

Corso di Laurea in Informatica Prova Scritta di Metodologie di Programmazione - Primo Canale

Sapienza Università di Roma

17 Maggio 2023

Durante l'esame non è consentito l'utilizzo di alcunché. Non è consentito inoltre l'utilizzo della matita o di penne il cui colore sia diverso dal nero o dal blu. Il ritiro dalla prova equivale al mancato superamento dell'esame.

Nome	Δ	Cognome:
nome	е	Cognome.

Matricola:

Domanda	1	2	3	4	5	6	Totale
Punteggio Totale	10	4	3	5	12	8	32
Punteggio Ottenuto							

1.	10 punti Per ogni domanda, indicare con una X la risposta desiderata. Si ricorda che ogni domanda
	ha al più una risposta corretta. L'assegnazione dei punti alle risposte è la seguente. Verranno attribuit
	2punti per ogni risposta esatta; -0.4 punti per ogni risposta errata; 0 punti per ogni risposta omessa
	Al fine del superamento della soglia è necesssario totalizzare un punteggio di almeno 5 punti.
	(a) Un ciclo for è un costrutto di iterazione \bigcirc Indeterminato \bigcirc Determinato \bigcirc Ciclico
	(b) Una costante in Java è indicata dal modificatore \bigcirc static \bigcirc abstract \bigcirc final

- (c) I metodi di un'interfaccia sono protected public private
 (d) Il contenuto di due stringhe è confrontabile tramite il metodo compare equals corresponds
- (e) Un metodo static può accessibile tramite 🂢 la sua classe 🔘 il suo oggetto 🔘 l'uso di alcun modificatore di visibilità
- 2. 4 punti Qual è la differenza tra overloading e overriding? Fornire un esempio minimale, scritto in Java, che descriva quanto richiesto.

Overloading e Overriding sono concetti legati al polimorfismo il quale, nella programmazione ad oggetti, si riferisce alla capacità di una classe A di assumere i valori di un qualunque tipo descritto dalla classe B, con B sottoclasse di A.

Si parla di overloading quando sono presenti più metodi all'interno di una classe o di una sua superclasse con lo stesso nome, tipo di ritorno ma con tipo e/o numero di parametri diverso. Si parla invece di overriding quando un metodo della classe derivata B sovrascrive il metodo della classe A mantenendo quindi la stessa firma ma cambiando il corpo della funzione.

Un esempio minimale in Java di quanto spiegato è fornito nella cartella Es₁.

3. 3 punti Per ogni costrutto iterativo, indicare il numero di volte per il quale viene eseguito il suo corpo. Se non diversamente indicato, si assume che la variabile contatore non venga modificata all'interno del corpo di ciascun costrutto iterativo.

```
(a) for(int i = 12; i <= 7; i++){...}</li>
(b) for(int i = 0; i < 10; i++){...}</li>
(c) for(int i = -10; i >= 0; i++){...}
(d) for(int i = -2; i >= 0; i++){...}
(e) for(int i = -8; i <= 3; i = i + 4){...}</li>
(f) for (int k = 0; k < 20; k+=2) {
    if (k + 3 == 1) System.out.println(k + " ");
    }</li>
```

4. | 5 punti | Spiegare le principali differenze tra ArrayList e Set in Java.

ArrayList implementa l'interfaccia List appartenente a sua volta all'interfaccia Collection di Java. Un ArrayList è una collezione di oggetti memorizzati sequenzialmente secondo una politica FIFO. Esso può contenere valori il cui tipo può essere primitivo o meno, ammettendo duplicati. La dimensione dell'ArrayList è dinamica e l'accesso ai dati è randomico (posizionale).

I Set, appartenenti anch'essi all'interfaccia Collection, rappresentano un insieme di valori senza duplicati. Tale interfaccia Set è a sua volta implementata da HashSet e TreeSet: la prima utilizza una tabella hash, ofrendo quindi un tempo di esecuzione costante per le operazioni di *add, remove, size, contains.* TreeSet, al contrario, utilizza un albero interno per le operazioni di memorizzazione offrendo quindi la possibilità di avere un ordinamento. HashSet permette la presenza di un valore nullo, mentre TreeSet no.

