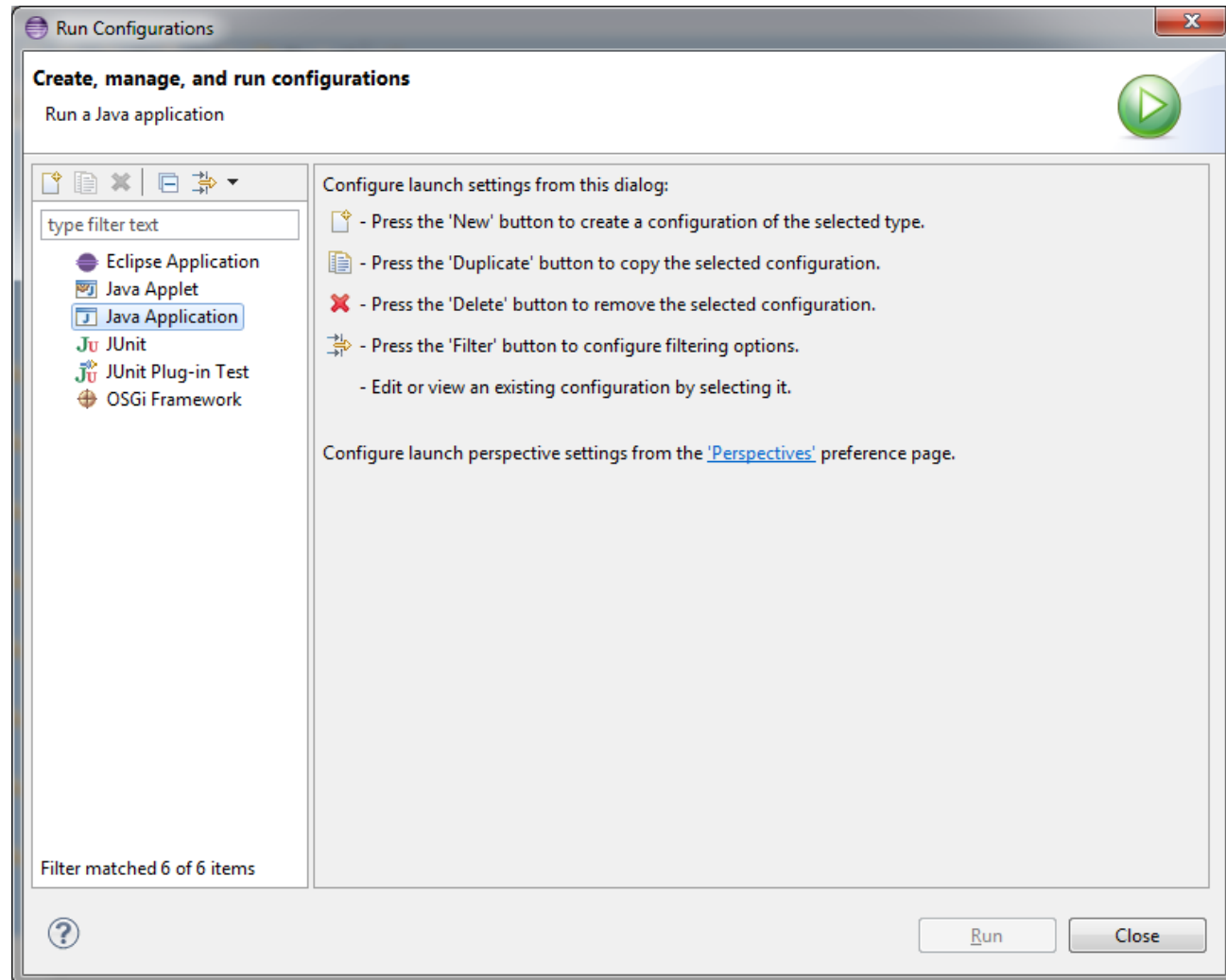
// Are there any moves left in the game?
public boolean isFinished();
```

Εκτέλεση του παιχνιδιού τοπικά

- ▶ Για να κάνουμε δοκιμές στη στρατηγική μας μπορούμε να εκτελέσουμε το παιχνίδι στον υπολογιστή μας.
