

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

23 май 2023 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 2

ЧАСТ 1 (Време за работа: 90 минути)

Отговорите на задачите от 1. до 16. включително отбелязвайте в листа за отговори!

1. На кой етап от жизнения цикъл на информационна система се създават детайлни спецификации за системата, включително архитектурата на софтуера и хардуера, базите данни и интерфейсите на потребителите?

А) Планиране Б) Разработка В) Дизайн Г) Внедряване

2. Кое от твърденията НЕ е вярно за статичен член на клас:

А) Полета, методи и свойства, маркирани като статични, принадлежат на самия клас, а не на някой създаден конкретен обект от този клас.

Б) За да се използват статични методи и свойства на класа, е достатъчно да има достъп или видимост до самия клас.

В) Ако има създадени различни инстанции от един клас, тогава статичните полета и свойства ще бъдат общи за тях и ще има само едно копие на статичното поле или свойство, което се споделя от всички обекти от дадения клас.

Г) Статичните полета и свойства на клас не могат да променят стойността си в методите на класа, в който са дефинирани.

3. Даден е следният клас:

C#
<pre>class Test { public static string Change(bool check) { if (check = true) return "true"; else return "false"; } static void Main() { Console.WriteLine(Change(false)); } }</pre>
Java
<pre>public class Test { public static String change(boolean check){ if(check = true) return "true"; else return "false"; } public static void main(String[] args) { System.out.println(change(false)); } }</pre>

Изпълнението на програмния фрагмент води до извеждане на:

- А) съобщение за синтактична грешка
- Б) true
- В) NullPointerException
- Г) false

4. Дефиниран е клас Room. На отбелязаните с (1), (2), (3) и (4) места в програмния код, дефиниращ класа Room, са пропуснати да се запишат модификаторите за достъп. Изберете в кой от отговорите са записани правилните модификатори за всяка от цифрите, като се спазва принципът на капсулация.

C# <pre> class Room { <u>(1)</u> int floor; <u>(2)</u> double area; <u>(3)</u> Room (int floor, double area) { this.floor = floor; this.area = area; } <u>(4)</u> double Volume (double height) { return area * height; } } </pre>
Java <pre> class Room { <u>(1)</u> int floor; <u>(2)</u> double area; <u>(3)</u> Room (int floor, double area) { this.floor = floor; this.area = area; } <u>(4)</u> double volume (double height) { return area * height; } } </pre>

- A) (1) private, (2) private, (3) public, (4) public;
- Б) (1) public, (2) public, (3) private, (4) private;
- B) (1) public, (2) public, (3) private, (4) public;
- Г) (1) private, (2) public, (3) private, (4) private.

5. На етап проектиране от жизнения цикъл на информационна система се изработва:

- А) технико-икономическо задание
- Б) документация
- В) детайлизиран модел
- Г) изискване към архитектурата

6. Какво ще се случи при стартиране на дадения програмен код?

C#	Java
<pre>static void Main(string[] args) { int p; while (p != 10) { Console.WriteLine(++p); } }</pre>	<pre>public static void main(String[] args){ int p; while (p != 10) { System.out.println(++p); } }</pre>

- А) Грешка при компилация
- Б) Грешка по време на изпълнение
- В) Програмата ще изведе числата от 1 до 10 всяко на нов ред.
- Г) Програмата ще изведе числата от 0 до 9, на един ред, разделени с интервал

7. В дадената таблицата **books**, съдържаща се в база от данни, са въведени следните записи:

id	title	author	year	price
1	Complot	Nicolas Beuglet	2020	20
2	The hound of the Baskervilles	Arthur Conan Doyle	2006	12.5
3	Sooley	John Grisham	2021	19
4	Girl you left behind	Jojo Moyes	2015	17.95
5	The kite runner	Khaled Hosseini	2018	18
6	Codex	Tom Egeland	2018	21
7	The lost symbol	Dan Brown	2009	19.89
8	The name of the rose	Umberto Eco	1980	24.9

Какъв ще бъде резултатът от изпълнението на заявката:

```
SELECT id,title FROM books
WHERE year NOT BETWEEN 2010 AND 2019
ORDER BY price DESC;
```

A)

id	title
8	The Name of the Rose
1	Complot
7	The lost symbol
3	Sooley
2	The Hound of the Baskervilles

Б)

id	title
2	The Hound of the Baskervilles
3	Sooley
7	The lost symbol
1	Complot
8	The Name of the Rose

В)

id	title
4	Girl you left behind
5	The Kite Runner
6	Codex

Г)

id	title
6	Codex
5	The Kite Runner
4	Girl you left behind

8. Потребител трябва да избере точно една от 99 допустими стойности за поле, съдържащо име на град. Кое от следните е най-подходящият елемент на потребителския интерфейс, с който може да бъде направен този избор?

- A) радио бутони (radio buttons)
- Б) полета за отметка (check boxes)
- В) падащо меню (drop-down menu)
- Г) поле за свободен текст (text box)

9. Даденият по-долу програмен код трябва да демонстрира работата с колекция, в която да се съхраняват уникални стойности. Отбележете коя от дадените по-долу възможности трябва да се постави на мястото на XXX за тази цел.

