

## **Автоматический задачник по программированию и его возможности**

Проф., к.т.н. Бондарев В. М., Черепанова Ю. Ю.,  
Харьковский национальный университет радиоэлектроники

В процессе обучения будущего программиста центральное место занимает решение задач. Задача представляет собой словесную спецификацию некоторой программы или ее части. Чтобы решить задачу, будущий программист (далее студент) должен написать программу, которая данной спецификации полностью удовлетворяет. Проверку соответствия спецификации можно делать по-разному. Традиционно это делает преподаватель – визуально, если программа маленькая, или «тестируя» ее на компьютере. Слово «тестируя» взято в кавычки, т.к. преподаватель не тестировщик, и качественной проверки не устроит, хотя бы, по недостатку времени. Вполне очевидно, что таким способом преподаватель сможет проверить относительно небольшое количество задач. Задачи же, которые не проверяют, студенты решать не станут (не потому, что не сознают пользы от их решения, просто есть более срочные дела), и решение задач отодвигается с центрального места в обучении на дальнюю периферию. Как следствие, заметный процент дипломированных программистов программировать не умеет и вынужден еще раз искать свое место в жизни.

В то же время относительно несложно создать сервис, который получает исходный код программы, компилирует и выполняет ее, возвращая результат выполнения. Каким же образом подобный сервис может помочь в проверке решения задач студентами?

Исходный код, отправляемый сервису, должен быть законченной программой, состоящей из авторского решения некоторой задачи и серии модульных тестов этого решения (назовем эту комбинацию сборкой). Сборку преподаватель готовит заранее и сохраняет вместе с условием задачи. Когда студент решает задачу, его решение занимает в сборке место авторского и измененная таким образом сборка отправляется упомянутому выше сервису. Результат компиляции и выполнения сборки рассматривается как результат проверки студенческого решения. Это может быть сообщение об ошибках, найденных при компиляции (задача не решена), сообщение о провале некоторых тестов (задача не решена), сообщение об успешном прохождении все тестов (задача решена).

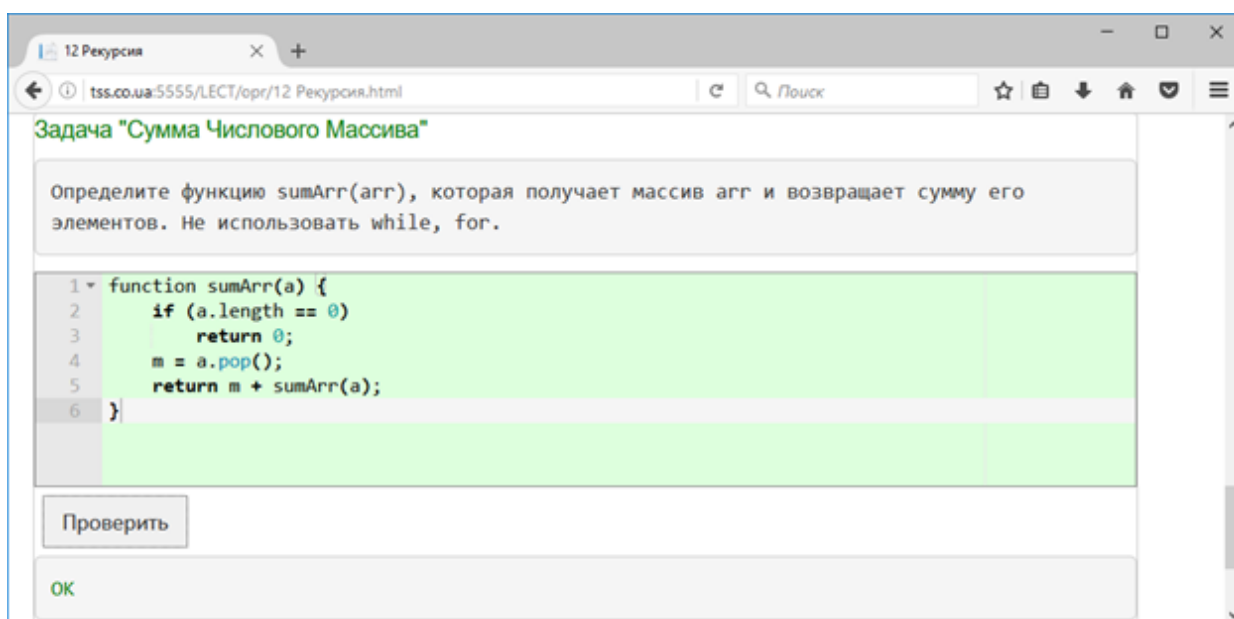
Условие задачи, плюс сборка, плюс идентификатор задачи составляют собственно задачу - единицу хранения в базе данных, которую мы назовем автоматическим задачником.

Как показала практика, наличие автоматического задачника вместе со службой компиляции и выполнения сборок радикально меняет процесс начального обучения программистов и заметно улучшает его результат.

Теперь от преподавателя не требуются сверхусилия по проверке студенческих каракулей, все задачи решаются студентами на компьютере и немедленно проверяются.

Конечно, необходимо позаботиться о наполнении задачника, но если задача попадет в задачник однажды, решена она будет сотни и даже тысячи раз. К тому же преподаватели могут наполнять задачник кооперативно.

Подавляющее большинство задач студенты решают самостоятельно, преподаватель должен лишь опубликовать их в сети. Он может делать это поэтапно, по мере подачи материала на лекциях, или группируя задачи иным способом, например, в рамках лабораторных работ. Не стоит задавать студентам все задачи сразу, лучше это делать порциями, обязательно включая в порцию и легкие, и средние задачи. Сложные задачи тоже нужны, но их стоит предлагать тем, кто справляется с легкими и средними.



Когда проверка решений автоматизирована, нетрудно организовать статистику и учитывать, кто сколько задач решил и каким образом. Преподаватель может наблюдать за процессом, анализировать, но, очевидно, не должен оценивать успеваемость студентов по этим данным.

Для оценки знаний (для мотивации студентов) необходимо использовать связку задачник-служба в ином режиме – в режиме экзамена. На «экзамене» студенты решают задачи за ограниченное время и в присутствии преподавателя, который должен обеспечить аутентификацию решающих и следить, чтобы они работали самостоятельно. Нельзя сказать, что в эпоху интернета и мобильных устройств это очень просто, но проблема решается и дела обстоят, по крайней мере, не хуже, чем на традиционных экзаменах. Заметим, что свою мотивационную функцию экзамены выполняют, если проводить их регулярно, хотя бы раз в две недели. В этом случае студентам становится ясно, для чего нужно решать задачи самостоятельно.

Описанная выше система в различных версиях эксплуатируется авторами уже третий год и показала свою полезность и, как теперь им кажется, незаменимость. Заметно повысилась вовлеченность студентов в процесс обучения, уверенность в объективности оценок, а главное, умение программировать.