|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| Предмет: | Высокоуровневое программирование |
| Подготовил студент гр. ИУК4-12Б: | Кузнецов И.П |

**Тема №1: Алгоритм и его свойства**

**Алгоритм** - это конечная последовательность шагов, каждый из которых

выполняется за конечное время и приводит к решению конкретной задачи. В

языке программирования С++, алгоритмы широко используются для обработки данных, поиска решений и других задач. Важными свойствами алгоритма являются корректность, эффективность и понятность.

**Основные свойства алгоритма:**

1. **Корректность("Correctness"):**

• Алгоритм считается корректным, если он решает поставленную задачу для

всех

возможных входных данных.

• Корректность означает, что алгоритм ведет себя так, как ожидается, и даёт

правильные результаты.

1. **Определённость ("Definiteness"):**

• Каждый шагалгоритма должен быть однозначно определен и понятен. Нет

неоднозначности в том, что нужно делать.

1. **Входные данные ("Input'):**

• Алгоритм должен иметь определенные входные данные. Это могут быть

параметры функции или данные, с которыми алгоритм работает.

1. **Выходные данные ("Output"):**

• Алгоритм должен производить определенные выходные данные, связанные

входными данными и решением задачи.

1. **Конечность ("Finiteness"):**

• Алгоритм должен завершить выполнение за конечное время. Он не должен

зацикливаться или выполняться бесконечно.

1. **Эффективность ("Efficiency"):**

• Алгоритм считается эффективным, если он использует ресурсы (время и/или

память) эффективно, и его производительность удовлетворительна для данной

задачи.

1. **Понятность ("Clarity"):**

• Алгоритм должен быть понятным и легко читаемым. Чем проще

структура, тем легче его поддерживать и модифицировать.

и яснее его

1. **Универсальность ("Generality"):**

• Алгоритм может быть применен к различным входным данным или

использован для решения более общих классов задач.

1. **Модульность ("Modularity"):**

• Алгоритм может быть разбит на отдельные блоки или модули, что упрощает

понимание и поддержку кода.

1. **Оптимальность ("Optimality"):**

• Алгоритм может быть оптимизирован с целью улучшения

производительности, но оптимизация не должна приводить к ухудшению

читаемости или понимания кода.

Название документа: ВП\_1\_КузнецовИП.docx