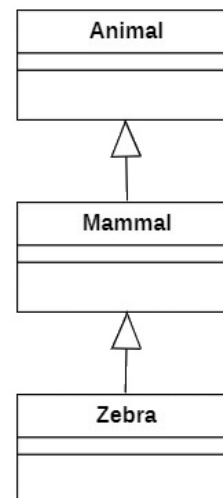
C#
<pre>XXX<string> obj = new XXX<string>(); obj.Add("apple"); obj.Add("banana"); obj.Add("orange"); if (obj.Contains("apple")) { Console.WriteLine("Contains an apple"); } obj.Remove("banana"); foreach (string fruit in obj) { Console.WriteLine(fruit); }</pre>
Java
<pre>XXX<String> obj = new XXX<>(); obj.add("apple"); obj.add("banana"); obj.add("orange"); if (obj.contains("apple")) { System.out.println("Contains an apple"); } obj.remove("banana"); for (String fruit : obj) { System.out.println(fruit); }</pre>

- A) List
- Б) ArrayList
- В) HashSet
- Г) Dictionary (аз C#) / Map (за Java)

10. Достъпът до елементите на структура от данни стек се осъществява:

- A) по индекса на елемента
- Б) от първия добавен в стека елемент
- В) от последния добавен в стека елемент
- Г) по стойност

11. Дадената UML диаграмата онагледява връзките между класовете **Animal**, **Mammal** и **Zebra**. Определете в кой от отговорите дефиницията на класовете съответства на връзките, описани в диаграмата.



A)

C#	Java
<pre> class Zebra { } class Mammal : Zebra { } class Animal : Mammal { } </pre>	<pre> class Zebra { } class Mammal extends Zebra { } class Animal extends Mammal { } </pre>

Б)

C#	Java
<pre> class Animal { } class Mammal : Animal { } class Zebra : Mammal { } </pre>	<pre> class Animal { } class Mammal extends Animal { } class Zebra extends Mammal { } </pre>

В)

C#	Java
<pre> class Mammal : Animal { } class Zebra : Animal { } class Zebra : Mammal { } </pre>	<pre> class Mammal extends Animal { } class Zebra extends Animal { } class Zebra extends Mammal { } </pre>

Г)

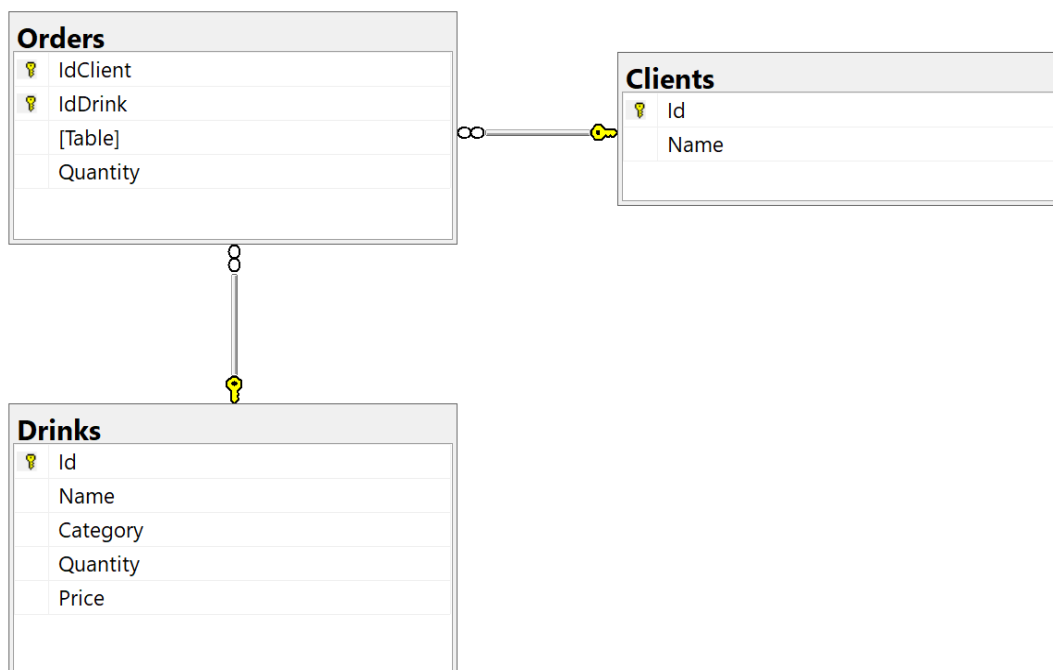
C#	Java
<pre> class Zebra { } class Animal : Zebra { } class Mammal : Zebra { } </pre>	<pre> class Zebra { } class Animal extends Zebra { } class Mammal extends Zebra { } </pre>

12. Отбележете какъв ще е изходът от следния програмен фрагмент:

```
C#  
Stack<string> words = new Stack<string>();  
words.Push("one");  
words.Push("two");  
words.Push("three");  
words.Push("four");  
words.Pop();  
words.Pop();  
  
Console.WriteLine(Array.LastIndexOf(words.ToArray(), "three"));  
  
Java  
Stack<String> words = new Stack<>();  
words.push("one");  
words.push("two");  
words.push("three");  
words.push("four");  
words.pop();  
words.pop();  
  
System.out.println(words.lastIndexOf("three"));
```

- A) 1 Б) 0 В) -1 Г) Програмният код не е коректен

13. На диаграмата са представени таблици, свързани с продажбата на напитки. Кое от твърденията за отношенията между таблиците е вярно?



