General Requirements

Introduction

To complete the tasks, you can use any of the tools provided according to the Infrastructure List.

If there is not enough time to complete all the remaining tasks, you can skip some tasks and complete others. However, you are expected to provide the most complete work at the end of each session to facilitate the assessment of your work.

Rules

During the skill competition the following rules must be observed:

- Internet access is prohibited (except for one-time access during a session for a period not exceeding 15 minutes);
- any gadgets are prohibited (smartphone, tablet, smart watch etc);
- your own storage devices are prohibited (USB drives, hard drives etc);
- communication with other participants is prohibited;
- books, notes etc are prohibited;
- personal input devices (keyboard, mouse, trackball etc) are allowed; however, such devices must be wired and nonprogrammable, they must work without installation of additional drivers (compliance with these requirements is pre-checked by a technical expert);
- personal means of improving ergonomics (mouse pad, wrist rest etc) and talismans are allowed (they are also checked by a technical expert);
- in case of any abnormal situation related to software or hardware/peripheral devices, you must immediately interrupt the work and contact an expert.

Failure to follow these rules may result in your removal from the competition.

Application name

Please use appropriate names for your applications and files. For example, the name of a desktop application must include the customer-company name.

File structure

The project file structure should reflect the logic of your application. For example, there should be different directories for forms, custom visual components and entity classes.

Project structure

Each entity must be represented in the application by at least one separate class. Classes should be small and straightforward, they should perform one single function (Single Responsibility Principle).

Use different forms to work with different entities where appropriate.

Logical structure

The presentation logic (working with user I/O, forms and event handling) should not be mixed with business logic (constraints and requirements in the tasks) or database access logic (SQL queries, data storage or data extraction). Ideally, there should be three independent modules.

Style guide

The visual components must comply with the Style Guide provided as the resources for the tasks in the appropriate file. Please ensure that the requirements of all components are met in the following areas:

- colour scheme,
- logo placement,
- use of fonts,
- setting the application icon.

Layout and specifications

All system components must have a consistent look and feel in accordance with the style guide. Also, the following requirements must be met:

- layout and design: scalable layout is preferred; a limitation on the minimum window size should be applied; the ability to resize the window, where necessary, should be implemented; along with an increase in the size of the window, the size of the content part should increase (for example, a table with data from a database);
- grouping of elements (into logical categories);
- using appropriate controls (e.g., drop-downs to display values from the database);
- arrangement and alignment of elements (labels, input fields etc.);
- sequential element switch focus transition (by pressing TAB);
- the overall layout being consistent, clear and easy to use;
- a consistent interface that allows users to navigate between existing windows in the application (including in the opposite direction, for example, using the Back button);
- appropriate titles in each application window (there should be no default values of MainWindow, Form1 etc.).

User feedback

It is necessary to notify the user about errors or prohibited actions within the task, ask for confirmation before deleting, warn about irreversible operations, inform about the absence of search results etc. Message windows of the corresponding types (for example, Error, Warning, Information) should be displayed with an appropriate title and a pictogram. The text of the message should be useful and informative, it should provide complete information about the user's mistakes and the procedure for correcting them. You can also display visual tips for users as they enter data.

Error handling

Do not allow the user to enter incorrect values in the text fields for entities. For example, this check is necessary if the data type or the size of the field does not match the entered value. Notify the user of their mistakes.

Note the use of absolute and relative image paths. The application should work correctly, including when moving the folder with its executable file.

When an unexpected error occurs, the application should not crash.

Code style

The identifiers of variables, methods and classes must reflect their essence and/or purpose of their use, including the names of controls (for example, you cannot use the default values of Form1, button3 etc.).

Identifiers must conform to the Code Convention and the CamelCase (for C # and Java) and snake_case (for Python) styles.

Only one command per line is allowed.

Comments

Add comments for non-obvious code snippets.

Quality code is perceived as plain text. Do not add comments for obvious code snippets. Comments should only be present where additional clarification is required.

Use the type of comments with appropriate tags (for example, param, return(s), summary etc.) that will later allow you to generate XML documentation.

Assessment

Each task is evaluated by testing the required functionality that has been implemented. The requirements for the implemented system are very high; therefore, it is possible that tools for automated testing of the application will be used. In this regard, during development, it may be necessary to follow certain rules for naming and structuring your project.

Presentation of results

All results should be transferred to the customer by uploading the files to the provided GIT version control repository. The results are as follows:

- application source code (as a commit of the current project version but not archived),
- executable files,
- other graphic/text files.

The results of work within each session should be uploaded to a separate branch called 'Session_X', where X is the session number.

When evaluating your work, only the contents of the repository will be considered. When evaluating, only electronic notes are considered (readme.md). Handwritten notes will not be included in the assessment.

The project must contain a description in Markdown format (see the templates in README-Template.md). Also please fill in additional information about the project and how to launch the application in the readme.md file.

Please note that you will not have additional time after the end of the session to save your files, so be vigilant and upload the results of your work in a timely manner within the session.