

## Soluzione esercizio [Quiz]

a. Definizione della classe base e delle sottoclassi:

```

/***** FILE: Quiz.java *****/
public class Quiz {
    private String text;
    private int score;

    public Quiz(String t, int s) {
        text = t;
        score = s;
    }

    public int getScore() { return score; }
    public boolean isCorrect(String answer) { return false; }

    public void print(int n) {
        System.out.println("Quiz "+n+" (score "+score+"");
        System.out.println(text);
    }
}

/***** FILE: QuizMulti.java *****/
public class QuizMulti extends Quiz {
    private String[] choice;
    private char answer;

    public QuizMulti(String t, int s, String[] ch, char a) {
        super(t, s);
        choice = ch;
        answer = a;
    }

    // ridefinisce il metodo della classe Quiz
    public boolean isCorrect(String a) {
        return (a.charAt(0) == answer);
    }

    // ridefinisce il metodo della classe Quiz
    public void print(int n) {
        super.print(n);
        for (int i = 0 ; i < choice.length ; i++)
            System.out.println(((char)('A'+i))+". "+choice[i]);
    }
}

/***** FILE: QuizNumber.java *****/
public class QuizNumber extends Quiz {
    private int[] answer;

    public QuizNumber(String t, int s, int[] a) {
        super(t, s);
        answer = a;
    }

    // ridefinisce il metodo della classe Quiz
    public boolean isCorrect(String a) {
        int ans = Integer.parseInt(a);
        for (int i = 0 ; i < answer.length ; i++)
            if (answer[i] == ans) return true;
        return false;
    }
}

```

```

}
/***** FILE: QuizWord.java *****/
public class QuizWord extends Quiz {
    private String[] answer;

    public QuizWord(String t, int s, String[] a) {
        super(t, s);
        answer = a;
    }
    // ridefinisce il metodo della classe Quiz
    public boolean isCorrect(String a) {
        for (int i = 0 ; i < answer.length ; i++)
            if (answer[i].equals(a)) return true;
        return false;
    }
}

```

b. /\*\*\*\*\* FILE: QuizTest.java \*\*\*\*\*/

```

import java.util.*;
import java.io.*;

public class QuizTest {
    // ritorna in un array i tokens della stringa line (i tokens sono separati
    static private String[] readTokens(String line) { // dal carattere '#'
        int nTokens = 1;
        for (int i = 0 ; i < line.length() ; i++) // calcola il numero di tokens
            if (line.charAt(i) == '#') nTokens++;
        String[] tokens = new String[nTokens]; // crea array per mantenere i tokens
        int k = 0, i = 0;
        for (int j = 1 ; j < line.length() ; j++) // leggi i tokens
            if (line.charAt(j) == '#') {
                tokens[k++] = line.substring(i, j);
                i = j + 1;
            }
        tokens[k] = line.substring(i, line.length());
        return tokens;
    }

    public static Quiz[] ReadQuizFile(String pathname) throws FileNotFoundException {
        Scanner fscan = new Scanner(new File(pathname)); // apri il file in lettura
        Quiz[] quizA = new Quiz[0]; // predisponi l'array dei quiz
        while (fscan.hasNextLine()) { // leggi tutte le linee del file
            String line = fscan.nextLine();
            int n = quizA.length; // crea un nuovo array
            Quiz[] q = new Quiz[n + 1]; // per mantenere un quiz in più
            for (int i = 0 ; i < n ; i++)
                q[i] = quizA[i];
            quizA = q;
            String[] tokens = readTokens(line); // leggi i tokens della linea
            char cat = tokens[0].charAt(0); // la categoria del quiz
            String text = tokens[1]; // il testo del quiz
            int score = Integer.parseInt(tokens[2]); // il punteggio del quiz
            switch (cat) {
                case 'M': { // quiz a scelta multipla
                    char ans = tokens[3].charAt(0);
                    String[] choice = new String[tokens.length - 4];
                    for (int i = 0 ; i < choice.length ; i++)
                        choice[i] = tokens[i + 4];
                    quizA[n] = new QuizMulti(text, score, choice, ans);
                    break;
                }
            }
        }
    }
}