5. 12 punti Realizzare la classe *Customer* al fine di gestire le informazioni relative ad un cliente di una campagna di promozione commerciale. Tale campagna si articola in questo modo: dopo aver effettuato una serie di acquisti per almeno \$100, il cliente riceverà uno sconto di \$10 sull'acquisto successivo. Progettare quindi i seguenti metodi:

```
public void makePurchase(double amount) // registra un acquisto
public boolean discountReached() // true se e solo se l'utente ha raggiunto lo sconto
```

Al fine di verificare la correttezza delle propria implementazione, si realizzi un programma di collaudo (test). Esso dovrà rappresentare la seguente situazione: un utente dovrà ottenere uno sconto, il quale lo utilizzerà per fare degli acquisti. L'importo totale derivante da tali acquisti dovrà essere maggiore di \$90 ma minore di \$100 in modo tale da non poter ottenere uno sconto. Successivamente, l'utente dovrà effettuare un altro acquisto, il quale lo porterà a godere dello sconto sull'acquisto successivo. cliente

6. 8 punti Un supermercato ha accesso ad un database contenente tutte le informazioni sui propri prodotti. Tali informazioni sono rappresentate in formato tabellare, in cui ciascuna riga contiene i seguenti campi separati da uno spazio:

codiceProdotto1 NomeProdotto1 Quantità1 PrezzoPerUnità1 codiceProdotto2 NomeProdotto2 Quantità2 PrezzoPerUnità2

. . .

Scrivere un programma che legga un file di testo con questa struttura, segnalando un opportuno errore in caso di file inesistente. Successivamente, visualizzare:

- Le informazioni relative al prodotto più costoso;
- Il prezzo medio per ciascun prodotto, calcolato come il rapporto tra prezzo per unità e la quantità disponibile. Gestire gli eventuali casi speciali
- $\bullet\,$ I nomi dei prodotti esauriti, ossia con una quantità pari a zero

							3)	9	=	1	_ 1¢	-0	= 10	,	C=	0	d	= 4		2 = 3	348	+1	=2		
(V)(r			NATO	•			-,					7						-6			4		7		
(B)	PIN	AL						۴.	20:	20 =	10														
(c)	PUB	LIC																							
(4)	£0	ι Δ ε																							
	115																								
2) (OVEA	LOA	JING	Ē	ονΕι	2Q 1	SING	SONO	2 0	امدا	erti	₽o.	va A	WENI	'AZ s	ו ומ	NISO	rokF	SMO						
A40 1	1040	11	dÆ4	l na	DING	Q	A N	O VEN	010																ig lia
CAE	671 P	lu co	≻ ™.	かりかい	'Co	y w	97£	SEGUIT	ויון שאית	ESEM	NPIO:	ll Soi	AASC	SIVE,	, wet	RENA	ME)	V CO	1200 b	WA cea Piú	SPE	CIFIC	e, 12	r que	W:
								ntni, N				٥	مناطه	فای	iss <i>j</i>	anim	ماد	ς							
•							י אשייני	מימו, וכ	чичей	02J}		, r			نا ہو			_							
	hune																۸.					In			
	14Me 5 <i>011M</i>						٤.٠						ysk	₩.o	al. fi	/ IM) 1.	un l	ver	o di	un ah	imal	e.) ^j			
Ę							,					١,	5												
المايد	ie c	MM.	ລມູ	LM1 QY	i (day)	nle nu	MM6)	01,doab	le nun	hero2)S	ک	LI:		ممدا	C.		4.	. 1	۸.٠	اد	٥,			
•								J., 10			,,	P	u pii	2 0	10155	a	ane	الاي	hds	Asia	no! I	દ			
	lumei Iumei												୍	ver	ride . void	1	·V.	- 1	١.						
	ONIM						٠,٠						P	lblic	. You	50	ı vei	COL	ع دار م دار		_ ,	٠_			
}														ડે)લી	em.	αĵ.	prisi	th	(")	au bi	au:	١;			
QuE	00	حه۱۷۱۲	: P5	amet.	e d	. Ac	ett	A AR SIA	Migu	INT	CHE		ૢઙૺ												
Æ	Dou	BLE,	SEA	134	CAOSA	AE	נ עס	erpore					ς .						000						
																_				UN					
												1	him	d a	nin	ale	= H &	W	Con	rel)	٤	51 /	NUU	CAN	2
												٨	ELSV	0 F	AIVE	رصود	16	Æls	NoN	SARA	' GE	WEA.	les 1	10 QU	EZZO
												S	rec	NCO	JEι	CM -	دهمي	2							
G) /	001	4. 4	AIEEF	Qe a la	34 (ኤርን/	NBULE	رم اع	ue /	٠. A	4419	/ICT	NEQ/L	مروور	A 44	تمرار	CRF	acci i	1/10	T II	A.E.A	T . C) cei	No.
								ENTO																	
UG.	N) E	MENI	TRE.	/ SE	T No	N So	'1. Wo	UN ELEM ARDINA	שם ול	י או וענינ	BISO	GNA	ITER	IAE S	u Tui	שם של מים	OAN	0	P1/12	20.0		31V 44PP	PV India	(12 E	AINIE()
	·,																			5NN~	<i>,</i> , ,	ME	6 20	Celon	11/4 2(1

