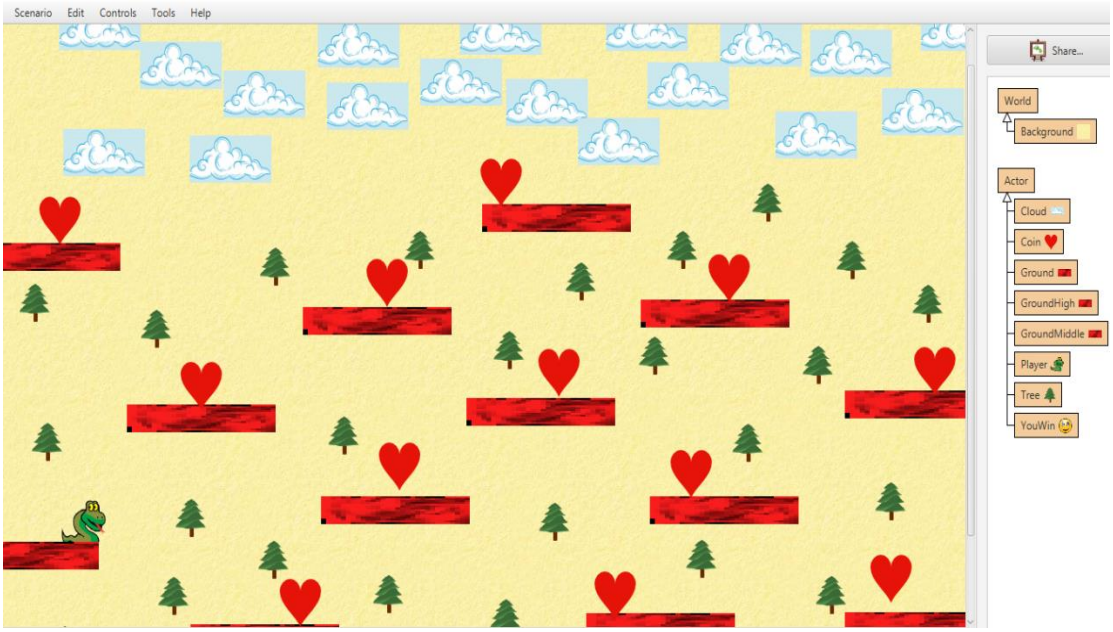
Άσκηση 8η

Δημιουργήστε τις κλάσεις και τις υποκλάσεις όπως απεικονίζονται παρακάτω, γράψτε τους κώδικες στους αντίστοιχους editor και «τρέξτε» το πρόγραμμα….



Το παρακάτω πρόγραμμα Java δημιουργεί ένα "παρασκήνιο" σε ένα παιχνίδι χρησιμοποιώντας το πλαίσιο εργασίας Greenfoot.

```
import greenfoot.*;

/**
 * Write a description of class Background here.
 *
 * Kaffetzis Vasileios (your name)
 * @version (a version number or a date)
 */
public class Background extends World
{
    public Background()
    {
        // Δημιουργία ενός νέου κόσμου με κύτταρα 2000x800 και μέγεθος κυττάρου 1x1 pixels.
        super(2000, 800, 1);
        prepare();
    }
    public void act()
    {
        // Μέθοδος που καλείται κάθε φορά που εκτελείται ένα frame του παιχνιδιού
        if (Greenfoot.getRandomNumber(100) <= 5) // Τυχαία πρόσθεση αντικειμένου Coin με πιθανότητα 5% σε κάθε frame.
        {
            addObject(new Coin(),getWidth()-1, Greenfoot.getRandomNumber(400) + 300);
        }
    }
    private void prepare() // Ιδιωτική μέθοδος για την αρχικοποίηση του περιβάλλοντος
    {
        // ... (Εδώ περιλαμβάνονται οι εντολές για τη δημιουργία και τοποθέτηση των αντικειμένων)
        Player player = new Player();
        addObject(player,30,576);
        player.setLocation(30,579);
        player.setLocation(100,579);
        Ground ground = new Ground();
        addObject(ground,28,613);
        Ground ground2 = new Ground();
        addObject(ground2,242,465);
        Ground ground3 = new Ground();
        addObject(ground3,376,565);
        ground3.setLocation(452,564);
        Coin coin21 = new Coin();
        addObject(coin21,1823,308);
    }
}
```

Στη υποκλάση του Background επειδή θέλουμε να βάλουμε πολλά αντικείμενα, βάλαμε τα αντικείμενα χαροκόψητα και μετά κάναμε Save the World.