- A) Връзката между таблиците Clients и Drinks е едно към едно
- Б) Връзката между таблиците Clients и Drinks е много към много
- В) Id на таблицата Clients и Id на таблицата Drinks са външни ключове за тези таблици
- Г) Връзката между таблиците Clients и Orders е много към много

14. Отбележете какъв резултат ще се изведе на стандартния изход след изпълнение на програмата:

C#	Java
<pre>static void Main(string[] args) { MetodN(6); } public static void MetodN(int n) { if (n > 0) { MetodN (n / 2); Console.Write (n % 2); } }</pre>	<pre>public static void main(String[] args) { metodN(6); } public static void metodN (int n) { if (n > 0) { metodN (n / 2); System.out.print (n % 2); } }</pre>

- A) 310 Б) 011 В) 110 Г) 013

15. Какъв резултат ще се изведе на стандартния изход след изпълнение на следния програмен фрагмент:

C#
<pre>double number = 6.78; number += Math.Floor(number); number = Math.Ceiling(number); Console.WriteLine(\$"{number:f3}");</pre>
Java
<pre>double number = 6.78; number += Math.floor(number); number = Math.ceil(number); System.out.println(String.format("%.3f", number));</pre>

- A) 12.78 Б) 13 В) 13.78 Г) 13.000

16. В коя от следните двойки първият обект е във връзка от тип „е“ (is-a) с втория?

- А) документ – папка
- Б) папка – документ
- В) SMS съобщение – текст
- Г) текст – SMS съобщение

Отговорите на задачите от 17. до 24. вкл. запишете в листите за отговори!

17. В база от данни, таблицата `restaurants`, за всеки ресторант са съхранени следните данни: уникален идентификатор (`id`), име (`name`) и брой звезди, с които е категоризиран (`stars`).

В таблицата `cooks`, за всеки готвач са съхранени следните данни: уникален идентификатор (`id`), име (`name`) и идентификатор на ресторанта, в който работи (`restaurant_id`).

Дадена е следната SQL заявка, която извежда имената на всички готвачи, подредени по азбучен ред:

```
SELECT name
FROM cooks
ORDER BY name;
```

В листа за отговори напишете променената заявката така, че тя да извежда само тези готвачи, които работят в 5-звезден ресторант.

18. В стек последователно са постъпили елементите със стойност 23, 4, -5, 0, -2, -1, 15, а в опашка: 6 и -2. От стека се изважда елемент и ако той е отрицателен, се добавя в опашката, а от нея се изважда един елемент. Действието се повтаря до изваждането на елемент със стойност 0 от стека.

В листа за отговори срещу (1), (2) и (3) запишете отговорите на въпросите:

- 1) Колко са елементите в стека?
- 2) Колко е броят на елементите в опашката?
- 3) Кои са елементите на опашката?

19. В дадения текст по-долу е обяснен алгоритъма за търсене на минимален елемент в едномерен масив и след това е записан програмният му код. И в двата записа има пропуснати части, отбелязани с цифри от (1) до (6). Попълнете липсващите думи в текста и пропуснатото в програмния код. Запишете ги в листа за отговори срещу съответния номер.

„При търсене на минимален елемент в масив в началото се приема, че (1) елемент на масива е най-малък и неговата стойност се запазва в променлива. Тази променлива се сравнява с всички останали елементи и когато се намери елемент с (2) стойност от нейната, се (3) неговата стойност в променливата.“

```
int[] arr = {12, 9, -13, 4, -27, 13, 29, 50};
int n = (4);
int minEl = arr[0];
for (int i = 1; i < n; i++)
{
    if ( (5) )
    {
        (6) = arr[i];
    }
}
```

20. Даден е програмният код:

C#	Java
<pre>static void Main(string[] args) { string word = "abba"; for (int i = 0; i < 3; i++) { word.Substring(2); word = word + word; } Console.WriteLine(word); }</pre>	<pre>public static void main(String[] args){ String word = "abba"; for(int i = 0; i<3; i++){ word.substring(2); word = word + word; } System.out.println(word); }</pre>

В листа за отговори запишете колко пъти ще се срещне буквата “а” в отпечатания на стандартния изход текст?

21. В листа за отговори запишете с какво трябва да се заменят празните места, отбелязани с многоточие в изреченията, така че коректно да са описани връзките между класове и данни, представени в следния програмен фрагмент. (В листа за отговори срещу (1), (2) и (3) запишете съответните думи или изрази.)

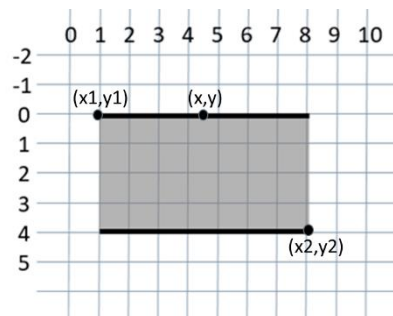
1) Author ... Book

2) Author ... Person

3) Book ... title

C#	Java
<pre>public class Book{ private string title; private int year; } public class Person{ private string name; } public class Author : Person{ private Book bestSeller; }</pre>	<pre>public class Book{ private String title; private int year; } public class Person{ private String name; } public class Author extends Person{ private Book bestSeller; }</pre>

22. Даденият програмен код описва метод, който определя дали точка с координати (x, y) лежи на някоя от двете хоризонтални страни от контура на правоъгълник (изчертани с удебелени линии на изображението). Правоъгълникът е зададен с горен ляв връх с координати (x1, y1) и долен десен връх с координати (x2, y2), като страните му са успоредни на координатните оси. В описанието на метода е допусната логическа грешка.



```
C#
static bool IsOnHorizontalBorder (int x, int y, int x1, int y1, int x2, int y2)
{
    bool onUpperLine = (y == y1) && (x >= x1) && (x <= x2);
    bool onLowerLine = (x == x1) && (y >= y1) && (y <= y2);
    return (onUpperLine || onLowerLine);
}
```

```
Java
static boolean isOnHorizontalBorder (int x, int y, int x1, int y1, int x2, int y2)
{
    boolean onUpperLine = (y == y1) && (x >= x1) && (x <= x2);
    boolean onLowerLine = (x == x1) && (y >= y1) && (y <= y2);
    return (onUpperLine || onLowerLine);
}
```

В листа за отговори срещу (1) и (2) запишете:

- 1) На коя от променливите е присвоена грешна стойност;
- 2) Израз, задаващ правилната стойност.

