

Т. Г. Алейникова,

доцент кафедры информатики и информационных технологий
Витебского государственного университета им. П. М. Машерова,

кандидат физико-математических наук, доцент

А. И. Шербаф,

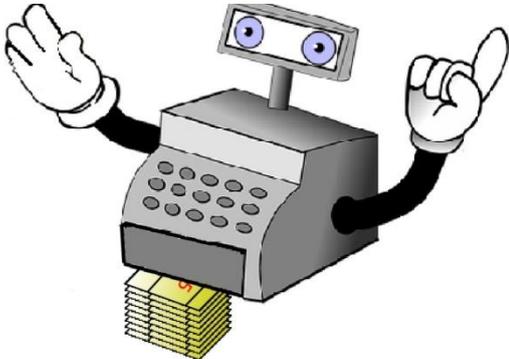
доцент кафедры информатики и методики преподавания информатики
Белорусского государственного педагогического университета

им. Максима Танка,

кандидат физико-математических наук, доцент

Рассмотрим более сложную задачу, которая была предложена учащимся 5–6 классов – участникам Витебской областной олимпиады по программированию в среде Scratch в 2017 году.

Задача «Банкомат»

 <p>Рисунок 6. Результат выполнения программы «Банкомат»</p>	<p>В банкомате имеется достаточное количество купюр номиналами 50, 20, 10 и 5 (рис. 6). Пользователь по запросу вводит сумму (не больше 500) и получает ее минимальным количеством купюр. Изображение купюр появляется на сцене, пользователю выдается сообщение. Необходимо предусмотреть обработку некорректного ввода запрашиваемой суммы.</p>
---	---

Для того, чтобы облегчить **понимание** задачи, используется прием «знакомая жизненная ситуация»: некоторую сумму надо разменять

имеющимися в наличии мелкими купюрами или монетами. Необходимо также обратить внимание на некоторые особенности постановки задачи по сравнению с реальной ситуацией: сумма должна быть предоставлена наименьшим количеством купюр, а также имеются ограничения для суммы (не более 500 – для удобства визуального отображения выдачи купюр и кратности значению минимальной купюры, чтобы размен был возможен).

Для выработки **плана** решения и использования известного в программировании "жадного алгоритма", целесообразно рассмотреть несколько частных случаев. Поясним, жадный алгоритм – это алгоритм, который на каждом шаге делает локально наилучший выбор в надежде, что итоговое решение будет оптимальным. Например, $30 = 20+10$, $95 = 50+2*20+5$, $335 = 6*50+20+10+5$. Отметим, что размен осуществляется сначала более крупными, а затем все более мелкими купюрами. Это может привести школьника на мысль хранить значения номиналов купюр в списке, отсортированном в порядке убывания (рис. 7).

купюры	
1	50
2	20
3	10
4	5
+	длина 4 =

Рисунок 7. Список с номиналами купюр

Далее следует этап **анализа**, в ходе которого сложная задача разделяется на более простые, что служит формированию понятия **декомпозиции**, как важнейшей составляющей ВМ. Представление анимированного размена в среде Scratch предусматривает участие двух исполнителей (не считая сцены) – спрайтов «Банкомат» и «Купюра». Для изображения банкомата можно воспользоваться встроенным графическим редактором или загрузить готовое изображение из Интернета. После непростого мысленного **обобщения** школьник приходит к заключению, что купюра с разными номиналами – это один объект в программе. Очевидно, чтобы делать подобные выводы школьник должен обладать определенным опытом и знать основы объектно-ориентированного подхода в программировании.