Το πρόγραμμα αναπαριστά τον παίκτη (κλάση **Player**) σε ένα παιχνίδι χρησιμοποιώντας το πλαίσιο εργασίας Greenfoot.

```
import greenfoot.*;

/**
 * Write a description of class Player here.
 *
 * Kaffetzis Vasileos
 * 15/11/2024
 */
public class Player extends Actor
{
    private int vSpeed = 0; // Καθορίζει την κατακόρυφη ταχύτητα του παίκτη
    private int acceleration = 1; // Επιτάχυνση καθόλη τη διάρκεια της πτώσης
    private int jumpHeight = -20; // Ύψος του άλματος
    private int collect = 0; // Μετρητής για τον αριθμό των συλλεγμένων αντικειμένων
    // Μέθοδος που καλείται κάθε frame του παιχνιδιού
    public void act()
    {
        moveAround(); // Κίνηση του παίκτη
        checkFalling(); // Έλεγχος αν ο παίκτης πέφτει
        collect(); // Έλεγχος σύλλεξης αντικειμένων
    }
    // Κίνηση του παίκτη ανάλογα με τα πλήκτρα που πατώνται
    private void moveAround()
    {
        if (Greenfoot.isKeyDown("right"))
        {
            move(2);
        }
        if (Greenfoot.isKeyDown("left"))
        {
            move(-1);
        }
        if (Greenfoot.isKeyDown("space") && (onGround() || onGround2() || onGround3()))
        {
            // Άλμα μόνο όταν ο παίκτης βρίσκεται στο έδαφος
            vSpeed = jumpHeight;
            fall();
        }
    }
    // Έλεγχος αν ο παίκτης βρίσκεται στο έδαφος
    boolean onGround()
    {
        Actor under = getOneObjectAtOffset(0, getImage().getHeight() / 2, Ground.class);
        return under != null;
    }
    boolean onGround2()
    {
        Actor under2 = getOneObjectAtOffset(0, getImage().getHeight() / 2, GroundMiddle.class);
        return under2 != null;
    }
    boolean onGround3()
    {
        Actor under3 = getOneObjectAtOffset(0, getImage().getHeight() / 2, GroundMiddle.class);
        return under3 != null;
    }
    // Έλεγχος αν ο παίκτης πέφτει και ρύθμιση της κατακόρυφης ταχύτητας
    private void fall()
    {
        setLocation(getX(), getY() + vSpeed);
        vSpeed = vSpeed + acceleration;
    }
    // Έλεγχος της κατάστασης πτώσης του παίκτη και ρύθμιση της ταχύτητας
    public void checkFalling()
    {
        if (!onGround() && !onGround2() && !onGround3())
        {
            fall();
        }
        if (onGround() || onGround2() || onGround3())
        {
            vSpeed = 0;
        }
        if (getY() >= 750)
        {
            // Αν ο παίκτης πέσει κάτω από το όριο του κόσμου, πραγματοποιείται ένα άλμα
            vSpeed = jumpHeight;
        }
    }
```

Θεωρητικά.. με τον ίδιο τρόπο.. Έχουμε ήδη κατασκευάσει 3 διαφορετικές πίστες, με διαφορετικό επίπεδο δυσκολίας..

```
// Έλεγχος σύλλεξης αντικειμένων (συγκεκριμένα, νομισμάτων)
public void collect()
{
    Actor coin = getOneIntersectingObject(Coin.class);
    if (coin != null)
    {
        getWorld().removeObject(coin);
        collect++;
    }
    if (collect == 50)
    {
        // Αν ο παίκτης συλλέξει 50 νομίσματα, εμφανίζεται ένα μήνυμα νίκης (YouWin)
        Greenfoot.stop();
        YouWin youWin = new YouWin(); // Δημιουργία αντικειμένου της κλάσης YouWin
        getWorld().addObject(youWin, getWorld().getWidth() / 2, getWorld().getHeight() / 2); // Προσθήκη του YouWin στο κέντρο του κόσ
        Greenfoot.stop(); // Σταματά το παιχνίδι
        getWorld().removeObject(this);
    }
}
}
```

Το παρακάτω πρόγραμμα Java αναπαριστά ένα αντικείμενο "Cloud" σε ένα παιχνίδι που χρησιμοποιεί το πλαίσιο εργασίας Greenfoot.