23. На изображението е дадено съдържанието на текстов файл *words.txt*, който съдържа думи, записани на отделни редове.

- (1) В листа за отговори срещу (1) напишете какво е съдържанието на файла *groups.txt*, след изпълнението на програмния фрагмент.
- (2) В листа за отговори срещу (2) напишете задължително ли е извикването на метода `Close()` (за C#) или `close()` (за Java) за обектите *f1* и *f2*? Защо?

words.txt
table
bed
TV
computer
DVD
parket
water
other

```
C#
try
{
    StreamReader f1 = new StreamReader("words.txt");
    StreamWriter f2 = new StreamWriter("groups.txt");
    int count = 0;
    string word = f1.ReadLine();
    while (word != null)
    {
        count++;
        f2.Write(word);
        if (count % 5 == 0) f2.WriteLine();
        else f2.Write(" ");
        word = f1.ReadLine();
    }
    f1.Close();
    f2.Close();
}
catch (Exception ex)
{
    Console.WriteLine("An error occurred: " + ex.Message);
}
```

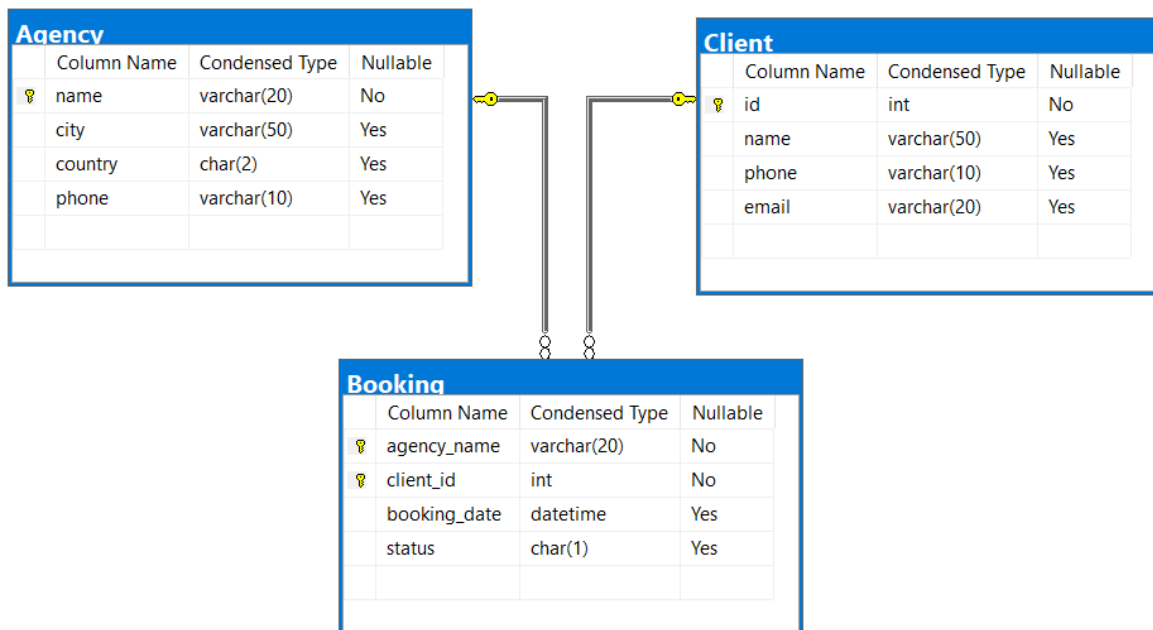
Java

```

try {
    Scanner f1 = new Scanner(new FileInputStream("words.txt"));
    PrintWriter f2 = new PrintWriter("groups.txt");
    int count = 0;
    while (f1.hasNext()) {
        String word = f1.nextLine();
        count++;
        f2.write(word);
        if (count % 5 == 0) {
            f2.println();
        } else {
            f2.write(" ");
        }
    }
    f1.close();
    f2.close();
} catch (Exception ex) {
    System.out.println("An error occurred: " + ex.getMessage());
}

```

24. Дадена е следната диаграма на релационна база от данни.



Написана е следната заявка на SQL, чрез която да се създаде таблицата *Booking* от диаграмата със съответните връзки към останалите таблици. В заявката има пропуски. В листа за отговори пренапишете правилно дадената заявка.

```

CREATE TABLE Booking(
    agency_name VARCHAR(20) NOT NULL,
    client_id int NOT NULL,
    booking_date DATETIME,
    status char(1)
);

```

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

23 май 2023 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 2

ЧАСТ 2 (Време за работа: 150 минути)

Файловете с отговорите на задачите от 25. до 28. включително запишете в изпитната система, като спазите указанията в условието на задачата!

Внимание! Имената на работните файлове, които прикачвате в изпитната система, **НЕ** трябва да съдържат текстове или символи, които могат да доведат до нарушаване на анонимността на изпитната Ви работа!

25. Да се напише приложение, което по въведено цяло число да казва дали то се дели на всяка една от цифрите си. Програмата да прихваща възможните изключения и да извежда съобщението "Something went wrong!". Програмата да извежда "Yes", ако въведеното число се дели на всяка от цифрите си и "No" в противен случай. При възникнало изключение не се очаква никакъв отговор, а само съобщението описано по-горе.

Пример:

Вход: 24 Изход: Yes

Вход: 23 Изход: No

Вход: -12 Изход: Yes

**Забележки:*

Приемат се и решения с графичен потребителски интерфейс (ГПИ), в които цялото число k се въвежда в подходяща контрола, а резултатът се извежда в етикет или редактируемо текстово поле.

