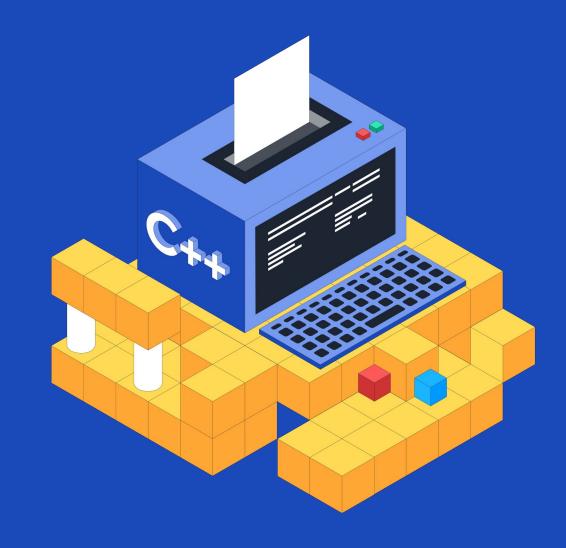




ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ C++

Материалы подготовлены отделом методической разработки

Средний уровень









Жадные алгоритмы









Жадный алгоритм

Жадный алгоритм называется так потому что старается сделать наиболее выгодный выбор на каждом шаге. Это не всегда приводит к лучшему результату, зато жадный алгоритм довольно легко реализовать.

Кроме того, жадные алгоритмы работают быстро, поэтому их использование оправдано, когда нужно получить приемлемый (пусть и не лучший) результат за короткое время.







Реализация

Реализацию жадного алгоритма можно рассмотреть на примере задачи с разменом: есть сумма, которую нужно выдать наименьшим количеством купюр разного номинала.

Для этой задачи номиналы купюр удобнее всего хранить в отсортированном по возрастанию списке. Тогда весь алгоритм займет несколько шагов:

- 1. Берем последнее число массива, сравниваем его с суммой.
- 2. Если число меньше или равно сумме, вычитаем его из суммы. Иначе берем предыдущее число.
- з. Повторяем шаги 1-2, пока сумма не будет выдана полностью или не дойдем до первого элемента массива.





Реализация

Реализация алгоритма, **summ** - сумма, которую надо выдать, **banknotes** - массив с номиналами купюр, **i** - последний индекс,

```
count - количество выданных банкнот:
while (summ > 0) {
     if (summ >= banknotes[i]) {
           summ -= banknotes;
           count++;
     else {
           i--;
           if (i < 0) {
                 count = -1;
                 break;
```







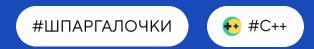
Структуры

Структура - это способ создать пользовательский тип данных, не используя классы. Структуры удобно использовать, когда нужно объединить в одну сущность несколько переменных, но методы не нужны.



Чтобы создать структуру, нужно использовать ключевое слово **struct**, а после него указать имя структуры. Поля в структурах по умолчанию публичные, поэтому модификаторы можно не использовать.

};





Структуры

Пример создания структуры, описывающей человека:

struct person {

string name;

int age;

Создание переменной типа **person** и доступ к полям:

```
person person1;
person1.age = 15;
person1.name = "Вася";
cout << "Тебя зовут " << person1.name << " и тебе " << person1.age << " лет" << endl;
```