Το πρόγραμμα δημιουργεί ένα αντικείμενο Cloud που μπορεί να κινείται προς τα αριστερά και προς τα δεξιά όταν πατιούνται τα αντίστοιχα πλήκτρα. Επιπλέον, εάν το αντικείμενο φτάσει στην αρχή του κόσμου, τοποθετείται στο τέλος με μια τυχαία Y συντεταγμένη.

```
import greenfoot.*;
public class Cloud extends Actor // Κλάση που αναπαριστά το αντικείμενο Cloud στο παιχνίδι
{
    public Cloud() // Κατασκευαστής
    {
    }
      public void act() // Μέθοδος που καλείται κάθε frame του παιχνιδιού
    {
        moveCloud(); // Κίνηση του αντικειμένου Cloud
    }
    private void moveCloud() // Μέθοδος για την κίνηση του αντικειμένου Cloud
    {
        if (Greenfoot.isKeyDown("right"))
        {
            move(-2); // Κίνηση προς τα αριστερά όταν πατιέται το κουμπί "right"
        }
        if (Greenfoot.isKeyDown("left"))
        {
            move(2); // Κίνηση προς τα δεξιά όταν πατιέται το κουμπί "left"
        }
        if (getX() == 0)
        {
            // Εάν το αντικείμενο φτάσει στην αρχή του κόσμου, τοποθετείται στο τέλος με τυχαία υ-συντεταγμένη
            setLocation(getWorld().getWidth() - 1, Greenfoot.getRandomNumber(170) + 30);
        }
    }
}
```

Το παρακάτω πρόγραμμα Java αναπαριστά ένα αντικείμενο "Ground" σε ένα παιχνίδι που χρησιμοποιεί το πλαίσιο εργασίας Greenfoot.

Το πρόγραμμα δημιουργεί ένα αντικείμενο Ground που μπορεί να κινείται προς τα αριστερά και προς τα δεξιά όταν πατιούνται τα αντίστοιχα πλήκτρα. Επιπλέον, εάν το αντικείμενο φτάσει στην αρχή του κόσμου, τοποθετείται στο τέλος με μια τυχαία Y συντεταγμένη.

```java
import greenfoot.*;
// Κλάση που αναπαριστά το αντικείμενο Ground στο παιχνίδι
public class Ground extends Actor
{
    // Κατασκευαστής
    public Ground()
    {
        getImage().scale(getImage().getWidth() * 6, getImage().getHeight() * 2);
        // Κλιμάκωση της εικόνας του αντικειμένου για να φαίνεται σαν ένα μεγάλο έδαφος
    }
    // Μέθοδος που καλείται κάθε frame του παιχνιδιού
    public void act()
    {
        moveGround(); // Κίνηση του αντικειμένου Ground
    }
    // Μέθοδος για την κίνηση του αντικειμένου Ground
    private void moveGround()
    {
        if (Greenfoot.isKeyDown("right"))
        {
            move(-10); // Κίνηση προς τα αριστερά όταν πατιέται το κουμπί "right"
        }
        if (Greenfoot.isKeyDown("left"))
        {
            move(2); // Κίνηση προς τα δεξιά όταν πατιέται το κουμπί "left"
        }
        if (getX() == 0)
        {
            // Εάν το αντικείμενο φτάσει στην αρχή του κόσμου, τοποθετείται στο τέλος με τυχαία υ-συντεταγμένη
            setLocation(getWorld().getWidth() - 1, Greenfoot.getRandomNumber(410) + 300);
        }
    }
}
```

