**16版.订阅报刊杂志**

1、实验要求

    本实验要求：以读者订阅报刊杂志为背景，体验多线程生产/订阅模式下的经典应用。

    1-1. 业务说明：

        1-1.1. 本实验以实际生活中报刊杂志的订阅为业务背景。

        1-1.2. 订阅者通常有很多种，他们可以同时向出版社发出订阅申请。

        1-1.3. 出版社通常根据自身的资源条件对收到的订阅申请做统一的处理。

        1-1.4. 出版社通常能接受并处理的订阅申请量是有一定限度的。

        1-1.5. 出版社汇总订阅申请信息后交由独立业务单元处理，直至完毕。

        1-1.6. 本业务主要是模拟多读者同时订阅、信息汇总后再行处理的业务情景。

        1-1.7. 本实验的关键业务概念：

                1） 订阅者，即订阅信息生产者，向共享池中放入订阅信息；

                2） 信息汇总，即订阅信息共享池，生产者和消费者共用；

                3） 业务单元，即订阅信息消费者，处理订阅者生产的订阅信息。

    1-2. 创建工程并配置环境：

        1-2.1. 限制1. 创建实验工程，命名为：SE\_JAVA\_EXP\_E046；

        1-2.2. 限制2. 创建包，命名为：cn.campsg.java.experiment；

        1-2.3. 限制3. 创建包，命名为：cn.campsg.java.experiment.entity。

    1-3. 创建订阅池业务类：

        1-3.1. 限制1. 在cn.campsg.java.experiment.entity包中创建订阅信息共享池业务类：SharePool。

        1-3.2. 为SharePool类创建属性，要求如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 属性权限 | 属性名称 | 属性类型 | 属性说明 |
| 1 | private | pool | List<String> | 订阅信息共享池，为生产者和消费者服务 |
| 2 | private | MAX | int | 最大订阅量。定为常量，初始化为15。 |
| 说明：1.只为上述pool属性实现setter/getter方法； | | | | |

        1-3.3. 为奖项SharePool类创建0参构造器。

                1） 在构造器内完成对共享池的初始化创建。

                2） 共享池初始化是为了包含最大限量的订阅信息。

        1-3.4. 为SharePool创建生成订阅信息的方法，要求如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 访问权限 | 方法名称 | 返回值 | 方法参数 | 参数说明 |
| public | produce | void | String media | 订阅报刊杂志名称 |
| 方法说明 | 本方法由订阅者调用，生成订阅信息并完成订阅信息的提交。 | | | |

            本方法的业务逻辑实现要求如下：

                1）应用同步机制控制本方法的实现；

                2） 获取当前订阅信息共享池的信息数量；

                3） 当共享池信息数量等于最大订阅量时，输出如下信息并开始等待：

                    “订阅请求队列已满，等待系统处理订阅请求中……”

                4） 当共享池信息数量小于最大订阅量时：

                    其一，把订阅信息加入到共享池；

                    其二，在控制台输出如下格式的信息：

                        "订阅者@"+当前线程名+":订阅《" + media + "》申请已提交.当前订阅数量为：" + 订阅数量

                    其三，发出完成订阅申请的通知。

        1-3.5. 为SharePool创建处理订阅信息的方法，要求如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 访问权限 | 方法名称 | 返回值 | 方法参数 | 参数说明 |
| public | consume | void | 无 | 无 |
| 方法说明 | 本方法由订阅信息消费者调用，完成订阅信息的处理。 | | | |

            本方法的业务逻辑实现要求如下：

                1） 应用同步机制控制本方法的实现；

                2） 获取当前订阅信息共享池的信息数量；

                3） 当共享池信息数量等于0时，输出如下信息并开始等待：

                    “处理者@”+当前线程名称+“暂无订阅请求信息，等待中……”

                4） 当共享池信息数量不为0时：

                    其一，从订阅共享池移除一项订阅信息；

                    其二，在控制台输出如下格式的信息：

                        "处理者@"+当前线程名+":处理《" + 移除订阅信息 + "》订阅已完毕。尚待处理订阅数量为：" + 剩余订阅数量

                    其三，发出完成订阅处理的通知。

    1-4.  创建订阅生产业务类：

        1-4.1. 限制1. 在cn.campsg.java.experiment.entity包中创建订阅信息生产者业务类：Producer。

                 1） Producer类需要实现线程类接口Runnable，并且实现run方法。

        1-4.2. 为Producer类创建订阅信息共享池属性,要求如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 属性权限 | 属性名称 | 属性类型 | 属性说明 |
| 1 | private | pool | SharePool | 表示订阅信息共享池 |
| 2 | private | count | int | 表示订阅者订阅的数量 |
| 说明：无需为上述属性实现对应的setter/getter方法。 | | | | |