Переходим к разработке **алгоритма**. Вначале нужно подготовить сцену (придать ей стартовый внешний вид, удалить последствия предыдущего запуска). Это можно сделать непосредственно в коде сцены, но мы вставили эту процедуру в код спрайта «Банкомат». Далее следует диалог с пользователем и обработка «корректности» введенного значения суммы. Для размена введенной купюры вычисляется количество купюр каждого номинала поочередно от большего к меньшему, пока сумма не станет равна 0. Здесь будут задействованы **вычислительные навыки**: операция целочисленного деления (в среде Scratch – встроенная функция «целое меньше») и остаток от деления (операция «модуль»). Количество купюр (переменная n) определяется как результат целочисленного деления оставшейся суммы на значение номинала. На рис. 8 представлен код основной процедуры алгоритма «разменять i купюрой» (переменная i – номер купюры в списке).

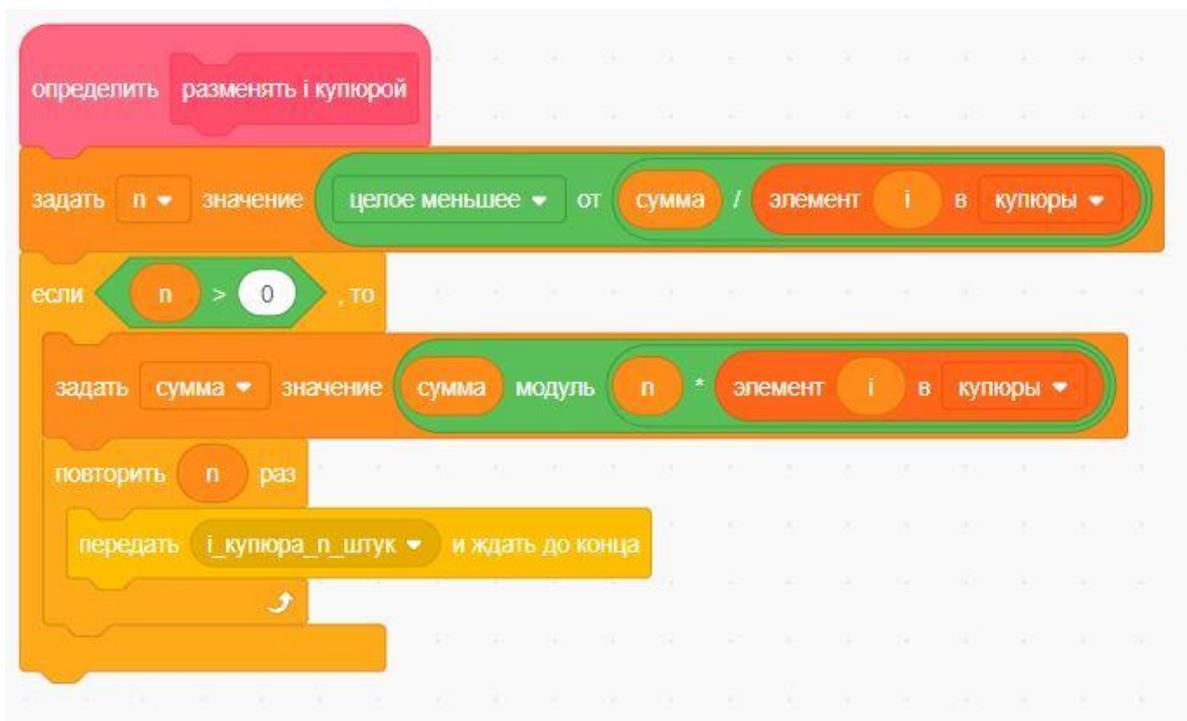


Рисунок 8. Процедура размена i -ой купюрой из списка

Если i -я купюра нужна для размена ($n > 0$), то вычисляется оставшаяся для размена сумма, посредством передачи сообщения управление передается спрайту «Купюра» для создания необходимого количества изображений

купюр. В процедуре «начать» можно выделить подзадачу «информирование пользователя об имеющихся номиналах» и оформить ее как вложенную процедуру «перечислить купюры» (рис. 9).