- ▶ Το παιχνίδι έχει ήδη μέσα υλοποιημένες τις στρατηγικές δύο απλών παικτών: του `SIMPLE_PLAYER_1` και του `SIMPLE_PLAYER_2` που παίζουν σε `NORMAL mode`.
- ▶ Θα εκτελέσουμε το αρχείο `Fserver.java` που θα υλοποιεί τη στρατηγική του αντιπάλου μας δίνοντας τις απαραίτητες παραμέτρους.
- ▶ Για να περάσουμε τις παραμέτρους που θέλουμε κάνουμε δεξί κλικ πάνω στο `Fserver.java` → `Run As` → `Run Configurations...`



Εκτέλεση του παιχνιδιού τοπικά

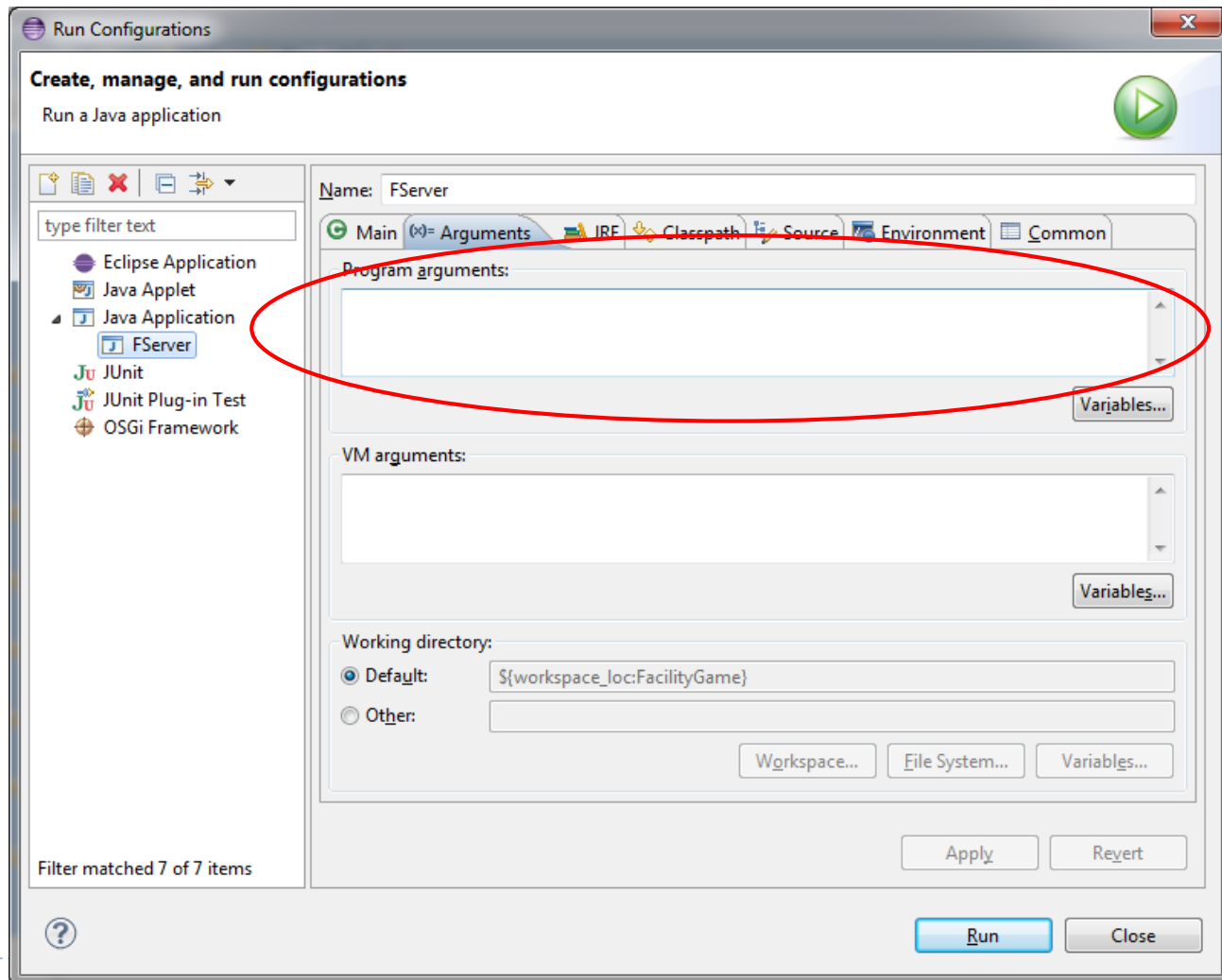


Εκτέλεση του παιχνιδιού τοπικά

- ▶ Κάνουμε διπλό κλικ στην επιλογή Java Application στο παράθυρο που εμφανίζεται για να δημιουργήσουμε νέα καταχώρηση παραμέτρων.