f	ic class vivate public balá boole turn (1	Custi Ince	emer = C	() { };		(ا ا	Ę			ρα	0	astol	ner o ner. Ner. o oune. Iner.	custan mai disco r. m disco	ner = ce fu unt l dkef unt R	hen vchá ledo urch each	, cu se la shed ase l ed l'	(); (); (10 3)	er(),	ŗ			
publ is 3	ic void (discou balanc	mak tRe	e Purc Ache	:hase d()\\$	(doy	ble	дm	ount)}	3	3	usto usto	ner Mer	. dis	e Pu wuh	rchð †Re	se (1) ach	ə,∙ ⊌()	;				
el }	se§ balano	e += i	9m04	nt;																			

6) public class Product E	NB.	product.	hame() E	CONTINUESO
private String code; private String home; private : at humber;		produci .n	whober ()	Se
private int number; private cloat frice U;				
Public Product (String cade, String name, int number, gloat price U) &				
this. code = code;				
this have - hame;				
this humber=mumber;				
this. price U = price U.				
public String cole(1) { return cole; 2				
public String name () { return name;				
public int number () {				
return munker.				
public ploat price U() §				
return price();				
public float mid Price () {				
is (Price U == 0) {				
return Float. Nan);				
ξe)seξ				
return price U/number;				
public boolean empty () {				
return (hamber== 0);				
public static Product convert (String line) {				
String [] prod = line. split("");				
¿ return new Product (prod [o], prod[1], Integer parse Int (not), Float parse	Float (pro	J[3J)) ,		
3				

F	ublic c	lass Store	. {														
ĺ	private	Arraylist	' <pre>l'oduct></pre>	products;	ς												
	Public this	Store (Arro, Products = p	ylisi < rod raducts:	uct> products)	2												
	public	void prini	1/h; 1/) E														
	for	(Product	product: f	Products){			-11							_			
	2 5	ystem.out.	printla ("11	presso me	dio dell'	arlico	do"+ p	rodust.	home()+"é	"+ pr	duct	. mid	Price	(1);		
	25																
	5																
		void print															
	50X(P	raduct prod Product.em	uct: froduct	β) ξ													
	171 Si	product.em vetem.out.or	intln (" 2' art	ticolo" + produ	t.name t	نے ۱۱	oc Auv	(ایل									
	_ } } '						C30-4	7									
	33																
		Product se		(){													
	Prod	uct max	= null;														
	for	lint i=o, ; .	products.	.sise(), i++)	ž												
	if	:==0)} max = prod	ucts.pet(i.) .													
					41.4.4												
	ijf	(products. g môs = prod	ed(i) frice	.U() = mds.	price()) <u>}</u>											
	2	mdz = frod	dii.gel(i)	;													
	7 5																
	en den den	mo)n;															
3	}																

