Урок № \_\_\_\_\_\_\_\_\_\_\_\_Дата\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |
| --- | --- |
| Тема: | Складання та виконання алгоритмів з розгалуженнями для опрацювання величин |
| Мета: | * *освітня:*формувати вміння складати та виконувати алгоритми з розгалуженням для опрацювання величин в середовищі програмування Lazarus, виконувати програми опрацювання величин у цьому середовищі; * аналізувати результати виконання програм опрацювання величин; * *розвивальна*: розвивати аналітичне мислення, пам’ять та увагу; * *виховна*: виховувати інтерес до інформатики, зосередженість при виконанні завдань, свідоме ставлення до вибору майбутньої професії. |
| Тип уроку: | *Формування вмінь та навичок* |
| Обладнання: | *Комп'ютери, підручники, інструкції з техніки безпеки, середовище програмування Lazarus.* |

Хід уроку

1. Організаційний момент.
2. Відтворення і корекція опорних знань.

***Дати відповіді на запитання.***

* Поясніть поняття «виконавець алгоритму». Перерахуйте характеристики виконавця.
* Які основні способи опису алгоритмів?
* Як зображується і що означає базова структура «розгалуження»?

1. Мотивація навчальної діяльності.

Для створення Додатків, що реалізують складну алгоритмічну логіку, потрібні засоби керування управління ходом роботи програми: зміни порядку виконання операторів в залежності від різних умов, та ефективної організації роботи над фрагментами, що частоповторюються. Математично доведено, що будь-який алгоритм можна закодувати з допомогою:

* оператора ***присвоювання;***
* ***умовного*** оператора;
* оператора ***циклу.***

Навчимося це робити на сьогоднішньому уроці.

1. Застосування вмінь та навичок.

**1.Практичне завдання. Інструктаж з БЖД.**

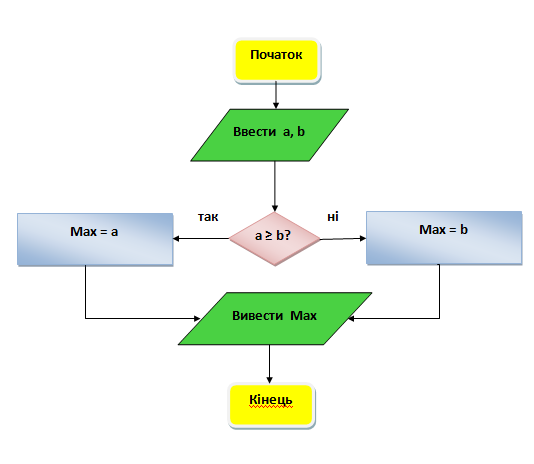
* 1. ***Виконання комплексу вправ для зняття зорової втоми.***

Учитель, враховуючи індивідуальні особливості учнів класу, самостійно визначає час і термін проведення комплексу вправ під час роботи (як правило, через 8-10 хвилин після початку роботи).

***б). Задача 1.*** Скласти програму пошуку більшого з двох даних чисел.

**Вказівки до виконання.**

* Побудуємо блок-схему для розв’язання даної задачі.
* Проаналізуємо разом з учнями дану блок- схему, звертаючи увагу на команду розгалуження та її виконання.
* Повторюємо правила запису умовного оператора. Можна використати демонстрацію



Умовний оператор дозволяє змінити порядок виконання операторів залежно від виконання деякої умови. Ось як він записується: if *Умова* then *Дія* else *Альтернатива* ; Слова if (якщо), then (тоді), else (інакше) - зарезервовані. *Дія* і else *Альтернатива* - це будь-які оператори або декілька операторів, взятих в логічні дужки begin/end, або виклик підпрограми. Якщо умова *істинна*, то виконується дія, якщо *хибна*, то виконується альтернатива.

Умовою є логічний вираз. У ньому порівнюються значення виразів (у тому числі також і логічних), виклик функцій, що повертають значення типа Boolean, і комбінування цих значень за допомогою логічних операцій:

|  |  |  |
| --- | --- | --- |
| **Знак операцій** | **Назва операції** | Логічні операції можуть комбінуватися за допомогою зв'язок: and (логічне І), or (логічне АБО) , xor (що виключає АБО). Для деяких типів даних є додаткові операції. Наприклад, для безлічі - оператор in, який перевіряє, чи входить деяке значення в безліч. Наприклад: X := [2, 4, 5, 7, 8, 10] Вираз 2 in X істинно (має значення true); Вираз 6 in X хибне (має значення false); |
| **=** | **Дорівнює** |
| **<>** | **Не дорівнює** |
| **>** | **Більше** |
| **<** | **Менше** |
| **>=** | **Більше або дорівнює** |
| **<=** | **Менше або дорівнює** |
| **not** | **Логічне Не - заперечення** |

Результат виконання операції логічного І дорівнює *true*, якщо обидва операнди дорівнюють *true*. Результат виконання операції логічного АБО дорівнює *true*, якщо хоч би один з операндів *true.* Результат виконання операції, що виключає АБО, дорівнює *true*, якщо операнди не дорівнюють один одному.

