



Модуль 04 - Piscine Java

JAR

Резюме: Сегодня вы научитесь создавать библиотечные архивы и использовать внешние библиотеки

Содержание

I	Предислови е	2
II	Инструкции	3
III	Упражнение Пакеты 00 :	5
IV	упражнение Первый JAR 01 :	7
V	упражнение JCommander & JCDP 02 :	8

Глава I

Предисловие

Любая библиотека или фреймворк Java представляет собой набор JAR-файлов - архивов скомпилированных классов и других ресурсов.

Таким образом, цель любого Java-разработчика - правильно организовать исходный код, а затем передать скомпилированный JAR-архив с реализованной функциональностью другому программисту.

Существует целый ряд инструментов для организации жизненного цикла создания проекта и его структуры. Тем не менее, определенные навыки использования стандартных инструментов инфраструктуры Java гарантируют правильное понимание того, как работают готовые и популярные решения.

Сегодня вы будете вручную собирать приложение с внешними библиотеками. Это ваш первый шаг в изучении Maven - самой популярной системы сборки.

Глава II


Инструкции

- Используйте эту страницу как единственную ссылку. Не слушайте никаких слухов и домыслов о том, как приготовить раствор.
- Теперь для вас существует только одна версия Java - 1.8. Убедитесь, что компилятор и интерпретатор этой версии установлены на вашей машине.
- Вы можете использовать IDE для написания и отладки исходного кода.
- Код чаще читают, чем пишут. Внимательно прочитайте [документ](#), в котором приведены правила форматирования кода. При выполнении каждого задания убедитесь, что вы следуете общепринятым [стандартам Oracle](#)
- Комментарии не допускаются в исходном коде вашего решения. Они затрудняют чтение кода.
- Обратите внимание на разрешения ваших файлов и каталогов.
- Чтобы ваше решение было оценено, оно должно находиться в вашем GIT-репозитории.
- Ваши решения будут оценивать ваши товарищи по аквариуму.
- Вы не должны оставлять в своем каталоге никаких других файлов, кроме тех, которые явно указаны в инструкциях к упражнению. Рекомендуется изменить свой .gitignore во избежание несчастных случаев.
- Когда вам нужно получить точный вывод в ваших программах, запрещено выводить предварительно рассчитанный вывод вместо правильного выполнения упражнения.
- У вас есть вопрос? Спросите своего соседа справа. В противном случае попросите поговорить с соседом слева.
- Ваше справочное пособие: товарищи / Интернет / Google. И еще кое-что. На любой ваш вопрос есть ответ на Stackoverflow. Научитесь правильно задавать вопросы.
- Внимательно прочитайте примеры. В них могут потребоваться вещи, которые не указаны в предмете.
- Используйте "System.out" для вывода

- И да пребудет с вами Сила!
- Никогда не оставляйте на завтра то, что вы можете сделать сегодня ;)

Глава III

Упражнение 00 : Пакеты

	Упражнение 00
Пакеты	
Входящий каталог : <i>exoo/</i>	
Файлы для сдачи : ImagesToChar-folder (исключить цель)	
Разрешенные функции : Все	

Код может быть организован на разных уровнях. Пакеты - это один из методов организации кода, при котором классы располагаются в отдельных папках.

Теперь ваша задача - реализовать функциональность, которая выводит в консоль двухцветное изображение.

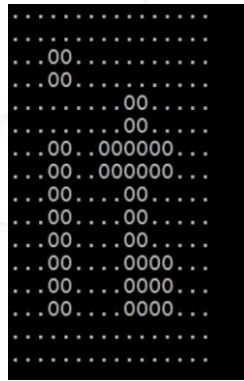
Пример черно-белого изображения в формате BMP (этот формат обязателен для решения). Размер изображения 16*16 пикселей.

it

Вы можете найти это изображение на странице проекта.

