|  |  |
| --- | --- |
| **Unity. Начальный уровень. Контрольный урок № 1** | |
| **Ученик:** Вадим | **Дата:07.06.2022** |
| **Ссылка на профиль: https://hwschool.bitrix24.ru/crm/contact/details/8811/** | |

|  |  |  |
| --- | --- | --- |
| Теория (устный опрос, 1-2 вопроса из раздела, в случае затруднения, можно задать еще вопрос из раздела) | | Отметка  +/- |
| **Тема 1 Переменные и типы данных** | |  |
| Что такое переменные? Для чего они нужны? | Ответит подходит под шаблон: “Это что-то, что можно как-то назвать и положить туда какие-то значения, а затем изменять их” | + |
| Какие бывают типы данных? Например? | float – дробное число, int – целое число, string – текст, char – символ, bool - булево | + |
| Как увеличить значение переменной на 1? | *ИмяПеременной=ИмяПеременной+1;*  либо  *ИмяПеременной+=1;*  либо  *ИмяПеременной++;* | + |
| **Тема 2 Условия** | | |
| Что такое условие? Приведи пример. | Если А больше Б… (или любой другой пример неравенств или логических выражений) | + |
| Как можно проверить сразу несколько условий? | При помощи операторов && (и), || (или) | + |
| Какой оператор мы используем, когда хотим выполнять различные действия в зависимости от находящегося в какой-либо переменной значения? | Оператор выбора switch | + |
| Как записывается тернарная операция? | *Условие ? Возвращаемый результат, если условие истинно : Возвращаемый результат, если условие ложно* | + |
| **Тема 3 Циклы** | | |
| Какие бывают циклы? | While, for, foreach | + |
| Когда мы обычно используем цикл while? | Когда не знаем заранее, сколько раз будем повторять цикл | + |
| Когда мы обычно используем цикл for? | Когда знаем заранее, сколько раз будем повторять цикл | + |
| Когда мы обычно используем цикл foreach? | Когда нужно что-то сделать для каждого элемента какой-то коллекции данных | + |
| **Тема 4** **Функции** | |  |
| Как вызвать функцию? | Написать её имя и скобочки | + |
| Что означает слово void в определении функции? | Что функция ничего не будет возвращать | + |
| Что делает оператор return? | Останавливает работу функции и возвращает какое-то значение на место вызова функции | + |
| Чем отличается простой return от yield return? | Простой return останавливает работу функции (и она начнет работать сначала при следующем вызове), а yield return ставит ее на паузу до следующего вызова (и функция продолжит работать со следующей строки кода после yield return) | + |
| **Тема 5 Коллекции данных** | | |
| Какие ты знаешь коллекции данных? | Массивы, списки, перечисления, словари, стеки, очереди | + |
| В чем главное отличие массивов от списков? | Массивы имеют фиксированный размер (нельзя увеличивать/уменьшать), а в списки можно добавлять/удалять элементы | + |
| Что можно хранить в перечислениях? Приведи пример набора данных, который имеет смысл забить в перечисление | Логически связанные константы (постоянные данные). Например, перечень названий дней недели, или перечень различных эффектов заклинаний (заморозка/воспламенение/замедление/ослепление/…) | + |
| Для чего используются словари? | Для хранения данных в парах ключ – значение. | + |
| Чем отличаются друг от друга стеки и очереди? | Из стека можно легко вытащить последнее вошедшее в него значение, а из очереди легко извлечь первое вошедшее в нее значение. | + |
| **Тема 6 Классы** | | |
| Приведи три примера объектов из видеоигр, для которых следует создавать отдельный класс. | Персонаж, корабль, игровое поле, автомобиль, оружие, мебель, здание… | + |
| Чем поле класса отличается от свойства класса? | Поле – это просто переменная, объявленная в классе, которая хранит данные. Свойство же управляет доступом к данным, хранящимся в каком-либо поле. | + |
| Чем отличаются друг от друга динамические и статические методы классов? | Динамические методы можно вызывать только из какого-то конкретного экземпляра класса (объекта), а статические методы можно вызывать напрямую из самого класса. | + |
| **Итого** | |  |

|  |  |  |
| --- | --- | --- |
| Практика (выполнение практических заданий) | |  |
| Переименовал классы и файлы Queue и QueueItem в Stack и StackItem | Выполнено/не выполнено | Выполнено |
| Переименовал метод Enqueue() в Push(), а Dequeue() в Pop() | Выполнено/не выполнено | Выполнено |
| В StackItem добавил свойство Previous (опционально: поставил его в конструкторе вместо Next) | Выполнено/не выполнено | Выполнено |
| В методе Push() установил старый хвост в свойство Previous нового элемента стека | Выполнено/не выполнено | Выполнено |
| В методе Pop() положил в результат значение хвоста, и передвинул хвост в Previous хвоста | Выполнено/не выполнено | Выполнено |
| В методе Pop() устанавливает голову в null, если хвост стал равен null | Выполнено/не выполнено | Выполнено |
| В методе Pop() если хвост не стал null, то устанавливает tail.Next в null | Выполнено/не выполнено | Выполнено |
| **Итого** | |  |

**Оценка отношения ученика к обучению (по шкале от 1 до 5):**

Интерес –5

Мотивации –5

Вовлеченность –5

Успеваемость –5

***Во время занятий ученик проявил себя как (выбрать 2-3):***

* Внимательный
* Усидчивый
* Любознательный
* Трудолюбивый
* Креативный, новатор,
* Изобретатель
* Свой вариант

***Качества, требующие дальнейшего внимания и развития (выбрать 2-3):***

* Внимательность,
* Усидчивость
* Трудолюбие
* Выполнение домашних заданий
* Математические расчеты
* Умение презентовать-рассказать и логически объяснить
* Творчество
* Инициативность

***Темы, особенно интересные ученику:***

* Программирование
* Рисование
* Сам игровой процесс

***Рекомендации от преподавателя:***

* Продолжить курс
* Сменить курс
* Увеличить количество самостоятельных занятий до….
* Уменьшить количество самостоятельных занятий до….