Τα παρακάτω προγράμματα Java αναπαριστούν τα αντικείμενα "GroundHigh" και "GroundMiddle" σε ένα παιχνίδι που χρησιμοποιεί το πλαίσιο εργασίας Greenfoot. Κατασκευάζονται με τρόπο παρόμοιο με το πρόγραμμα του αντικειμένου "Ground".

```java
import greenfoot.*;
public class GroundHigh extends Actor
{
    public GroundHigh()
    {
        getImage().scale(getImage().getWidth()*6,getImage().getHeight()*2);
    }
    public void act()
    {
        if (Greenfoot.isKeyDown("right"))
        {
            move (-5);
        }
        if (Greenfoot.isKeyDown("left"))
        {
            move (5);
        }
        if (getX() == 0)
        {
            setLocation(getWorld().getWidth()-1, Greenfoot.getRandomNumber(100)+480);
        }
    }
}
```

```java
import greenfoot.*;
public class GroundMiddle extends Actor
{
    public GroundMiddle()
    {
        getImage().scale(getImage().getWidth()*6,getImage().getHeight()*2);
    }
    public void act()
    {
        if (Greenfoot.isKeyDown("right"))
        {
            move (-5);
        }
        if (Greenfoot.isKeyDown("left"))
        {
            move (5);
        }
        if (getX() == 0)
        {
            setLocation(getWorld().getWidth()-1, Greenfoot.getRandomNumber(100)+480);
        }
    }
}
```

Ο κώδικας αναπαριστά μια κλάση Tree σε ένα παιχνίδι που χρησιμοποιεί το πλαίσιο εργασίας Greenfoot.

Ο κώδικας δημιουργεί ένα αντικείμενο Tree που μπορεί να κινείται προς τα αριστερά και προς τα δεξιά όταν πατιούνται τα αντίστοιχα πλήκτρα. Επιπλέον, εάν το αντικείμενο φτάσει στην αρχή του κόσμου, τοποθετείται στο τέλος με μια τυχαία Y συντεταγμένη.

```
import greenfoot.*;
// Κλάση που αναπαριστά το αντικείμενο Tree στο παιχνίδι
public class Tree extends Actor
{
    // Μέθοδος που καλείται κάθε frame του παιχνιδιού
    public void act()
    {
        moveTree(); // Κίνηση του αντικειμένου Tree
    }
    // Μέθοδος για την κίνηση του αντικειμένου Tree
    private void moveTree()
    {
        if (Greenfoot.isKeyDown("right"))
        {
            move(-8); // Κίνηση προς τα αριστερά όταν πατιέται το κουμπί "right"
        }
        if (Greenfoot.isKeyDown("left"))
        {
            move(2); // Κίνηση προς τα δεξιά όταν πατιέται το κουμπί "left"
        }
        if (getX() == 0)
        {
            // Εάν το αντικείμενο φτάσει στην αρχή του κόσμου, τοποθετείται στο τέλος με τυχαία Y συντεταγμένη
            setLocation(getWorld().getWidth() - 1, Greenfoot.getRandomNumber(410) + 300);
        }
    }
}
```

Ο κώδικας παραπάνω αναπαριστά μια κλάση Coin σε ένα παιχνίδι που χρησιμοποιεί το πλαίσιο εργασίας Greenfoot.

Ο κώδικας δημιουργεί ένα αντικείμενο Coin που κινείται προς τα αριστερά και προς τα δεξιά όταν πατιούνται τα αντίστοιχα πλήκτρα. Επιπλέον, εάν το αντικείμενο φτάσει στην αρχή του κόσμου, αφαιρείται από τον κόσμο (removeObject(this)). Είναι σημαντικό να σημειωθεί ότι αυτή η λειτουργία μπορεί να καταργηθεί ή να τροποποιηθεί ανάλογα με τον σκοπό του παιχνιδιού σας.

```
import greenfoot.*;
public class Coin extends Actor // Κλάση που αναπαριστά το αντικείμενο Coin στο παιχνίδι
{
    // Μέθοδος που καλείται κάθε frame του παιχνιδιού
    public void act()
    {
        moveCoin(); // Κίνηση του αντικειμένου Coin
    }
    // Μέθοδος για την κίνηση του αντικειμένου Coin
    private void moveCoin()
    {
        if (Greenfoot.isKeyDown("right"))
        {
            move(-10); // Κίνηση προς τα αριστερά όταν πατιέται το κουμπί "right"
        }
        if (Greenfoot.isKeyDown("left"))
        {
            move(2); // Κίνηση προς τα δεξιά όταν πατιέται το κουμπί "left"
        }
        if (getX() == 0)
        {
            getWorld().removeObject(this); // Κατάργηση του αντικειμένου Coin από τον κόσμο όταν φτάσει στην αρχή του κόσμου
        }
    }
}
```