Прикачете в изпитната система архив с име zad_25.zip, съдържащ файловете с Вашите решения.

26. Създайте проект с име *Zad_26*, в който се създава списък от думи. От клавиатурата се въвеждат в произволен ред командите: *Add*, *Remove*, *Search*, *Update*, *Length*, *Insert*, *Print* до въвеждане на низа *END*.

След въвеждане на командата *Add* – от стандартния вход на един ред се въвеждат думи, разделени с интервал, които трябва да се съхранят в създадения списък.

След въвеждане на командата *Update* – всички първи символи на думите в списъка, ако са букви трябва да се направят главни, ако са други символи да си останат същите.

След въвеждане на командата *Remove* – от стандартния вход се въвежда цяло положително число и се изтрива елементът от списъка със съответния номер.

След въвеждане на командата *Search* – от стандартния вход се въвежда дума. Ако търсената дума се съдържа в списъка – да се изведе на стандартния изход, ако не – да се изведе съобщението: "Not contained."

След въвеждане на командата *Length* – от стандартния вход се въвежда цяло число *n* и се извеждат на стандартния изход всички низове от списъка с дължина *n*. Низовете се извеждат на един ред, разделени с тире, а ако няма - да се изведе съобщението: "Not contained."

След командата *Insert* – от стандартния вход се въвежда цяло положително число *n* и символен низ. Низът се добавя в списъка на позиция *n*, ако е възможно. Ако не е възможно да се осъществи вмъкването да се генерира изключение със съобщението: "There are not enough items in the list."

След командата *Print* – на стандартния изход се извежда съдържанието на списъка на един ред, а елементите му са разделени с точка и запетая, последвана от интервал.

Пример:

```
Add matematika informatika IT fizika izpit
```

```
Remove 2
```

```
Add BEL papka
```

```
Update
```

```
Print
```

```
Matematika; Informatika; Fizika; Izpit; BEL; Papka
```

```
Insert 4 Istoria
```


Length 5

Izpit-Papka

Search IT

Not contained.

Insert 9 Matura

There are not enough items in the list.

Insert 3 DZI

Print

Matematika; Informatika; Fizika; DZI; Izpit; Istoria; BEL; Papka

END

**Забележка:*

Резултатът от командата се извежда веднага след въвеждането на командата. На дадените примери изхода от програмата е даден в курсив.

Прикачете в изпитната система архив с име `zad_26.zip`, съдържащ файловете с Вашите решения.

27. Създайте база от данни с име `university`, която съхранява данни за един университет. Базата трябва да съдържа две таблици с имена `students` и `fee`

А) Създайте в базата таблица `students` със следните атрибути:

- `Admission_no` – цяло число – номер на студент;
- `First_name` – низ – първо име на студент;
- `Last_name` – низ – фамилия на студент;
- `City` – низ – град на студент;

Б) Създайте в базата `university` таблица `fee` със следните атрибути:

- `Admission_no` – цяло число – номер на студент;
- `Course` – низ – име на курс, който студент изучава;
- `Amount_paid` – цяло число – такса за курса;

Всички низове трябва да имат размер до 25 символа.

В) Като използвате средствата на средата, в която работите;

- Добавете в таблицата `students` следните кортежи (данните за таблицата са в ресурсен файл с име `resources.txt`):

`Admission_no` `First_name` `Last_name` `City`;

3354	Георги	Георгиев	Варна
4321	Милена	Красимирова	Стара Загора
8345	Михаил	Мartiнов	Пловдив
7555	Антонио	Тачев	Стара Загора
2135	Мартин	Иванов	София

- Добавете в таблицата `fee` следните кортежи (данните за таблицата са в ресурсен файл с име `resources.txt`):

Admission_no	Course	Amount_paid;
3354	Java	2000
7555	C#	1800
4321	SQL	1600
4321	Java	2000
8345	C++	1700

Г) Напишете заявка, с която да се изведе информация за града на студент с номер 8345.

Д) Напишете заявка, с която да се изведе средната такса, платена от студент за записан курс.

Е) Напишете заявка, с която да се промени името на курса с такса 1800 на Java.

Ж) Напишете заявка, с която да се групират курсовете по име на курс и да се изведе броят на записаните студенти във всеки курс.

З) Напишете заявка, с която да се извеждат за всеки студент имената му и имената на записаните от него курсове. Ако има студент без записани курсове, изведете `null` за име на курс.

Прикачете в изпитната система файл с име `zad_27.zip`, съдържащ създадената база от данни и написаните от Вас заявки.

28. Създайте проект с име *Furniture*, в който се създава абстрактен клас с име *Furniture*, който има следните полета:

- `typeProduct` – низ.

- `productionPrice` – реално число – това е цената необходима за производството на 1 брой мебел.

Класът притежава следните методи:

- `get()` и `set()` за полето `typeProduct`, като се има предвид, че за стойност на полето не може да се подава празен низ;
- `get()` и `set()` за полето `productPrice`, като се има предвид, че за стойност на полето не може да се подава отрицателно число или 0.
- абстрактен метод `PriceClient()` (за C#) / `priceClient()` (за Java) от тип `double`, чрез който се изчислява цената за клиенти.

Класът трябва да съдържа конструктор с два параметъра.

Нека класът с име `Table` наследява класа `Furniture`.

А) Реализирайте метода `PriceClient()` (за C#) / `priceClient()` (за Java) като знаете, че цената за клиент е с 20% по-висока от производствената цена (`productionPrice`).