Наголошуємо на дві форми існування команди галуження:повної та скороченої.

* Відкриваємо середовище програмування Lazarus.
* Створюємо у власній папці папку **Проекти,** а в ній папку **«Задача 1»,** куди і зберігаємо **Project1,** натиснувшивказівку **Файл/Зберегти всі.**

Виділивши форму, змінимо її властивості у Інспекторі об’єктів на такі:

Caption= Більше з двох,

Color = ‘рожевий’.

За відображення тексту відповідають компоненти Label, Edit(рядок введення), Label(мітка) iButton(кнопка).

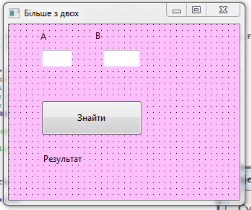
Вибираючи одним клацанням мишки або клавішею Tab потрібний компонент, в Інспекторі oб'єктів ми можемо дістати доступ до його властивостей.

Ці властивості отримують значення в той момент, коли компоненти потрапляють на форму.

За візуальне представлення відповідають для компонентів:

* *Label і Button* - властивість **Caption**,
* *Edit* - властивість  **Text**,
* Розмістимо на формі 2 поля для вставки *Edit*  , 3 мітки для напису *Label* і кнопку *Button*.
* Змінимо їх властивості в Інспекторі oб'єктів, виділяючи послідовно кожен об'єкт:
* *Edit* - властивість  **Text** очистити,
* *Button* - властивість **Caption**, написати **Знайти.**
* Потім - компонент ***Label*** (*мітка*). У властивості ***Font*** для наочності можна змінити розмір шрифту і колір. Властивість ***Caption*** також очистити.

Отримаємо форму такого вигляду:



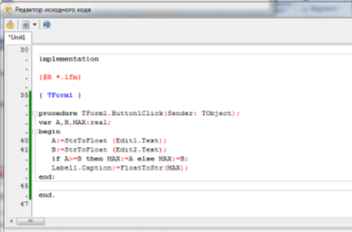
Двічі клацнемо по кнопці. Автоматично ми попадаємо у вікно модуля, де система Lazarus створила для нас заготовку обробника події *натискання на кнопку.*

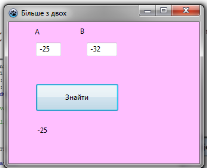
**Отже, після вибору кнопки Button1(подія Click) настає подія OnClick, яка полягає у виконанні команд процедури TForm1.Button1Click**.

Автоматично курсор зупиняється в тому місці, де необхідно ввести текст процедури**TForm1.Button1Click.**

Число, яке вводиться в поле, стає значенням його властивості **Text** і сприймається програмою як текст. Тому перші дві команди процедури є команди присвоювання, які призначені для переведення текстового представлення дійсних чиселу самі дійсні числа (для цього використана функція **StrToFloat()** і присвоювання цих чисел змінним ***х*** і ***у*** відповідно.

Далі записуються основні команди обчислення результату.

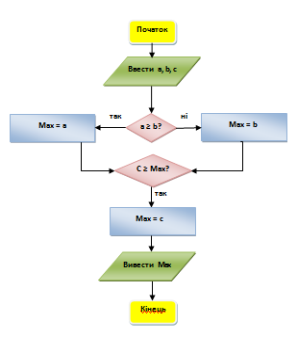
Остання команда переводить результат додавання (значення змінної) з числа в текст, використовуючи функцію **FloatToStr()** і присвоює цей текст значенню властивості **Caption** напису з іменем **Label1** для виведення результату.

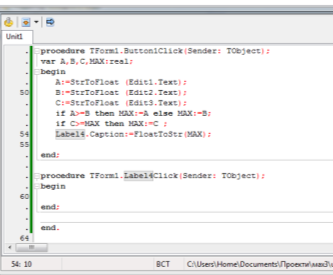


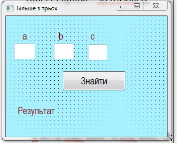
* Зберігаємо зміни у проекті.

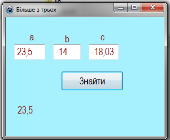
***в). Задача 2.*** Скласти програму пошуку більшого з трьох даних чисел.

**Вказівки до виконання.**

* + 1. Побудуємо блок-схему для розв’язання даної задачі.

* + 1. Проаналізувавши блок-схему, переходимо до створення Project2 за попередньою схемою.
    2. Отримаємо результат:





**V.Підсумок уроку.**

***Рефлексія.***

Наприклад, учням пропонується закінчити речення на зразок: "Для мене сьогодні важливим було...", "Сьогодні я дізнався про...", "Мені хотілося в майбутньому дізнатись про..., навчитись …".

* 1. **Перевірка, аналіз і оцінка робіт, виконаних під час уроку.**

VI. **Домашнє завдання**

* Опрацювати відповідний параграф підручника, конспект уроку.
* Повторити тему «Алгоритми з повторенням».