	200	M a :.	2																							
رے ہے۔ عزامات	ass i	ic vo	id m	din (ti:ne	57.	ires	2																		
					Y		8 - 4																			
	, ς									_																
	k	eade	= h	eW	Sednh	er (m	ew F	ile ("	elem	ents	.txt"),														
	fc	itch	(File	Notf	ound	Exce	ption	exc)	, ''/ }																	
				T. pp	iatim	ارع")	e hat	Foun	9.)																	
	} "	21414	,																							
Stri	ayli ine 1	ine;			ľ			ur A	fr3}	List	< 7	(),	ŗ													
	line.	= řed	der.i	nextl	ine() :																				
1	roduc	ts.add	I(Pro	duct.	cohiu	eif (l:	he)) _;																			
\\ \text{\chi}	و مرام	tore	- 401	. ઉ	ve 6	rodu	ets)																			
Pro	duct	klam	= sta	re.	searc	hMa:	k();																			
					d1.co	de L)	+ 1)	n t m	os.ha	me!)+ ") + M	dr.Huj	mber (()+"	11+1	nax.	or:ce(ν());							
sT _o . 7	re.pr	iut Be	اوع احم)	· ();																						
S																										
	Arr Str, Wh F Syriste	Array Li String I While (Ine. Product System. Store S Product System. Store S	ublic static wo Scanner rea try & reader f catch System eturn Array List < String line; while (read line = rea products.ad Store store Product man System. out. store. Print l'	Scanner reader, try & reader = n } catch (File System.cu return; } Array List < frodu String line; while (reader.he line = reader. products.add(fro } Store store = neu froduct max = sto System.out. printli store. print hid()	ublic static wid main() Scanner reader; tr; & reader = new } catch (FileNot) System cut. pr. return; } ArrayList < Product = String line; While (reader. nas Ne. line = reader. nextl products. add (Product.) \$ Store Store = new Sto Product max = store. s System. out. println (m store. print hid ();	ublic static void main(string Scanner reader; try & reader = new Scann f catch (FileNat Found System.cut. Arintlm return; } ArrayList < Product = prod String line; while (reader.has NextLine) line = reader.nextLine(products.add (Product.con) Store store = new Store (Product max = store. searc System.out.printlm (max.co	ublic static upid main(string). I canner reader; try & reader = new Scanner (m f catch (FileNot Found Ence System. cut. printlm ("sil return; } Array List < Product = products String line; while (reader. has Next Line()); products. add (Product. convert (li } Store store = new Store (product Product max = store. search Ma: System. out. printlm (max.code L). store. print Mid ();	ublic static void indin(string. L. dirgs. Scanner reader; tr; & reader = new Scanner (new F catch (FileNot Found Enception System.cut. printlin ("sile not return; } ArrayList < froduct = products = ne string line; while (reader. has Next Line()); products. add (Product. convert (line)); Store store = new Store (products) Product max = store. search Max(); System. out. printlin (man.code L)+" store. print hid ();	ulolic static void main(string L) args)? Scanner reader; try & reader = new Scanner (new File! } catch (File Not Found Exception exc) System.cut. Arintlm ("gile not Foun return; } Array (ist < froduct = products = new A string line; while (reader.has Next (ine()); Ine = reader.next line(); products.add (Broduct.convert (line)); \$ store store = new Store (products) Product max = store. search Max(); System.out.printlm (max.code L)+ "+ m store.print Mid();	store store = new Store (products) Scanner reader; tr; & reader = new Scanner (new File l'elem f catch (File Not Found Exception exc) & System. cut. print Im ("File not Found"), return; } Array List < froduct = products = new Array (String line; While (reader. has Mext Line ()) { line = reader. next Line (); products. add (Product. convert (line)); } Store Store = new Store (products) Product max = store. search Max(); System. out. print Im (max.code L) + "" + mox.na store. print Mid ();	static void main (String Llargs)? Scanner reader; try { reader = new Scanner (new File l'elements } catch (File Not Found Exception exc) { System.cut. printlm ("sile not Found"), return; } Array List < Product > products = new Appay List String line; while (reader has Next Line ()) { line = reader next Line (); products.add (Product convert (line)); } Store store = new Store (products) Product max = store. search Max(); System.out. printlm (max.code L) + " "+ mox.name! store. print Mid ();	scanner reader; tr; { reader = new Scanner (new File l'elements.txt) reader = new Scanner (new File l'elements.txt) reader = new Scanner (new File l'elements.txt) reader = new Array ("sile not found"); return; } Array (ist < froduct > products = new Array (ist < > String line; while (reader.next (ine ()) { line = reader.next (ine ()); products.add (Product.convert (line)); } Store Store = new Store (products) Product max = store. search Max(); System.out. println (max.code L) + " + mos.name !) + " store. print Mid ();	static void main (string L) args)? Scanner reader; try { reader = new Scanner (new File l'elements.txt"); } catch (File Not Found Exception exc) { System.cut. printlm ("sile not Found"), return; } Array List < froduct = products = new Apparlist < > (), String line; while (reader.has Vext Line ()) { line = reader.next Line (); products.add (Product.convert (line)); } Store Store = new Store (products) Product max = store. search Max(); System.out. printlm (max.code L) + " "+ mos.name!) + " "+ m store. print Mid();	ublic static void main (String Llargs)? Scanner reader; try & reader = new Scanner (new File l'elements.txt"); } catch (File Not Found Exception exc) { System.cut. Arintlm ("sile not Found"); return; } Array (ist < Product > products = new Appay (List < > (); String line; while (reader has Next (ine ()) { line = reader next (ine ()); products.add (Product convert (line)); } Store store = new Store (products) Product max = store. search Max(); System. out. printlm (max.code l) + "" + max.name () + "" + max.nam store. print Mid ();	Scanner reader; try & reader = new Scanner (new File l'elements.txt"); reader = new Scanner (new File l'elements.txt"); catch (FileNot Found Exception exc) { System. cut. printlm ("File not Found"), return; } Array List < Product = products = new Apray List < > (); String line; While (reader. nas Vext Line ()) { line = reader. next Line (); products. add (Product. convert (line)); } Store Store = new Store (products) Product max = store. search Max(); System. out. println (max.code l) + " "+ mox. name!) + " "+ max.name!) +" "+ max.name!]	Scanner reader; try { reader = new Scanner (new File "elements txt"); reader = new Scanner (new File "elements txt"); reader = new Scanner (new File "elements txt"); return; } ArrayList < froduct > products = new ArrayList < > (); String line; while (reader next Line()); line = reader next Line(); products add (froduct convert (line)); } Store Store = new Store (products) Product max = store . search Max(); System. out. printle (max.code L) + " + max.name!) + " + max.namber () + " store . print Mid ();	Scanner reader; tr; { reader = new Scanner (new File l'elements.txt"); reatch (FileNot Found Exception exc) { System.cut. printlm ("sile not Found"); return; } Array List < froduct = products = new Appay List <> (); String line; While (reader. nast Vext Line ()) { line = reader. next line (); products. add (Product. convert (line)); } Store Store = new Store (products) Product max = store. search Max(); System. out. printlm (max.code L) + " + max. maxere!) + " + max. mumber () + " + store. print Mid();	static void main (string) drys) Scanner reader; try { reader = new Scanner (new File l'elements.txt"); } catch (FileNot Found Exception exc) { System.cut. printlm ("sile not Found"), return; } Array List < froduct = products = new Array List => (); String line; While (reader. has Next Line ()) { line = reader. next Line (); products. add (Product. convert (line)); } Store Store = new Store (products) Product max = store. search Max(); System. out. printlm (max.code L) + " "+ max.name!) + " "+ max.name) () + " "+ max.name) store. print Mid();	Scanner reader; tr; { reader = new Scanner (new File l'elements txt"); reatch (FileNot Found Exception exc) { System.cut. printlm ("sile not Found"), return; } Array (list < froduct > products = new Aproy (list < > (); String line; While (reader. next line ()) { line = reader. next line (); products. add (Product. convert (line)); } Store Store = new Store (products) Product max = store. search Max(); System. out. println (max.code l) + " + max.name!) + " + max.number () + " + max.price(store. print Mid ();	ublic static void main(string) drys)? Scanner reader; try { reader = new Scanner (new File l'elements.txt'); } catch (File Not Found Exception exc) { System.cut. Arintla ("sile not Found"); return; } ArrayList < Product = new ArrayList < > (); String line; while (reader.has MextLine()) { line = reader.nextLine(); products.add(Broduct.convert (line)); } Store store = new Store (products) Product max = store. search Max(); System.cut. printla (max.codel) + " + max.name!) + " + max.number() + " + max.price()(); store. print Mid();	ublic static woid mains (string L) args)? Scanner reader; try { reader = new Scanner (new File elements.txt); } catch (File Not Found Exception exc) { System.cut. printlm ("sile not Found)), return; } Array List < Product = products = new Appay List <> (); String line; while (reader has Pext Line ()) { line = reader next Line (); products.add (Broduct. convert (line)); } Store store = new Store (products) Product max = store. search Max (); System. out. printlm (max.code l) + " + max.name) + " + max.namber () + " + max.price () (); store. print hid ();	ublic static void main(string. L.) drys)? Scanner reader; try { reader = new Scanner (new File l'elements.txt"); } catch (File Not Found Exception exc) { System.cat. printlm ("gile not Found"). return; } Array List < Product = products = new Appay List < > (); String line; while (reader. has Pext Line ()) { line = reader. next Line (); products. add (Product. convert (line)); } Store store = new Store (products) Product max = store. search Max(); System. out. printlm (man.code L) + " + max. name!) + " + max. nice (()); store. print Mid();	ublic static wid main (string L) drgs)? Scanner reader; try { reader = new Scanner (new File l'elements txt'); } catch (FileNot Found Exception exc) { System.cxt. Arintlm ("gile not Found"), return; } ArrayList < froduct = products = new ArrayList < > (); String line; while (reader has NextLine ()) { line = reader nextLine (); products.add (Product - convert (line)); } Store store = new Store (Aroducts) Product max = store. search Max(); System. out. printlm (max.code L) + " + max.name!) + " + max.number () + " + max.price()()); store. print Mid!);	Scanner reader; try { reader = new Scanner (new File l'elements.txt"), } catch (File Not Found Exception exc); System.cut. printlm ("gale not Found"), return; } Array (ist < froduct = products = new Appay List < > (); String line; While (reader.has Next Line()); I'me = reader.next Line(); products.add (Product.convert (line)); } Store store = new Store (products) Product max = store. search Max(); System.out. printlm (max.code L)+" "+ max.newhor()+""+ max.nexce(); store. print Mid();	Scanner reader; try { reader = new Scanner (new File "elements.txt"); reader = new Scanner (new File "elements.txt"); reader = new Scanner (new File "elements.txt"); return; } Array(ist < froduct > products = new Appay(ist < > (); String line; while (reader.has Wext(ine())); line = reader.next(); products.add(froduct.convert (line)); } Store store = new Store (products) Product max = store. search Max(); System.out.printle (max.code L) + " + max.name() + " + max.number() + " + max.price()); store.printle();	Scanner reader; try { reader = new Scanner (new File l'elements.txt"), } catch (File Not Found Exception exc) { System.ccit. printlm ("sile not Found"), return; } Array (ist < froduct = products = new Apray List < > (); String line; while (reader.has MextLine()) { line = reader.nextLine(); products.add (Product convert (line)); } Store store = new Store (products) Product max = store. search Max(); System. out. printlm (max.cade L) + "" + max.name!) + "" + max.namelor() + "" + max.price(); store. print Mid();