Manuale per il docente:

Linee guida attività “Attività Multilivello”.

1. Introduzione:

Questa attività propone un totale di 4 esercizi, di difficoltà crescente. Uno degli scopi di questa attività è quello di mettere in risalto schemi di programmazione comuni, in modo tale che quello che cambi siano poche righe di codice. Considerando il fatto che in parallelo a questo corso si sta svolgendo anche il corso di Java avanzato “Object Oriented Software Design”, è opportuno e interessante fare dei paralleli immediati con il linguaggio C e con il linguaggio Assembly.

1. Prerequisiti e Modelling:

Prima di proporre questa attività, è consigliabile prima aver introdotto e spiegato concetti come quello di:

1. Puntatore;
2. Puntatore a funzione;
3. Array a 2 dimensioni;
4. Strutture;
5. Array di Strutture;

Un buon esercizio preliminare può essere quello di provare a scrivere un programma in cui si fa notare che l’implementazione con il linguaggio Java è sostanzialmente identica. A tal proposito, un buon esercizio preliminare è il seguente:

Dato un array di interi, verificare, se esistono, due elementi che hanno lo stesso valore. (facendo notare che, sostanzialmente, lo stesso esempio implementato con il linguaggio Java è uguale).

1. Concetti richiamati:

In questo caso vengono proposti schemi di programmazione in comune tra Java e il linguaggio C (vedi primo esempio). In uno degli esempi presentati viene utilizzato un array in cui i suoi elementi sono puntatori. Nell'esempio proposto possiamo notare che tutti gli elementi dell'array sono puntatori. In Java, un array può essere visto come un vero e proprio oggetto. Quando, ad esempio, definiamo un'array di interi (int[]array), stiamo puntualizzando il fatto di aver introdotto un puntatore all'oggetto array. Quelli che nel linguaggio C vengono chiamati puntatori, in Java vegnono chiamati riferimenti. In effetti, tutti gli oggetti che definiamo in Java sono riferimenti, e sono molto simili ai puntatori usati nel linguaggio C.

1. Attività e Coaching:

Di seguito, ecco una serie di attività da proporre agli studenti:

1. Primo esempio:

Dato un array di interi, eseguire la ricerca del massimo e restituire tale valore (facendo notare che in Java l’implementazione è sostanzialmente identica).

1. Secondo esempio:

Ripetere la stessa operazione su un array di stringhe con un piccolo accorgimento: fare in modo di rispettare riga per riga le istruzioni, in modo tale che la differenza nei 2 blocchi di codice sia presente solo nell'istruzione if (cioè l’istruzione che confronta il massimo corrente e l’elemento in i-esima posizione).

1. Terzo esempio:

Provare a scrivere un programma che rappresenta un polinomio a coefficienti interi come un array di monomi (coefficiente, esponente) e calcola il grado del polinomio. Per la rappresentazione, valutare le alternative: array di strutture, array di puntatori a strutture, array di array di dimensione 2.

Coaching:

Tra le varie alternative proposte per la rappresentazione, la più consigliabile è quella rappresentata da array di strutture, in quanto è la struttura più adatta alla rappresentazione di un monomio. Definire quindi, per prima cosa, una struttura che rappresenta un monomio. Un array definito come sequenza di più strutture “monomio”, andrà quindi a costituire un polinomio su cui effettuare la verifica del suo grado massimo.

1. Quarto esempio:

Generalizzare, inserendo il parametro puntatore a funzione per effettuare il confronto tra coppie di elementi. Notare che per generalizzare, è necessario che tutti gli elementi dell’array siano dei puntatori.

Coaching:

In questo caso si richiede di andare a definire un puntatore a funzione. Definire quindi la funzione che andrà ad effettuare il confronto tra gli elementi. Una volta fatto ciò, definire un array di puntatori (andando quindi ad utilizzare una funzione di allocazione della memoria, come malloc) alla struttura definita nell’esercizio precedente. Il punto finale è, quindi, utilizzare il puntatore a funzione per scoprire il grado massimo del polinomio creato.