Б) Пренапишете метода `ToString()` (за C#) / `toString()` (за Java) за клас `Table` така, че да връща информацията във вид: „The table costs <`PriceClient()`> lv.”

Цената да бъде закръглена до втори знак след десетичната точка.

Нека класът с име `Cabinet` наследява класа `Furniture` и има собствено поле с име `numberOfHinges` – цяло число

А) Реализирайте метода с име `PriceClient()` (за C#) / `priceClient()` (за Java), като знаете, че цената за клиент е с 15% по-висока от производствената цена (`productionPrice`) и към нея се добавя общата цена за всички допълнителни части. Всяка част струва по 4,50.

Б) Пренапишете метода `ToString()` (за C#) / `toString()` за клас `Cabinet` така, че да връща информацията във вид: „The cabinet costs <`PriceClient()`> lv.”.

Цената да бъде закръглена до втори знак след десетичната точка.

В главната програма създайте списък от мебели.

Информацията ще се въвежда за всяка мебел на един ред, разделена с интервал във формат:

```
<typeProduct> <productionPrice>
```

или

```
<typeProduct> <productionPrice> <numberOfHinges>
```

Въвеждането да приключва след въвеждане на низа „END”.

Програмата да изведе мебелите, групирани по тип: първо масите, а след това и шкафовете.

Примерен вход:

```
table 100
table 120.50
cabinet 200 10
table 80
cabinet 220.50 5
cabinet 300 10
table 95
END
```

Изход:

```
All tables:
The table costs 120.00 lv.
The table costs 144.60 lv.
The table costs 96.00 lv.
The table costs 114.00 lv.
All cabinets:
The cabinet costs 275.00 lv.
The cabinet costs 276.08 lv.
The cabinet costs 390.00 lv.
```

Прикачете в изпитната система архив с име zad_28.zip, съдържащ файловете с Вашите решения.

МИНИСТЕРСТВО НА ОБРАЗОВАНИЕТО И НАУКАТА

ДЪРЖАВЕН ЗРЕЛОСТЕН ИЗПИТ ПО

ИНФОРМАТИКА

23 май 2023 г.

ПРОФИЛИРАНА ПОДГОТОВКА

ВАРИАНТ 2

Задача от 1. до 16. Ключ с верните отговори

Въпрос №	Верен отговор	Брой точки
1.	В	1
2.	Г	1
3.	Б	1
4.	А	1
5.	В	1
6.	А	1
7.	А	1
8.	В	1
9.	В	1
10.	В	1
11.	Б	1
12.	В	1
13.	Б	1
14.	В	1
15.	Г	1
16.	В	1

Задача 17. Примерно решение – 3 точки

```
SELECT cooks.name
FROM cooks
JOIN restaurants ON restaurant_id = restaurants.id
WHERE stars = 5
ORDER BY cooks.name;
```

Задача 18. Примерно решение – 3 точки

- 1) Броят на елементите в стека е 3
- 2) Броят на елементите в опашката е 2
- 3) Елементите в опашката са -1 и -2

Задача 19. Примерно решение – 3 точки

- (1) първият / последният / произволен
- (2) по-малка
- (3) записва / присвоява / запомня / замества
- (4) `arr.Length` (за C#) / `arr.lenght` (за Java)
- (5) `arr[i] < minEl` / `arr[i] <= minEl` / `minEl > arr[i]` /
`minEl >= arr[i]`
- (6) `minEl`

Задача 20. – 3 точки

Отговор: 16

Задача 21. Примерно решение – 3 точки

- 1) *има* или *has-a*
- 2) *е* или *is-a*
- 3) *има* или *has-a*

Задача 22. Примерно решение – 3 точки

1) onLowerLine

2) (y == y2) && (x >= x1) && (x <= x2)

Задача 23. Примерно решение – 3 точки

1) Всички думи от файла "words.txt" ще се запишат във файла "groups.txt" на групи по 5 на ред, разделени с интервал.

groups.txt
table bed TV computer DVD
parket water other

2) Препоръчително е, защото при извикването на методите Close() или close() обектите, за които са били извикани методите приключват нормално работата, т.е. файловете се затварят и се освобождават използваните от тях ресурси.

Задача 24. Примерно решение – 3 точки

```
CREATE TABLE Booking(  
agency_name VARCHAR(20) NOT NULL FOREIGN KEY REFERENCES  
Agency([name]),  
client_id int NOT NULL REFERENCES Client(id),  
booking_date DATETIME,  
[status] char(1),  
PRIMARY KEY(agency_name, client_id)  
);
```

Или

```
CREATE TABLE Booking(  
agency_name VARCHAR(20) NOT NULL,  
client_id int NOT NULL REFERENCES Client(id),  
booking_date DATETIME,  
[status] char(1),  
FOREIGN KEY (agency_name) REFERENCES Agency([name]),  
PRIMARY KEY(agency_name, client_id)  
);
```

Или

```
CREATE TABLE Booking(  
agency_name VARCHAR(20) NOT NULL CONSTRAINT FK1 REFERENCES  
Agency(name),  
client_id int NOT NULL CONSTRAINT FK2 REFERENCES Client(id),  
booking_date DATETIME,
```

```
status char(1),
PRIMARY KEY(agency_name, client_id)
);
```

Или

```
CREATE TABLE booking(
agency_name VARCHAR(20) NOT NULL,
client_id int NOT NULL,
booking_date DATE,
status char(1),
FOREIGN KEY(agency_name) REFERENCES Agency(name),
FOREIGN KEY(client_id) REFERENCES Client(id),
PRIMARY KEY(agency_name, client_id)
);
```