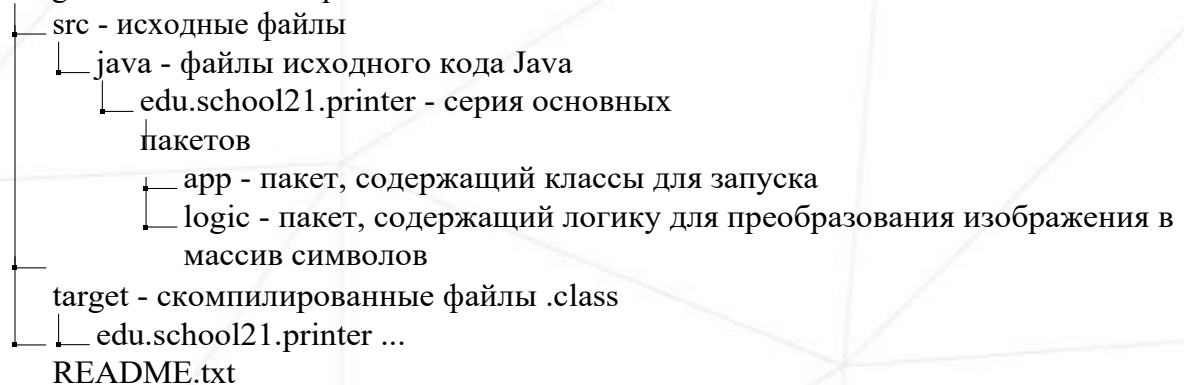
Ваше приложение должно принимать входные параметры, соответствующие символам, которые должны отображаться вместо белых и черных пикселей. Другим основным параметром запуска функции является полный путь к изображению на вашем диске.

Если символ "." используется для белого цвета, а "0" - для черного, изображение в консоли может выглядеть следующим образом:



Логика приложения должна быть распределена между различными пакетами и иметь следующую структуру:


ImagesToChar - папка проекта



- Файл README.txt должен содержать инструкции по компиляции и запуску вашего исходного кода из консоли (не IDE). Инструкция пишется для состояния, когда консоль открыта в корневой папке проекта.

Глава IV

упражнение 01 : Первый JAR

	Упражнение
	01 Первый
	JAR
Входящий каталог : <i>ex01/</i>	
Файлы для сдачи : ImagesToChar-folder (исключить цель)	
Разрешенные функции : Все	

Теперь необходимо создать дистрибутивный пакет приложения - JAR-архив. Важно, чтобы образ содержался в этом архиве (параметр командной строки для полного пути к файлу в этой задаче не требуется).

Должна соблюдаться следующая структура проекта:


ImagesToChar - папка проекта

- ├─ src - исходные файлы
 - ├─ java - файлы с исходным кодом Java
 - └─ ...
 - └─ resources - папка с файлами ресурсов
 - └─ image.bmp - отображаемое изображение
- ├─ manifest.txt - файл, содержащий описание начальной точки для запуска архива
- ├─ target - скомпилированные файлы .class и архив
 - ├─ edu.school21.printer ...
 - ├─ ресурсы
 - └─ images-to-chars-printer.jar
- └─ README.txt

- Архив и все скомпилированные файлы должны быть помещены в целевую папку во время сборки (без ручного переноса файлов; вы можете применить команду `cp` к папке ресурсов).
- Файл README.txt также должен содержать информацию о сборке и запуске архива.

Глава V

упражнение 02 : JCommander & JCDP

	Упражнение 02
JCommander & JCDP	
Входящий каталог : <i>ex02/</i>	
Файлы для сдачи : ImagesToChar (excludelib и target)	
Разрешенные функции : Все	

Теперь вы должны использовать внешние библиотеки:

- JCommander для командной строки.
- JCDP или JColor для использования цветного вывода

Архивы с этими библиотеками должны быть загружены и включены в проект предыдущего задания.

Теперь параметры запуска приложения должны быть обработаны с помощью инструментов JCommander. Изображение должно быть выведено с помощью опции "цветного" вывода библиотеки JCDP.

Требуемая структура проекта:

```
ImagesToChar - папка
├── проекта lib - папка
│   ├── внешней библиотеки
│   │   ├── jcommander-*.**.jar
│   │   └── JCDP-*.**.jar/JCOLOR-*.**.jar
│   └── src - исходные файлы
│       ├── target - скомпилированные файлы .class и архив
│       │   ├── edu.school21.printer
│       │   ├── com/beust .....файлы классов библиотеки JCommander
│       │   ├── com/diogonunes ..... файлы классов библиотеки JCDP
│       │   ├── ресурсы
│       │   └── images-to-chars-printer.jar
│       └── README.txt
```

Файл README.txt также должен содержать информацию о включении внешних библиотек в финальную сборку.

Пример работы программы:

```
$ java -jar images-to-chars- printer .jar -- white=RED --black=GREEN
```