- ▶ Στην καρτέλα Arguments της νέας καταχώρησης παραμέτρων πληκτρολογούμε στο κατάλληλο πλαίσιο τις παραμέτρους που θέλουμε.

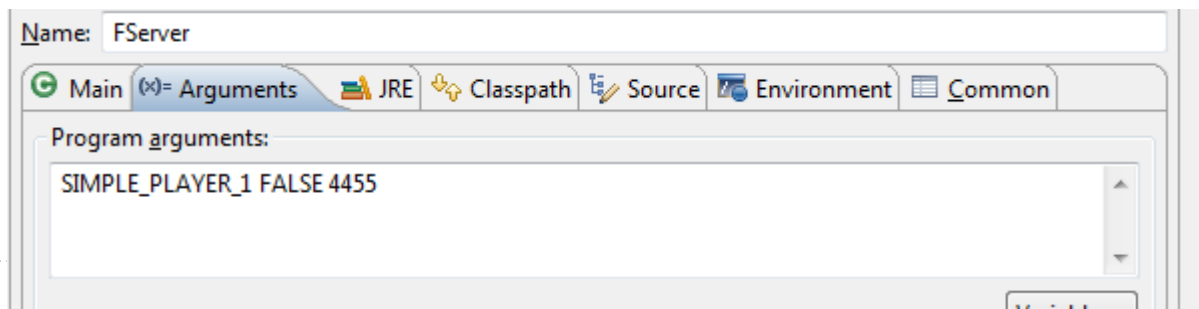


Εκτέλεση του παιχνιδιού τοπικά



Παράμετροι για τον Server

- ▶ Οι παράμετροι που πληκτρολογούμε είναι
 - ▶ `<PlayerType> <verbose> <port>`
- ▶ Στο `PlayerType` δίνουμε όνομα του παίκτη που θέλουμε να είναι αντίπαλος του παίκτη μας π.χ.
`SIMPLE_PLAYER_1`
- ▶ Στο `verbose` καθορίζουμε αν το πρόγραμμα θα εμφανίζει αναλυτικά μηνύματα ή συνοπτικά (`TRUE` ή `FALSE`)
- ▶ Στο `port` καθορίζουμε σε ποια θύρα θα εκτελείται ο server (συνιστάται η θύρα 4455)



Εκτέλεση του παιχνιδιού τοπικά

- ▶ Πατάμε Apply στο παράθυρο και στη συνέχεια Run.
- ▶ Στον χώρο μηνυμάτων στο Eclipse εμφανίζονται τα μηνύματα που εκτυπώνει το Fserver.