```

```

    }
    case 'N': { // quiz a risposta numerica
        int[] ans = new int[tokens.length - 3];
        for (int i = 0 ; i < ans.length ; i++)
            ans[i] = Integer.parseInt(tokens[i + 3]);
        quizA[n] = new QuizNumber(text, score, ans);
        break;
    }
    case 'W': { // quiz a risposta testuale
        String[] ans = new String[tokens.length - 3];
        for (int i = 0 ; i < ans.length ; i++)
            ans[i] = tokens[i + 3];
        quizA[n] = new QuizWord(text, score, ans);
        break;
    }
    }
}
return quizA;
}
}
}

```

c. Definizione del metodo main() della classe QuizTest:

```

public static void main(String[] args) throws FileNotFoundException {
    Quiz[] quizA = ReadQuizFile(args[0]);
    Scanner ascan = new Scanner(new File(args[1]));
    int totalScore = 0;
    for (int i = 0 ; i < quizA.length ; i++) {
        quizA[i].print(i + 1);
        String ans = ascan.nextLine();
        System.out.print("Answer: "+ans);
        String a;
        if (quizA[i].isCorrect(ans)) {
            a = "CORRECT";
            totalScore += quizA[i].getScore();
        } else a = "NOT CORRECT";
        System.out.println(" "+a);
        System.out.println();
    }
    System.out.println("Total score: "+totalScore);
}
}

```

**Soluzione esercizio [Errori]** Ci sono due errori.

1. La linea `Value v = val[0].getValue();` provoca un errore in compilazione perché il metodo `getValue()` ritorna un riferimento a `Object` che non è un sottotipo di `Value`.
2. La linea `String s = (String)obj;` produce un errore in esecuzione perché la variabile `obj` conterrà il riferimento a un oggetto di tipo `Value` che non è un sottotipo di `String` (è lanciata l'eccezione `ClassCastException`).

**Soluzione esercizio [Insiemi\_di\_anagrammi]**

a. `public class Anagrammi {`

```

private String[] insieme;

public Anagrammi(String s) {
    insieme = new String[1];
    insieme[0] = s;
}

// metodo ausiliario che conta il numero di occorrenze del carattere
private int conta(String s, char c) { // c nella stringa s
    int occ = 0;
    for (int i = 0 ; i < s.length() ; i++)
        if (s.charAt(i) == c) occ++;
    return occ;
}

// metodo ausiliario che determina se le stringhe s1 e s2 sono anagrammi
private boolean anagram(String s1, String s2) {
    if (s1.length() != s2.length()) return false;
    for (int i = 0 ; i < s1.length() ; i++) {
        if (conta(s1, s1.charAt(i)) != conta(s2, s1.charAt(i)))
            return false;
        if (conta(s1, s2.charAt(i)) != conta(s2, s2.charAt(i)))
            return false;
    }
    return true;
}

public boolean aggiungi(String s) {
    int n = insieme.length;
    if (n > 0 && !anagram(insieme[0], s)) // controlla che s sia un anagramma
        return false; // delle stringhe nell'insieme
    for (int i = 0 ; i < n ; i++)
        if (insieme[i].equals(s)) // controlla che s sia diversa da
            return false; // tutte le stringhe nell'insieme
    String[] ins = new String[n + 1]; // crea un nuovo array per mantenere
    for (int i = 0 ; i < n ; i++) // una stringa in più
        ins[i] = insieme[i];
    ins[n] = s;
    insieme = ins;
    return true;
}
}

```

b. Definizione del metodo seleziona() della classe Anagrammi:

```

public void seleziona(String[] sA) {
    int n = insieme.length;
    boolean[] mark = new boolean[n]; // array che serve a marcare le
    for (int i = 0 ; i < n ; i++) // stringhe da selezionare
        mark[i] = false;
    for (int i = 0 ; i < sA.length ; i++)
        for (int j = 0 ; j < n ; j++)
            if (sA[i].equals(insieme[j])) // marca la stringa selezionata
                mark[j] = true;
    int nSelezionate = 0;
    for (int i = 0 ; i < n ; i++) // conta il numero di stringhe
        if (mark[i]) nSelezionate++; // selezionate
    String[] sel = new String[nSelezionate]; // crea un nuovo array per
    int k = 0; // mantenere l'insieme selezionato
    for (int i = 0 ; i < n ; i++) // copia nel nuovo array le
        if (mark[i]) sel[k++] = insieme[i]; // le stringhe selezionate
}

```

```
    insieme = sel;  
}
```