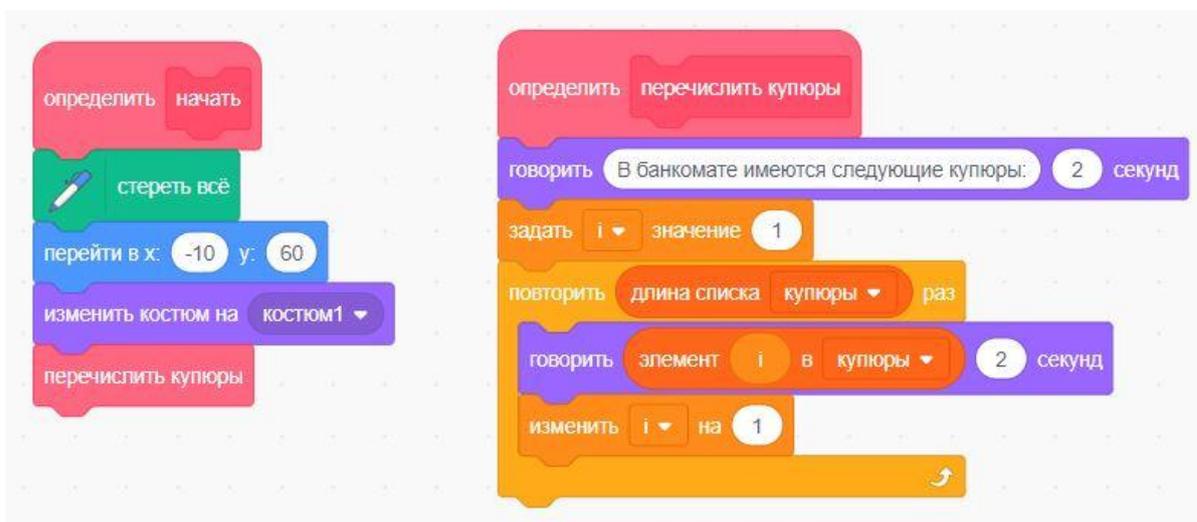


Рисунок 9. Стартовые процедуры спрайта «Банкомат».

Приведем основной код «Банкомата», объединяющий составные части алгоритма в единое целое (рис. 10).

Конструкция «если-то-иначе» позволяет проверить правильность введенных пользователем данных. Сложное условие проверки «корректности» введенной пользователем суммы рассчитано не только на очевидное ограничение (от 0 до 500), но и на возможность размена имеющимися в банкомате купюрами. Кратность суммы значению наименьшего номинала является условием возможности размена. Поскольку самый маленький номинал хранится в списке последним, то его порядковый номер равен длине списка, его значение используется в операции «модуль» (остаток от деления). Отметим, что сначала в условии можно включить конкретное число (5) и лишь на заключительном этапе перейти к обобщению произвольного списка номиналов.

Сообщение «Готово» передается для того, чтобы завершить сценарий анимации и спрятать спрайт с купюрами.



Рисунок 10. Основной код спрайта «Банкомат».

Алгоритм спрайта «Купюра» фактически состоит из обработчиков сообщений « i _купюра_n_штук» и «Готово» (рис. 11). Костюмы спрайта соответствуют номерам купюр в списке номиналов. Это позволяет использовать в имени костюма порядковый номер i (используется операция соединения строк «объединить»). Поворот и перемещение спрайта подобраны таким образом, чтобы печаталась пачка из n купюр.

На каждом этапе решение проверяется на практике. В процессе конструирования кода проводится отладка отдельных фрагментов, **оценивается** их работоспособность, устраняются ошибки. Заключительным этапом решения является итоговое тестирование. Следует подобрать тестовые значения суммы для крайних случаев (используется все купюры или только одна). Анимация решения также накладывает свои требования, поэтому в данной задаче сумма ограничивается 500, чтобы на размещение стопки купюр на сцене хватало места.

