

Entornos de desarrollo

Bloque 2

Tema 6: Clases y objetos en Java. Herencia.

Generación automática de componentes

Soluciones

Ejercicios propuestos

2.6.1.1. Crea un programa llamado *ListaVideoJuego* para almacenar objetos de una clase llamada *VideoJuego* que deberás definir. Para cada videojuego, almacenaremos su título, género y precio. Añade también los correspondientes *getters*, *setters* y constructores para establecer los valores. Define el *main*, la clase pública y la clase *VideoJuego* en el mismo fichero fuente. En el método *main* crea un *array* de 5 videojuegos, pide al usuario que rellene la información de cada videojuego y, muestra el título del videojuego más barato y más caro del *array*.

```
import java.util.Scanner;

/*
 * Main class
 */
public class VideoGameList
{
    public static final int MAX_VIDEOGAMES = 5;

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        VideoGame[] games = new VideoGame[MAX_VIDEOGAMES];
        int minPos = 0, maxPos = 0;

        /*
         * One way of getting video game data from the user:
         * Defining individual variables for each attribute and
         * then calling the constructor
         */
        for(int i = 0; i < MAX_VIDEOGAMES; i++)
        {
            String title, genre;
            float price;

            System.out.println("Enter title for videogame " + (i+1) + ":");
            title = sc.nextLine();

            System.out.println("Enter genre for videogame " + (i+1) + ":");
```

```
        genre = sc.nextLine();

        System.out.println("Enter price for videogame " + (i+1) + ":");
        price = sc.nextFloat();

        games[i] = new VideoGame(title, genre, price);
    }
    */

    /*
     * Another way: Calling an empty constructor and filling the
     * information of the object by calling the setters
     */
    for(int i = 0; i < MAX_VIDEOGAMES; i++)
    {
        games[i] = new VideoGame();

        System.out.println("Enter title for videogame " + (i+1) + ":");
        games[i].setTitle(sc.nextLine());

        System.out.println("Enter genre for videogame " + (i+1) + ":");
        games[i].setGenre(sc.nextLine());

        System.out.println("Enter price for videogame " + (i+1) + ":");
        games[i].setPrice(sc.nextFloat());
        sc.nextLine();
    }

    /*
     * Get cheapest and most expensive video game
     */
    for (int i = 1; i < MAX_VIDEOGAMES; i++)
    {
        if (games[i].getPrice() > games[maxPos].getPrice())
        {
            maxPos = i;
        }

        if (games[i].getPrice() < games[minPos].getPrice())
        {
            minPos = i;
        }
    }

    System.out.println("Cheapest videogame: "
        + games[minPos].getTitle());
    System.out.println("Most expensive videogame: "
        + games[maxPos].getTitle());
```

```
}  
}
```

```
/*  
 * VideoGame class  
 */  
class VideoGame  
{  
    String title;  
    String genre;  
    float price;  
  
    public VideoGame()  
    {  
        title = "";  
        genre = "";  
        price = 0;  
    }  
  
    public VideoGame(String t, String g, float p)  
    {  
        title = t;  
        genre = g;  
        price = p;  
    }  
  
    public String getTitle()  
    {  
        return title;  
    }  
  
    public void setTitle(String t)  
    {  
        title = t;  
    }  
  
    public String getGenre()  
    {  
        return genre;  
    }  
  
    public void setGenre(String g)  
    {  
        genre = g;  
    }  
  
    public float getPrice()  
    {  
        return price;  
    }  
}
```

```
    }  
  
    public void setPrice(float p)  
    {  
        price = p;  
    }  
}
```

