**http://www.numbeo.com/health-care/rankings\_by\_country.jsp**

**About Pollution Indices At This Website**

This section is based on surveys from visitors of this website. Questions for this surveys are similar to many similar scientific and government surveys.

Each entry in the survey is saved as the number in the range [-2, +2], with -2 having meaning of strongly negative and +2 meaning of strongly positive.

We filter surveys to eliminate potential spam, like people entering the large amount of data which are differentiating from the median value.

To present survey result, we use the scale [0, 100] for values since it is easier to read for users.

Most of our data are based on perceptions (opinions) from visitors of this website. For pollution section, we include some relevant data from World Health Organization and other institutions if we find it helpful. Please consult our [Terms of use](http://www.numbeo.com/common/terms_of_use.jsp) for details.

*Pollution Index* is an estimation of the overall pollution in the city. The biggest weight is given to air pollution, than to water pollution/accessibility, two main pollution factors. Small weight is given to other pollution types.

*Pollution Exp Scale* is using an exponential scale to show very high numbers for very polluted cities, and very low numbers for unpolluted cities. Therefore to calculate formula it uses the exponential function to calculate the index.

Actual formulas to calculate indices is a subject to change and at this moment, quite complex empirical formulas are used. Those formulas as written in Java programming language are as follows:

public void calculateIndex() {

//assumes air\_quality and other entries from user are in the range [-2, 2], where -2 means perceived as very low, and +2 means very high

//PollutionDbEntry.IS\_POLLUTION\_AIR\_QUALITY and similar are constant variables which are either -1 and 1; i.e. IS\_POLLUTION\_AIR\_QUALITY = -1.0

//These constant variables in PollutionDbEntry class are 1 for values which represent pollutions and -1 for values which represent opposite (purity, cleanliness)

index = new PollutionIndex();

double overall = 0.0;

overall += 7 \* getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_AIR\_QUALITY \* air\_quality);

overall += 2 \* getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_DRINKING\_WATER\_QUALITY\_ACCESSIBILITY \* drinking\_water\_quality\_accessibility);

overall += 2 \* getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_WATER\_POLLUTION \* water\_pollution);

overall += getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_GARBAGE\_DISPOSAL\_SATISFACTION \* garbage\_disposal\_satisfaction);

overall += getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_CLEAN\_AND\_TIDY \* clean\_and\_tidy);

overall += getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_NOISE\_AND\_LIGHT\_POLLUTION \* noise\_and\_light\_pollution);

overall += getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_GREEN\_AND\_PARKS\_QUALITY \* green\_and\_parks\_quality);

overall += 2 \* getIndexPartPreCalc(PollutionDbEntry.IS\_POLLUTION\_COMFORTABLE\_TO\_SPEND\_TIME \* comfortable\_to\_spend\_time);

double overallExpScale = 0.0;

overallExpScale += 7 \* getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_AIR\_QUALITY \* air\_quality);

overallExpScale += 2 \* getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_DRINKING\_WATER\_QUALITY\_ACCESSIBILITY \* drinking\_water\_quality\_accessibility);

overallExpScale += 2 \* getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_WATER\_POLLUTION \* water\_pollution);

overallExpScale += getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_GARBAGE\_DISPOSAL\_SATISFACTION \* garbage\_disposal\_satisfaction);

overallExpScale += getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_CLEAN\_AND\_TIDY \* clean\_and\_tidy);

overallExpScale += getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_NOISE\_AND\_LIGHT\_POLLUTION \* noise\_and\_light\_pollution);

overallExpScale += getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_GREEN\_AND\_PARKS\_QUALITY \* green\_and\_parks\_quality);

overallExpScale += 2 \* getIndexPartPreCalcExpScaleStandard(PollutionDbEntry.IS\_POLLUTION\_COMFORTABLE\_TO\_SPEND\_TIME \* comfortable\_to\_spend\_time);

index.main = overall / 14.5; //max 17

index.expScale = calcScaleStandardIndexFromSum(overallExpScale, 12);

}

protected double getIndexPartPreCalc(double internalValue) {

return (internalValue + 2) \* 25;

}

protected double getIndexPartPreCalcExpScaleStandard(double internalValue) {

return getIndexPartPreCalcExpScale(internalValue, Math.E);

}

protected double getIndexPartPreCalcExpScale(double internalValue, double exp) {

return Math.pow((internalValue + 2) \* 25, exp);

}

protected double calcScaleStandardIndexFromSum(double scaleSum, int elems) {

return Math.pow(scaleSum / elems, 1 / (Math.E \* 8.8 / 10));

}