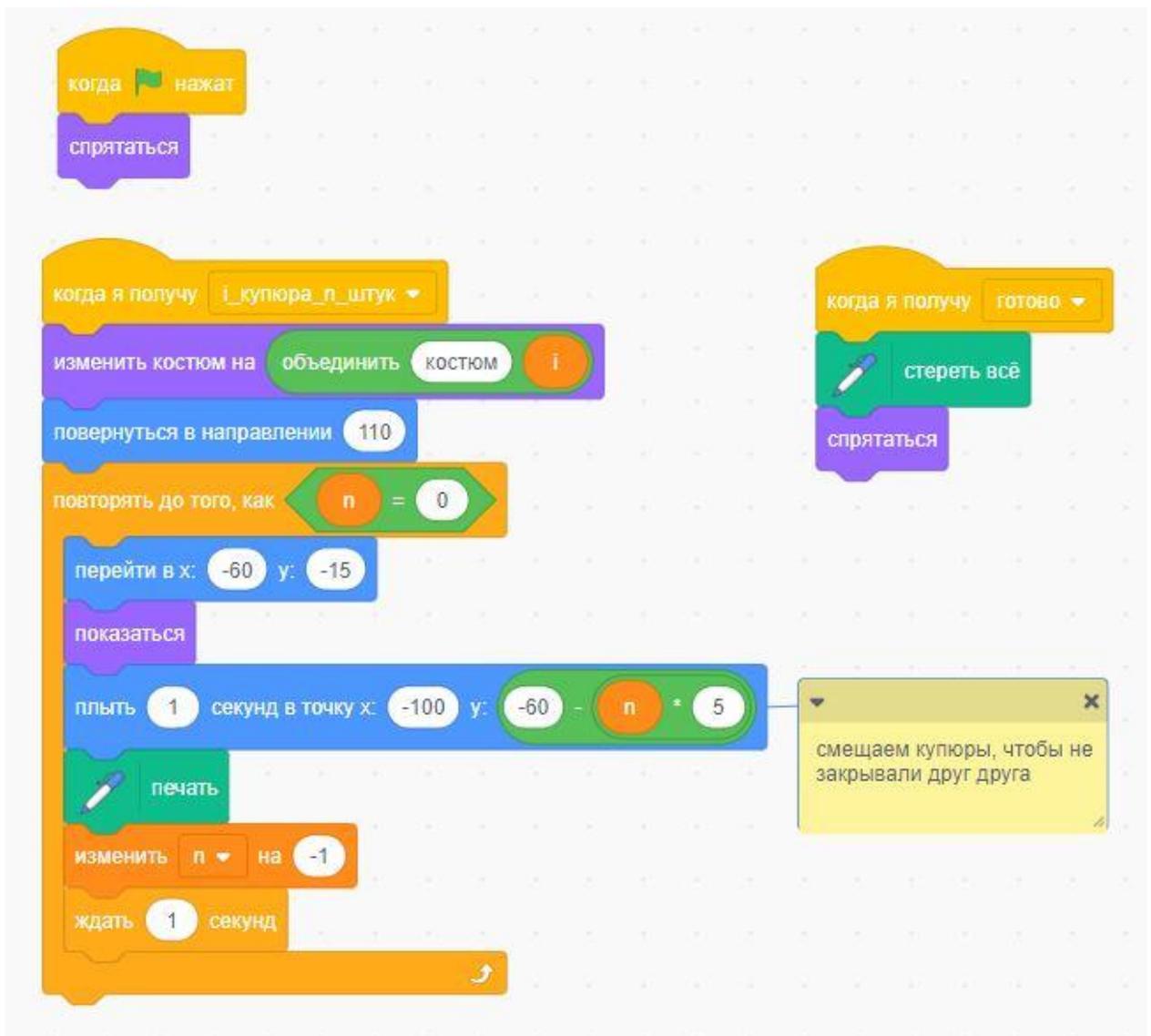


Рисунок 11. Код спрайта «Купюра».

Можно развить эту задачу, например, расширить список номиналов купюр или, учитывая реальные возможности банкоматов предусмотреть ограниченное количество купюр каждого номинала.