2.6.2.1 Mejora el ejercicio anterior *ListaVideoJuego* en otro fichero fuente llamado *ListaVideoJuego2*. Ahora, cada videojuego deberá tener una **Compañía** (clase) que lo ha creado. Para cada compañía, necesitamos almacenar su nombre y año de fundación. Asocia una compañía a cada videojuego, de forma que algunos videojuegos puedan compartir el mismo objeto compañía. Luego, después de pedir al usuario que introduzca 5 videojuegos y mostrar el más barato y el más caro, muestra la compañía que aparece en la mayor parte de los videojuegos de la lista.

```
import java.util.Scanner;

/*
 * Main class
 */
public class VideoGameList2
{
    public static final int MAX_VIDEOGAMES = 5;

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        VideoGame[] games = new VideoGame[MAX_VIDEOGAMES];
        int minPos = 0, maxPos = 0;

        for(int i = 0; i < MAX_VIDEOGAMES; i++)
        {
            games[i] = new VideoGame();

            System.out.println("Enter title for videogame "
                               + (i+1) + ":");
            games[i].setTitle(sc.nextLine());

            System.out.println("Enter genre for videogame "
                               + (i+1) + ":");
            games[i].setGenre(sc.nextLine());

            System.out.println("Enter price for videogame "
                               + (i+1) + ":");
            games[i].setPrice(sc.nextFloat());
            sc.nextLine();

            // Add company to the video game

            System.out.println("Enter company name for videogame "
                               + (i+1) + ":");
            String name = sc.nextLine();

            System.out.println("Enter foundation year for videogame "
                               + (i+1) + ":");
```

```
        int foundationYear = sc.nextInt();
        sc.nextLine();

        Company comp = new Company(name, foundationYear);
        games[i].setCompany(comp);
    }

    // Cheapest and most expensive video game

    for (int i = 1; i < MAX_VIDEOGAMES; i++)
    {
        if (games[i].getPrice() > games[maxPos].getPrice())
        {
            maxPos = i;
        }

        if (games[i].getPrice() < games[minPos].getPrice())
        {
            minPos = i;
        }
    }

    System.out.println("Cheapest videogame: "
        + games[minPos].getTitle());
    System.out.println("Most expensive videogame: "
        + games[maxPos].getTitle());

    // Most repeated company

    int posCompany = 0;
    int maxCountCompany = 0;

    for (int i = 0; i < MAX_VIDEOGAMES; i++)
    {
        int countCompany = 1;
        for (int j = i+1; j < MAX_VIDEOGAMES; j++)
        {
            if (games[i].getCompany().getName().equals(
                games[j].getCompany().getName()))
            {
                countCompany++;
            }
        }

        if (countCompany > maxCountCompany)
        {
            maxCountCompany = countCompany;
            posCompany = i;
        }
    }
}
```

```
        System.out.println("Most repeated company = "
            + games[posCompany].getCompany().getName());
    }
}
```

```
/*
 * VideoGame class
 */
class VideoGame
{
    String title;
    String genre;
    float price;
    Company company;

    public VideoGame()
    {
        title = "";
        genre = "";
        price = 0;
    }

    public VideoGame(String t, String g, float p)
    {
        title = t;
        genre = g;
        price = p;
    }

    public String getTitle()
    {
        return title;
    }

    public void setTitle(String t)
    {
        title = t;
    }

    public String getGenre()
    {
        return genre;
    }

    public void setGenre(String g)
    {
        genre = g;
    }
}
```

```
public float getPrice()
{
    return price;
}

public void setPrice(float p)
{
    price = p;
}

public Company getCompany()
{
    return company;
}

public void setCompany(Company c)
{
    company = c;
}
}
```

```
/*
 * Company class
 */
class Company
{
    String name;
    int foundationYear;

    public Company(String n, int y)
    {
        name = n;
        foundationYear = y;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String n)
    {
        name = n;
    }

    public int getFoundationYear()
    {
        return foundationYear;
    }
}
```



```
    }  
  
    public void setFoundationYear(int y)  
    {  
        foundationYear = y;  
    }  
}
```

2.6.3.1. Mejora el ejercicio anterior *ListaVideoJuego2* en otro fichero fuente llamado *ListaVideoJuego3*. Añade una nueva clase llamada *PCVideoJuego* la cual hereda de la clase *VideoJuego*. Ésta tendrá dos atributos nuevos llamados *RAMminima* y *HDminimo* para almacenar la cantidad mínima de memoria RAM y de espacio en disco duro requerido para jugar al videojuego (ambos enteros). Define el constructor correspondiente para asignar estos valores (y utiliza **super** para llamar al constructor del padre para establecer los valores heredados). Luego añade algunos videojuegos de PC un *array* y repite los mismos pasos que en el ejercicio anterior.