The screenshot shows the Eclipse IDE's Console window. The title bar indicates the application is 'FServer [Java Application]' running from 'C:\Program Files\Java\jre7\bin\javaw.exe' on 3 Nov 2013 at 10:26:26 μ.μ. The console output shows the following messages:

```
PlayerA: Usagejava -cp <jarfile.jar> facilityGame.FServer <player_type> <verbose> <port>
PlayerA: FacilityServer v0.50
PlayerA: FacilityServer (PlayerA) is starting ...
PlayerA: The server is listening at Sotos-Lap/192.168.2.3 on port: 4455 and waiting for the client (PlayerB) to connect ...
```

An arrow points from the text below to the IP address '192.168.2.3' in the console output.

- ▶ Παρατηρούμε τη διεύθυνση IP του υπολογιστή μας για να τη χρησιμοποιήσουμε στις παραμέτρους του παίκτη μας.

Εκτέλεση του παιχνιδιού τοπικά

- ▶ Μετά την εκτέλεση του `FServer.java` εκτελούμε τον παίκτη μας για να παίξει με τον αντίπαλο. Ο παίκτης μας θα εκτελεστεί με τη βοήθεια του `FClient.java` αφού περάσουμε τις κατάλληλες παραμέτρους.
- ▶ Δημιουργούμε μια καταχώρηση παραμέτρων για το αρχείο `FClient.java` με τον ίδιο τρόπο που δημιουργήσαμε για το `FServer.java` (δεξί κλικ πάνω στο `FClient.java` → `Run As` → `Run Configurations...` → διπλό κλικ στο `Java Application`)