Задача 25. Примерно решение – 10 точки

За C#

```
using System;
namespace DZI
{
    class Zad25
    {
        static void Main(string[] args)
        {
            try
            {
                int number = Int32.Parse(Console.ReadLine());
                int copy = number;
                while (copy != 0)
                {
                    if (copy % 10 == 0 || number % (copy % 10) != 0)
                    {
                        Console.WriteLine("No");
                        return;
                    }
                    copy /= 10;
                }
                Console.WriteLine("Yes");
            }
            catch (Exception e)
            {
                Console.WriteLine("Something went wrong!");
            }
        }
    }
}
```


3a Java

```
package variant1;

import java.util.Scanner;

public class Zad25 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        try{

            int number = input.nextInt();

            int copy = number;

            while(copy != 0){

                if(copy % 10 == 0 || number % (copy % 10) != 0){

                    System.out.println("No");

                    return;

                }

                copy /= 10;

            }

            System.out.println("Yes");

        }

        catch(Exception e){

            System.out.println("Something went wrong!");

        }

    }

}
```

Задача 26: Примерно решение – 15 точки

За С#

```
using System;
using System.Collections.Generic;
using System.Linq;
namespace DZI
{
    class Zad26
    {
        static void Main(string[] args)
        {
            List<string> words = new List<string>();

            var input = Console.ReadLine().Split(' ').ToList();
            while (input[0] != "END")
            {
                switch (input[0])
                {
                    case "Add":
                        for (int i = 1; i < input.Count; i++)
                            words.Add(input[i]);
                        break;
                    case "Update":
                        for (int i = 0; i < words.Count; i++)
                        {
                            string firstLetter =
                                words[i].Substring(0, 1).ToUpper();
                            string restOfWord =
                                words[i].Substring(1);
                            words[i] = firstLetter + restOfWord;
                        }
                        break;
                    case "Remove":
                        int index = int.Parse(input[1]);
                        words.RemoveAt(index);
                        break;
                    case "Search":
                        string search = input[1];
                        if (words.IndexOf(search) != -1)
                            Console.WriteLine(words[words.IndexOf(search)]);
                        else Console.WriteLine("Not contained.");
                        break;
                    case "Length":
```

```

        int length = int.Parse(input[1]);
        var result = words.Where(x => x.Length ==
                                length).ToList();
        if (result.Any())
            Console.WriteLine(string.Join("-",
                                        result));
        else Console.WriteLine("Not contained.");
        break;
    case "Insert":
        int position = int.Parse(input[1]);
        string item = input[2];
        try
        {
            words.Insert(position, item);
        }
        catch (Exception ex)
        {
            Console.WriteLine("There are not enough
                                items in the list.", ex);
        }
        break;
    case "Print":
        Console.WriteLine(string.Join("; ", words));
        break;
    }
    input = Console.ReadLine().Split(' ').ToList();
}
}
}
}
}
}
}

```

3a Java

```
package variant1;
```

```
import static java.lang.Character.toUpperCase;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.stream.Collectors;
public class Zad26 {

    public static void main(String[] args) {
        List<String> words = new ArrayList<>();
        Scanner input = new Scanner(System.in);
        String in[] = input.nextLine().split(" ");
        while (!in[0].equals("END")) {
            switch (in[0]) {
                case "Add":
                    for (int i = 1; i < in.length; i++) {
                        words.add(in[i]);
                    }
                    break;
                case "Remove":
                    words.remove(Integer.parseInt(in[1]));
                    break;
                case "Search":
                    if (!words.contains(in[1])) {
                        System.out.println("Not contained.");
                    } else {
                        System.out.println(in[1]);
                    }
            }
        }
    }
}
```

```
        break;
    case "Update":
        int j = 0;
        for (String w : words) {
            char firstLetter = toUpperCase(w.charAt(0));
            String lastLetters = w.substring(1);
            w = firstLetter + lastLetters;
            words.set(j, w);
            j++;
        }
        break;
    case "Length":
        int length = Integer.parseInt(in[1]);
        List<String> wordsByLenght =
            (ArrayList<String>) words.stream().
            filter(x -> x.length() == length).
            collect(Collectors.toList());
        System.out.println(String.join("-",
            wordsByLenght));
        break;
    case "Insert":
        try {
            words.add(Integer.parseInt(in[1]), in[2]);
        } catch (IndexOutOfBoundsException ex) {
            System.out.println("There are not enough items
                in the list.");
        }
        break;
    case "Print":
```

```

        System.out.println(String.join("; ", words));
    }
    in = input.nextLine().split(" ");
}
}
}

```

Задача 27: Примерно решение – 15 точки

```
DROP DATABASE IF EXISTS University;
```

```
CREATE DATABASE IF NOT EXISTS University;
```

```
USE University;
```

```
CREATE TABLE Student (
```

```
    admission_no int PRIMARY KEY,
```

```
    first_name varchar(25) NOT NULL,
```

```
    last_name varchar(25) NOT NULL,
```

```
    city varchar(25) NOT NULL
```

```
);
```

```
CREATE TABLE Fee (
```

```
    admission_no int NOT NULL,
```

```
    course varchar(25) NOT NULL,
```

```
    amount_paid int
```

```
);
```

```
INSERT INTO Student (admission_no, first_name, last_name, city)
```

```
VALUES (3354, 'Георги', 'Георгиев', 'Варна'),
```

```
(4321, 'Милена', 'Красиминова', 'Стара Загора'),
```

```
(8345, 'Михаил', 'Мартинов', 'Пловдив'),
```

```
(7555, 'Антонио', 'Тачев', 'Стара Загора'),
```

```
(2135, 'Мартин', 'Иванов', 'София');  
INSERT INTO Fee (admission_no, course, amount_paid)  
VALUES (3354, 'Java', 2000),  
(7555, 'C#', 1800),  
(4321, 'SQL', 1600),  
(4321, 'Java', 2000),  
(8345, 'C++', 1700);
```

```
SELECT city FROM student WHERE admission_no = 8345;  
SELECT Avg(amount_paid) FROM fee;  
UPDATE fee SET course = 'Java' WHERE amount_paid = 1800;  
SELECT first_name, last_name, course  
FROM student  
LEFT JOIN fee ON student.admission_no = fee.admission_no;  
SELECT course, COUNT(course)  
FROM fee  
GROUP BY course;
```