Sobrescribe también el método **toString** en la clase *VideoJuego* de forma que podamos mostrar en pantalla la información de un videojuego con una simple llamada a **System.out.println**.

2.6.3.2. Añade una variable *static* en la clase *VideoJuego* del ejercicio anterior para almacenar cuantos videojuegos (de cualquier tipo) han sido creados. Cada vez que se llame al constructor de *VideoJuego*, incrementa el valor de esta variable. Después de cargar todo el *array*, muestra el valor final del contador.

```
import java.util.Scanner;

/*
 * Main class
 */
public class VideoGameList3
{
    public static final int MAX_VIDEOGAMES = 5;

    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        VideoGame[] games = new VideoGame[MAX_VIDEOGAMES];

        games[0] = new VideoGame("Mario Kart", "Sport", 25);
        games[1] = new PCVideoGame("Call of Duty", "Shooter", 19, 1, 600);
        games[2] = new PCVideoGame("Fifa 17", "Sports", 30, 2, 900);
        games[3] = new VideoGame("Kingdom Hearts", "Adventure", 30);
        games[4] = new VideoGame("Heavy Rain", "Adventure", 50);

        for (int i = 0; i < games.length; i++)
        {
            System.out.println(games[i]);
        }
    }
}
```

```
/*
 * VideoGame parent class
 */
class VideoGame
{
    protected String title;
    protected String genre;
    protected float price;
    protected Company company;

    public VideoGame()
    {
        title = "";
        genre = "";
        price = 0;
    }

    public VideoGame(String title, String genre, float price)
    {
        this.title = title;
        this.genre = genre;
        this.price = price;
    }

    public String getTitle()
    {
        return title;
    }

    public void setTitle(String title)
    {
        this.title = title;
    }

    public String getGenre()
    {
        return genre;
    }

    public void setGenre(String genre)
    {
        this.genre = genre;
    }

    public float getPrice()
    {
        return price;
    }

    public void setPrice(float price)
    {

```

```
        this.price = price;
    }

    public Company getCompany()
    {
        return company;
    }

    public void setCompany(Company company)
    {
        this.company = company;
    }

    // Overriding toString method

    @Override
    public String toString()
    {
        return title + " (" + genre + ", " + price + " eur.)";
    }
}
```

```
/*
 * Company class
 */
class Company
{
    private String name;
    private int foundationYear;

    public Company(String name, int foundationYear)
    {
        this.name = name;
        this.foundationYear = foundationYear;
    }

    public String getName()
    {
        return name;
    }

    public void setName(String name)
    {
        this.name = name;
    }

    public int getFoundationYear()
    {
        return foundationYear;
    }
}
```

```
}

public void setFoundationYear(int foundationYear)
{
    this.foundationYear = foundationYear;
}
}
```

```
/*
 * PCVideoGame subclass
 */
class PCVideoGame extends VideoGame
{
    private int minimumRAM;
    private int minimumHD;

    public PCVideoGame(String title, String genre,
                        int price, int minimumRAM,
                        int minimumHD)
    {
        super(title, genre, price);
        this.minimumRAM = minimumRAM;
        this.minimumHD = minimumHD;
    }

    public int getMinimumRAM()
    {
        return minimumRAM;
    }

    public void setMinimumRAM(int minimumRAM)
    {
        this.minimumRAM = minimumRAM;
    }

    public int getMinimumHD()
    {
        return minimumHD;
    }

    public void setMinimumHD(int minimumHD)
    {
        this.minimumHD = minimumHD;
    }

    // Overriding toString method and using "super" to call parent's
    // toString

    @Override
```

```
public String toString()
{
    return super.toString() + " " + minimumRAM + "GB, " + minimumHD +
"MB";
}
}
```