Ο κώδικας που παρέχεται παρακάτω αναπαριστά μια κλάση YouWin σε ένα παιχνίδι με το πλαίσιο εργασίας Greenfoot. Η κλάση δεν περιέχει καμία ενέργεια (act() μέθοδος) στην παρούσα μορφή. Αυτό σημαίνει ότι δεν κάνει τίποτα κατά τη διάρκεια κάθε κύκλου ενημέρωσης του παιχνιδιού.

Η κλάση YouWin χρησιμοποιείται για να εμφανίζει ένα μήνυμα ή εικόνα που δείχνει ότι ο παίκτης κέρδισε το παιχνίδι, αλλά όπως αναφέρεται, η act() μέθοδος δεν περιέχει κώδικα.

Εάν θέλετε να προσθέσετε λειτουργίες στην κλάση YouWin, μπορείτε να προσθέσετε κώδικα στη μέθοδο act(). Για παράδειγμα, μπορείτε να προσθέσετε κώδικα για το κλείσιμο του παιχνιδιού όταν ο παίκτης κερδίζει ή οποιαδήποτε άλλη λειτουργία που επιθυμείτε.

```
import greenfoot.*;
public class YouWin extends Actor
{

    public void act()
    {
        // Add your action code here.
    }
}
```

Στη γλώσσα προγραμματισμού Java, οι λέξεις-κλειδιά private και public χρησιμοποιούνται για τον έλεγχο της πρόσβασης σε μέλη (μεταβλητές, μέθοδοι, κ.λπ.) ενός αντικειμένου ή μιας κλάσης. Παρακάτω παρουσιάζεται η διαφορά μεταξύ τους:

Private:

Όταν μια μεταβλητή ή μια μέθοδος δηλώνεται ως private, αυτό σημαίνει ότι είναι προσβάσιμη μόνο εντός της ίδιας κλάσης.

Δεν μπορεί να προσπελαστεί από άλλες κλάσεις ή αντικείμενα που δημιουργούνται από άλλες κλάσεις.

Παρέχει υψηλότερο επίπεδο απομόνωσης και ελέγχου των δεδομένων, καθώς περιορίζει την πρόσβαση σε εσωτερικά μέλη της κλάσης.

Παράδειγμα:

```
public class MyClass
 {
   private int private Variable;
   private void private Method()
{
     // Κώδικας μεθόδου
   }
}
```

Public:

Όταν μια μεταβλητή ή μια μέθοδος δηλώνεται ως public, αυτό σημαίνει ότι είναι προσβάσιμη από οποιοδήποτε άλλο σημείο του προγράμματος.

Παρέχει ευρεία πρόσβαση στα μέλη της κλάσης και μπορεί να χρησιμοποιηθεί από άλλες κλάσεις ή αντικείμενα.

Είναι συχνά χρήσιμη για μέλη που πρέπει να είναι προσβάσιμα από άλλα τμήματα του προγράμματος.

Παράδειγμα:

```
public class MyClass
{
   public int public Variable;
   public void public Method()
 {
      // Κώδικας μεθόδου
   }
}
```

Η επιλογή μεταξύ private και public εξαρτάται από τη σχεδίαση του προγράμματος και το επίπεδο ελέγχου που επιθυμεί ο προγραμματιστής να διατηρήσει στον κώδικά του.

Αναφορές

- https://www.youtube.com/@CodingClub
- https://www.youtube.com/watch?v=LT6IPr2W8dw (ανακτήθηκε στις 14/11/2023)