- ▶ Στην καρτέλα `Arguments` πληκτρολογούμε τις παραμέτρους για τον παίκτη μας.

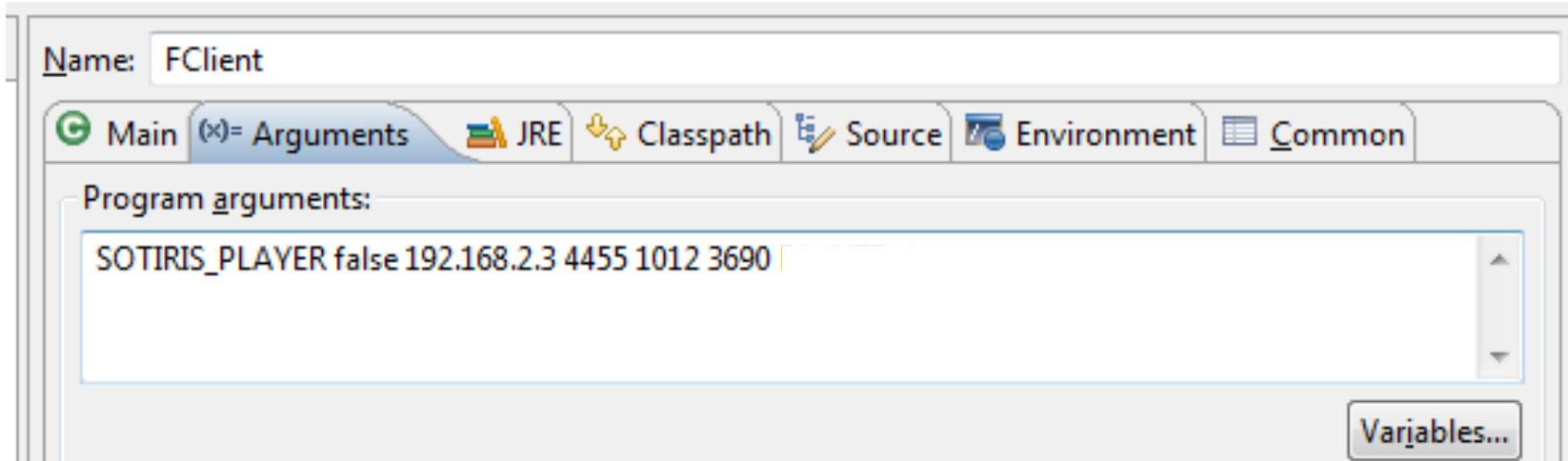


Παράμετροι για τον παίκτη μας

- ▶ Οι παράμετροι για τον παίκτη μας είναι
<PlayerType> <verbose> <host> <port> <nodes> <seed> <game type> <Player_A>
- ▶ Το PlayerType είναι το όνομα που δώσαμε στον παίκτη μας στο EnumPlayerType (π.χ. SOTIRIS_PLAYER)
- ▶ Το verbose είναι σχετικό με αναλυτικά ή συνοπτικά μηνύματα (TRUE ή FALSE)
- ▶ Το host είναι η IP του Server όπως την παρατηρήσαμε κατά την εκτέλεση του Server (π.χ. 192.168.2.3)
- ▶ Το port είναι η θύρα του Server (4455)
- ▶ Το nodes είναι το πλήθος των κόμβων (π.χ. 1500)
- ▶ Το seed χρησιμοποιείται για την ανάθεση αξιών στους κόμβους (π.χ. 234)
- ▶ Το game type καθορίζει αν παίζετε παιχνίδι NORMAL ή NOQUADRUPLES
- ▶ Στο Player_A μπορείτε να δηλώσουμε προαιρετικά αν θέλουμε να παίζουμε πρώτοι (PLAYER_A) ή δεύτεροι (PLAYER_B)



Παράμετροι για τον παίκτη μας



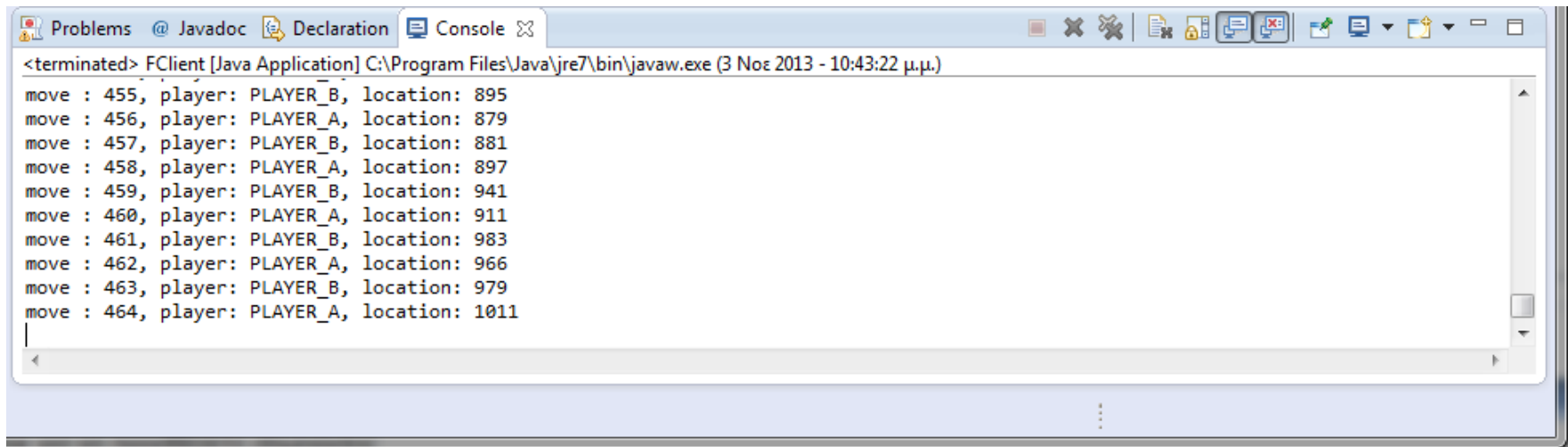
- ▶ Παρατήρηση: παραλείποντας το game type θεωρείται ότι παίζουμε σε NORMAL mode
- ▶ Πατάμε Apply στο παράθυρο και στη συνέχεια Run.
