

Контрольные вопросы:

1. Интерпретатор - система последовательных операций для решения логич.-н задач.

Св-ва:

- Детерминированность (определенно)
- Результативность
- Массовость
- Дискретность
- Конечность

Виды

- Линейный
- Ветвящийся
- Циклический.

2. Компилирование - превращение исходного кода программы в машинный код.

Компиловка - включение переносимых ф-ций из сторонних библиотек, которые прописаны и используются в программе.

3. ДООП

Последовательной вазов команд, имеющей заданное, с которыми работает программа.

2) Функциональное программирование

Программа создается, как инструмент, который

поставляет определенную задачу и в итоге
получает конкретный результат.

2) Структурное программирование

Разновидность императивной парадигмы прог.
Фрагменты кода выполняются по командам, которые
применяются в определенном порядке
и последовательно меняют состояние
памяти.

3) Метапрограммирование

Программа, сама генерирует код новой прог.
или модифицирует свой.

3) Обобщенное программирование

Создание обобщенного представления для
массивов и функций.

3) Логическое программирование

Подвид декларативного. Основ. на базе
информации из заданных фактов и
логических правил, которые к ним можно
применить. При выполнении используются
правила формальной логики.

4. 1) В Visual Studio в меню **File** выберите пункт

Создать > Проект, чтобы открыть диалоговое окно

Создание проекта. Выберите шаблон **Консольное приложение** с тегом **C#**.

Применив фильтр **языка** и **платформы**. Выберите шаблон **Консольное приложение (.NET Core)** и нажмите **Далее**.

В поле имя проекта окна Настроить
новый проект ввести Yellowbird & создать

CTRL + F5 - Запуск без отладки

CTRL + F6 - Скрывающее окно документа

5. Нажмите SHIFT + F11 (или выберите
Оглазка > меню с выходов). Эта команда
возвращает вспомогательные приложения
(и переменные отладки) до возврата теку-
щей функции.

6. CLR

Основное задание CLR - автоматическое
обнаружение, загрузка и управление
типами .NET. В данном случае типами
управляет .NET, а не программист.

Также среди CLR работает с резе-
кционно-уровневыми деталями - управление
памятью, обработка потоков, выполне-
ние разных проверок, связанный с бэкоп.

CTS

Общая система типов (Common Type
System) или система CTS предоставляет
полное описание всех возможных
типов данных и программистов инструм-
ментов, которые поддерживаются исполн. сред.

7 bool: (хранит True или False)

byte (целое число от 0 до 255 занимает
1 байт.)

sbyte (целое число от -128 до 127, 1 байт)

short (целое число от -32768 до 32767
занимает 2 байта)

short (целое число от 0 до 65535) 2 байта
int (целое число от -2147483648 до 2147483647) 4 байта
long (целое число от 0 до 18446744073709551615) 8 байт

float (число с точкой) 4 байта

double (число с точкой) 8 байт

decimal (число с точкой) 16 байт

char (одиночный символ) 1 байт

string (набор символов)

8. Переменные - именованная область памяти, адрес которой можно использовать, для осуществления доступа к данным.

Переменные объявляются с помощью оператора следующей формы:

Тип имя - переменной:

где тип - это тип данных, которого содержится в переменной, а имя - переменной - это имя.

Константы объявляются:

const тип имя - константа = значение;

const - ключевое слово, которое определяет, что именно есть константа.

9. var используется для объявления переменных.
Это предопределенный способ.

10.

Сжимающие и расширяющие преобразования
сжатие (padding) и расширение (widening). Расширяющие расширяют размер объекта в памяти, сжимающие сужают.

Явное и неявное преобразование
Неявное - компилятор за нас выполняет преобразование данных.

Явное - мы сами должны указать преобразование (например, `(int)`).

11. Меращие инкремента (увеличивает на 1), а декременты (уменьшает на 1).

12. Часто используют метод `ToString()` красивее объект. Если метод `ToString()` вызывается без параметров, то использует формат вывода по умолчанию.