Задача 28: Примерно решение – 20 точки

За C#

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
  
namespace DZI  
{  
    class Zad28  
    {  
        static void Main(string[] args)  
        {  
            List<Furniture> furnitures = new List<Furniture>();
```

```

var input = Console.ReadLine().Split(' ').ToList();
while (input[0] != "END")
{
    if (input[0] == "table")
    {
        Table table = new Table(input[0],
            double.Parse(input[1]));
        furnitures.Add(table);
    }
    if (input[0] == "cabinet")
    {
        Cabinet cabinet = new Cabinet(input[0],
            double.Parse(input[1]), int.Parse(input[2]));
        furnitures.Add(cabinet);
    }

    input = Console.ReadLine().Split(' ').ToList();
}

Console.WriteLine("All tables:");
foreach (Furniture furniture in furnitures)
{
    if (furniture is Table)
        Console.WriteLine(furniture.ToString());
}

Console.WriteLine("All cabinets:");
foreach (Furniture furniture in furnitures)
{
    if (furniture is Cabinet)
        Console.WriteLine(furniture.ToString());
}
}
}

namespace DZI
{
    public class Cabinet : Furniture
    {
        private int numberOfhinges;
    }
}

```



```

    public int NumberOfHinges
    {
        get { return numberOfhinges; }
        set { numberOfhinges = value; }
    }

    public Cabinet(string type, double price, int
numberofhinges) : base(type, price)
    {
        this.NumberOfHinges = numberofhinges;
    }

    public override double PriceClient()
    {
        return base.ProductionPrice * 1.15 + this.NumberOfHinges
* 4.50;
    }

    public override string ToString()
    {
        return $"The cabinet costs {PriceClient():f2} lv.";
    }
}
}

namespace DZI
{
    public abstract class Furniture
    {
        private string typeProduct;
        private double productionPrice;
        public string TypeProduct
        {
            get { return typeProduct; }

```

```

        set
        {
            if (string.IsNullOrEmpty(value))
                throw new Exception("Type not empty.");
            else typeProduct = value;
        }
    }

    public double ProductionPrice
    {
        get { return productionPrice; }
        set
        {
            if (value <= 0)
                throw new Exception("Price not negative.");
            else productionPrice = value;
        }
    }
}

namespace DZI
{
    public class Table : Furniture
    {
        public Table(string type, double price) : base(type, price)
        {
        }

        public override double PriceClient()
        {
            return base.ProductionPrice * 1.20;
        }

        public override string ToString()
        {

```

```

        return $"The table costs {PriceClient():f2} lv.";
    }
}
}

```

3a Java

```

public class Zad28 {
    public static void main(String[] args) {
        ArrayList<Furniture> list = new ArrayList<>();
        Scanner scan = new Scanner(System.in);
        String in[] = scan.nextLine().split(" ");
        while (in.length != 1) {
            if(in.length == 3)
                list.add(new Cabinet(in[0],
                    Double.parseDouble(in[1]), Integer.parseInt(in[2])));
            else
                list.add(new Table(in[0],
                    Double.parseDouble(in[1])));
            in = scan.nextLine().split(" ");
        }
        System.out.println("All tables:");
        for(Furniture furniture : list) {
            if(furniture.getTypeProduct().equals("table"))
                System.out.println(furniture.toString());
        }
        System.out.println("All cabinets:");
        for(Furniture furniture : list) {
            if(furniture.getTypeProduct().equals("cabinet"))
                System.out.println(furniture.toString());
        }
    }
}

```

```

}

public class Cabinet extends Furniture {
    private int numberOfHinges;
    public int getNumberOfHinges() {
        return numberOfHinges;
    }
    public void setNumberOfHinges(int numberOfHinges) {
        this.numberOfHinges = numberOfHinges;
    }
    @Override
    public double priceClient() {
        return 1.15*this.getProductionPrice() + numberOfHinges*4.5;
    }
    @Override
    public String toString() {
        return String.format("The cabinet costs %.2f lv.",
this.priceClient());
    }

    public Cabinet(String typeProduct, double productionPrice, int
numberOfHinges) {
        super(typeProduct, productionPrice);
        this.numberOfHinges = numberOfHinges;
    }
}

public abstract class Furniture {
    private String typeProduct;
    private double productionPrice;
    public String getTypeProduct() {
        return typeProduct;
    }
}

```

```

public void setTypeProduct(String typeProduct) {
    if(!typeProduct.equals(""))
        this.typeProduct = typeProduct;
}

public double getProductionPrice() {
    return productionPrice;
}

public void setProductionPrice(double ProductionPrice) {
    if(ProductionPrice > 0)
        this.productionPrice = ProductionPrice;
}

public class Table extends Furniture {
    public Table(String typeProduct, double ProductionPrice) {
        super(typeProduct, ProductionPrice);
    }

    @Override
    public double priceClient() {
        return 1.2*this.getProductionPrice();
    }

    @Override
    public String toString() {
        return String.format("The table costs %.2f lv.",
this.priceClient());
    }
}

```