Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

Отчёт защищён с оценкой\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Преподаватель Троицкий В.С.

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2020 г.

Отчёт

по лабораторной работе № 7

«Ассоциация»

по дисциплине «Программирование - 3»

Студент группы ПИ 92 В.М. Шульпов

Преподаватель доцент, к.т. н. Троицкий В.С.

Барнаул 2020

ЗАДАЧА

**C++**

1) Продемонстрировать возврат значения из метода через указатель (\*) и через ссылку (&);

2) Продемонстрировать разумное использование this;

3) Создать дружественную функцию и продемонстрировать ее использование;

4) Выполнить перегрузку операторов '+', '++' (два варианта, префиксный и постфиксный). Продемонстрировать в main;

5) Заменить массивы char на std::string, продемонстрировать работу с этим классом;

**Java**

6) Продемонстрировать работу с массивом объектов;

7) Продемонстрировать возврат целочисленного значения из метода через вспомогательный класс;

8) Продемонстрировать разумное использование this;

9) Продемонстрировать обработку строк (String);

**C#**

+10) Для полей добавить свойства и продемонстрировать работу с ними;

+11) Cоздать массив объектов и продемонстрировать работу с ним;

12) В отдельной ветке проекта заменить класс (class) на структуру (struct). Продемонстрировать различие между присваиванием объектов класса и структуры;

+14) Продемонстрировать возврат значения через параметр out и через параметр ref. Показать различие этих механизмов;

+16) Продемонстрировать перегрузку операторов '+', '++'.

+17) Продемонстрировать обработку строк (string);

!) Использование Git обязательно.

ТЕОРИЯ

**Когда использовать возврат по адресу**:

  при возврате динамически выделенной памяти;

  при возврате аргументов функции, которые были переданы по адресу.

**Когда не использовать возврат по адресу**:

  при возврате переменных, которые были объявлены внутри функции (используйте возврат по значению);

  при возврате большой структуры или класса, который был передан по ссылке (используйте возврат по ссылке).

**Когда использовать возврат по ссылке**:

  при возврате ссылки-параметра;

  при возврате элемента массива, который был передан в функцию;

  при возврате большой структуры или класса, который не уничтожается в конце функции (например, тот, который был передан в функцию).

**Когда не использовать возврат по ссылке**:

  при возврате переменных, которые были объявлены внутри функции (используйте возврат по значению);

  при возврате стандартного массива или значения указателя (используйте возврат по адресу).

2)

|  |  |  |
| --- | --- | --- |
| **Вопрос** | **Структура** | **Класс** |
| И какого же типа экземпляр объекта? | Структура значимый (value) тип | Класс ссылочный (reference) тип |
| А где “живут” экземпляры этих объектов? | Экземпляры структуры называют *значениями* и “живут” они в стеке (stack). | Экземпляры классов называют *объектами* и “живут” они в куче (heap). |
| Можно ли создать конструктор по умолчанию? | Нет | Да |
| Если создается свой конструктор будет ли компилятор генерировать конструктор по умолчанию? | Да | Нет |
| Если в своём конструкторе не будут инициализированы некоторые поля, будут ли они автоматически инициализированы компилятором? | Нет | Да |
| Разрешается ли инициализировать переменные в месте их объявления? | Нет | Да |

КОД ПРОГРАММЫ

ТЕСТЫ