        1-4.3. 为Producer类创建0参构造器。

        1-4.4. 为Producer类创建2参构造器：

                1） 两个参数分别表示订阅共享池和订阅的数量；

                2） 在构造器内容为对应的属性赋值。

        1-4.5. 为Producer类实现生产订阅信息的方法，描述如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 访问权限 | 方法名称 | 返回值 | 方法参数 | 参数说明 |
| public | run | void | 无 | 无 |
| 方法说明 | 实现本方法，以产生指定数量订阅信息并存放到订阅共享池中。 | | | |

            本方法的业务逻辑实现要求如下：

                1） 模拟输出订阅者相关信息，格式如下：

                    “订阅者@”+当前线程名称+“：订阅”+订阅数量+“份。”；

                2） 通过循环方式发出所有指定数量的订阅请求：

                    其一，当订阅数量发送完毕后，结束订阅并输出如下信息：

                        “订阅者@”+当前线程名称+“->完成订阅。”

                    其二，定义计数器，把信息放入到订阅信息共享池并输出如下信息：

                        “订阅者@”+当前线程名称+“ 在提交第”+计数器+“ 份订阅申请。”

                    其三，让当前线程休眠半秒钟；

                    其四，订阅完毕后退出方法。

    1-5. 创建订阅处理业务类：

        1-5.1. 限制1. 在cn.campsg.java.experiment.entity包中创建订阅信息处理者业务类：Consumer。

                1） Consumer类需要实现线程类接口Runnable，并且实现run方法。

        1-5.2. 为Consumer类创建订阅信息共享池属性,要求如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 属性权限 | 属性名称 | 属性类型 | 属性说明 |
| 1 | private | pool | SharePool | 表示待处理的订阅信息共享池 |
| 说明：无需为上述属性实现对应的setter/getter方法。 | | | | |

        1-5.3. 为Consumer类创建0参构造器。

        1-5.4. 为Consumer类创建1参构造器：

                1） 参数代表订阅信息共享池；

                2） 在构造器内为对应的属性赋值。

        1-5.5. 为Consumer类实现处理订阅信息的方法，要求如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 访问权限 | 方法名称 | 返回值 | 方法参数 | 参数说明 |
| public | run | void | 无 | 无 |
| 方法说明 | 本方法产生指定数量的订阅信息并存放到订阅信息共享池中。 | | | |

            本方法的业务逻辑实现要求如下：

                1） 创建处理订阅信息数量的计数器；

                2） 通过循环方式处理共享池内的所有订阅信息：

                    其一，启动计数器，当计数器大于15时，结束处理并输出如下信息：

                        “本线程完成订阅处理量，即刻退出。处理者@”+当前线程名称

                    其二，当计数器小于15时，处理共享池内的订阅信息并输出如下信息：

                        “订阅者@”+当前线程名称+“ ：处理第”+计数器+“ 份订阅。”

                    其三，处理全部完成后，退出信息订阅池。

    1-6.  模拟现实生产/处理的结果：

        1-6.1. 限制1. 在cn.campsg.java.experiment包中创建主类：MainClass；

        1-6.2. 为MainClass创建入口主方法：main；

        1-6.3. 在main中，创建1个订阅信息共享池对象；

        1-6.4. 在main中，创建5个不同的订阅者，订阅总量为15，共用共享池；

        1-6.5. 在main中，创建1个订阅处理类对象，处理所有提交的订阅信息。

2、实现思路

     2-1. 创建工程并配置环境。

     2-2. 创建订阅池业务类：

        2-2.1. 在包cn.campsg.java.experiment.entity中创建奖项类：SharePool；

        2-2.2. 为SharePool类定义属性：

                1） 订阅信息共享池： private List<String> pool;

                2） 最大订阅量： private final int MAX;

                3） 为pool属性实现对应的setter/getter方法

        2-2.3. 为SharePool类创建默认的0参构造器。

                1） 在构造器内创建空的共享池对象，并赋值给pool变量。

        2-2.4. 为SharePool类创建生产订阅信息的方法，其形式如下:

|  |
| --- |
| + 提示   public void produce(String media) {   } |

            方法实现逻辑如下：

                其一，在方法体内建立对象本身的同步锁处理块:

