

Модуль 04 - Бассейн Java

JAR

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

Содержание

I Предисловие

II Инструкции

III Упражнение 00: Пакеты

IV упражнение 01: Первая банка

Упражнение 02: JCommander и JCDP

Глава I

Предисловие

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

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


Имеется ряд инструментов для организации жизненного цикла построения 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 вручную создадите приложение с внешними библиотеками. Это ваш первый шаг в изучении Maven — самой популярной системы построения.

Глава III

Упражнение 00: Пакеты

	Exercise 00
Packages	
Turn-in directory : <i>ex00/</i>	
Files to turn in : ImagesToChar-folder (exclude target)	
Allowed functions : All	

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 функционал, выводящий в консоль двухцветное изображение.

Пример черно-белого BMP-изображения (данный формат обязателен для решения).

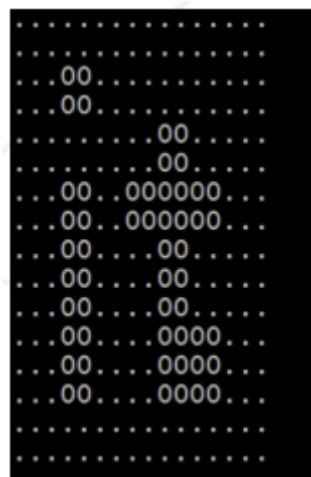
Размер изображения 16*16 пикселей.

it

Вы можете найти это изображение на странице проекта.

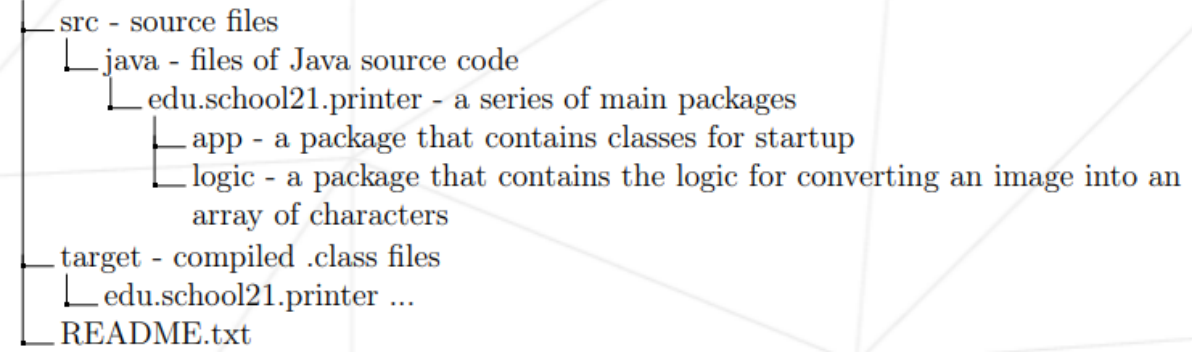
Ваше приложение должно принимать входные параметры, соответствующие символам, которые должны отображаться вместо белых и черных пикселей. Еще один параметр запуска основной функции это полный путь к образу на вашем диске.

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



Логика приложения должна быть распределена между разными пакетами и иметь следующие структура:


ImagesToChar - project folder



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

Глава IV

Упражнение 01: первая банка

	Exercise 01
First JAR	
Turn-in directory : <i>ex01/</i>	
Files to turn in : ImagesToChar-folder (exclude target)	
Allowed functions : All	

Теперь необходимо создать дистрибутив 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 - project folder


```
├── src - source files
│   ├── java - files of Java source code
│   │   └── ...
│   ├── resources - a folder with resource files
│   │   ├── image.bmp - the displayed image
│   │   └── manifest.txt - a file containing the description of the initial point for archive startup
│   └── target - compiled .class files and archive
│       ├── edu.school21.printer ..
│       ├── resources
│       ├── images-to-chars-printer.jar
│       └── README.txt
```

Архив и все скомпилированные файлы должны быть помещены в целевую папку при сборке (без ручного переноса файлов, вы можете применить команду `sr` к папке ресурсов).

- Файл `README.txt` также должен содержать информацию о сборке и запуске архива.

Глава V

Упражнение 02: JCommander и JCDP

	Exercise 02
JCommander & JCDP	
Turn-in directory : <i>ex02/</i>	
Files to turn in : ImagesToChar (excludelib and target)	
Allowed functions : All	

Теперь вы должны использовать внешние библиотеки:

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

Архивы с этими библиотеками нужно скачать и включить в проект предыдущей задачи.

Теперь параметры запуска приложения должны обрабатываться средствами JCommander. Изображение должно отображаться с использованием «цветного» параметра вывода библиотеки JCDP.

Необходимая структура проекта:

ImagesToChar - project folder

```
├── lib - external library folder
│   ├── jcommander-*.**.*.jar
│   └── JCDP-*.**.*.jar/JCOLOR-*.**.*.jar
├── src - source files
├── target - compiled .class files and archive
│   ├── edu.school21.printer
│   ├── com/beust ... - .class files of JCommander library
│   ├── com/diigonunes ... - .class files of JCDP library
│   ├── resources
│   └── images-to-chars-printer.jar
└── README.txt
```

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

Пример работы программы:

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