- ▶ Με την εκτέλεση του FClient γίνεται η σύνδεση με τον Server και οι παίκτες εκτελούν τις κινήσεις τους. Στο χώρο των μηνυμάτων εμφανίζονται τα αποτελέσματα του παιχνιδιού

Εκτέλεση του παιχνιδιού στον server του μαθήματος

- ▶ Για την εκτέλεση του παιχνιδιού στον server του μαθήματος, εκτελείται μόνο το `Fclient.java` με τις κατάλληλες παραμέτρους ώστε να επικοινωνήσει με τον server.
- ▶ Ο server που χρησιμοποιείται είναι ο
`euclid.ee.duth.gr`
- ▶ Ανάλογα με το port που θα ορίσετε κατά την εκτέλεση του `Fclient.java` θα αναμετρηθεί ο παίκτης σας με τον αντίστοιχο αντίπαλο.

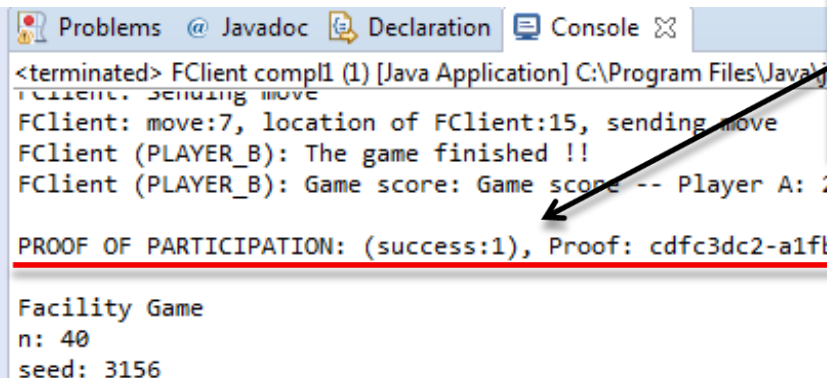


Αποτελέσματα εκτέλεσης



```
<terminated> FClient [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (3 Νοε 2013 - 10:43:22 μ.μ.)
move : 455, player: PLAYER_B, location: 895
move : 456, player: PLAYER_A, location: 879
move : 457, player: PLAYER_B, location: 881
move : 458, player: PLAYER_A, location: 897
move : 459, player: PLAYER_B, location: 941
move : 460, player: PLAYER_A, location: 911
move : 461, player: PLAYER_B, location: 983
move : 462, player: PLAYER_A, location: 966
move : 463, player: PLAYER_B, location: 979
move : 464, player: PLAYER_A, location: 1011
```

- ▶ Ανατρέχοντας παραπάνω μπορούμε να δούμε το αποτέλεσμα



```
<terminated> FClient comp11 (1) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe
FClient: sending move
FClient: move:7, location of FClient:15, sending move
FClient (PLAYER_B): The game finished !!
FClient (PLAYER_B): Game score: Game score -- Player A: 20316, Player B: 119670

PROOF OF PARTICIPATION: (success:1), Proof: cdfc3dc2-a1fb-470e-9983-7987a0b331fe

Facility Game
n: 40
seed: 3156
```

Οι επιτυχημένες παρτίδες έχουν success:1
Το αποδεικτικό χρειάζεται για την υποβολή

Εκτέλεση του παιχνιδιού στον server του μαθήματος

▶ Παραδείγματα παραμέτρων

- ▶ MY_PLAYER false euclid.ee.duth.gr 4480 40 3156
- ▶ MY_PLAYER true euclid.ee.duth.gr 4480 40 3158
- ▶ MY_NQ false euclid.ee.duth.gr 4483 1500 234 NOQUADRUPLES



Πηγές για υποστήριξη - ερωτήσεις:

- ▶ Στο eclass

- ▶ <https://eclass.duth.gr/courses/TMA392/>

- ▶ Στο Forum deece

- ▶ <https://www.deece.edu.gr/forum/viewtopic.php?f=198&t=9022>