                    建立同步锁：synchronized (this) {}

                以下业务逻辑，要求在同步块内处理。

                其二，判断共享池内的订阅数量：达到MAX时，则输出如下信息并等待：

                    控制台输出：“订阅请求队列已满，等待系统处理订阅请求中……”

                    调用系统API开始等待：wait();

                其三，若未达到MAX，则把media加入到pool，并输出如下信息：

                    "订阅者@"+当前线程名+":订阅《" + media + "》申请已提交.当前订阅数量为：" + 订阅数量

|  |
| --- |
| + 提示    当前线程名的获取方法为：Thread.currentThread().getName() |

                其四，加入共享池后，调用系统API通知方法：notify();

        2-2.5. 为SharePool创建订阅信息的处理方法，其形式如下:

|  |
| --- |
| + 提示   public void  consume() {   } |

            该方法实现逻辑如下：

                其一，在方法体内建立对象本身的同步锁处理块:

                    建立同步锁：synchronized (this) {}

                以下业务逻辑，要求在同步块内处理。

                其二，判断共享池内的订阅数量：若为0，则输出如下信息并等待：

                    控制台输出：“处理者@”+当前线程名称+“暂无订阅请求信息，等待中…”

                    然后调用系统方法开始等待：wait();

                其三，判断共享池内的订阅数量：若不为0，则移除一项订阅信息，并输出如下信息：

                    移除订阅信息：String media = pool.remove(0);

                    控制台输出："处理者@"+当前线程名+":处理《" + media + "》订阅已完毕。尚待处理订阅数量为：" + 剩余订阅数量

                其四，处理完毕后，调用系统通知方法：notify()；

    2-3. 创建订阅生产业务类:

        2-3.1. 在包cn.campsg.java.experiment.entity中创建订阅信息生产者业务类：Producer；

        2-3.2. 为Producer类定义私有属性：

                1） 订阅信息共享池：SharePool pool;

                2） 订阅数量：int count；

        2-3.3. 为Producer类创建0参构造器；

        2-3.4. 为Producer类创建2参构造器，其形式如下：

|  |
| --- |
| + 提示   public  Producer(SharePool  pool,int num) {   } |

                1） 参数pool为共享池；参数num为订阅数量；

                2） 在构造器内为对应的属性赋值：

                    this.pool = pool ;

                    this.count = num ;

        2-3.5. 为Producer类实现生产订阅信息的方法，其形式如下：

|  |
| --- |
| + 提示   public void run() {   } |

            本方法的业务实现逻辑如下：

                1） 定义订阅信息名称变量：String media=“报刊杂志”；

                2） 定义订阅信息计数器变量：int counter=0；

                3） 输出如下信息：

                    “订阅者@”+当前线程名称+“：订阅”+count+“份。”；

                4） 通过循环方式开始在循环体内提交指定数量的订阅申请：

                    其一，若count<=0成立，则输出如下信息并退出循环：

                        “订阅者@”+当前线程名称+“->完成订阅。”

                    其二，启动计数器并输出如下信息：

                        counter++；

                        “订阅者@”+当前线程名称+“ 在提交第”+counter+“ 份订阅申请。”

                    其三，把一份订阅信息放入到共享池并更新订阅数量：

                        pool.produce(media+counter);

                        count--；

                    其四，让当前线程休眠半秒钟：

                        Thread.sleep(500);

    2-4. 创建订阅处理业务类:

        2-4.1. 在包cn.campsg.java.experiment.entity中创建订阅信息处理者业务类：Consumer；

        2-4.2. 为Consumer类定义私有属性：

                1） 订阅共享池：SharePool pool;

        2-4.3. 为Consumer类创建0参构造器；

        2-4.4. 为Consumer类创建1参构造器，其形式如下：

|  |
| --- |
| + 提示   public Consumer(SharePool  pool) {   } |

                1） 参数pool为共享池；

                2） 在构造器内为对应的属性赋值：

                    this.pool = pool ;

        2-4.5. 为Consumer类实现处理订阅信息的方法，其形式如下：

|  |
| --- |
| + 提示   public void run() {   } |

                本方法的业务实现逻辑如下：

                    1） 定义订阅信息计数器变量：int counter=0；

                    2） 通过循环方式开始在循环体内处理共享池内的订阅信息：

                        其一，启动计数器：counter++；

                        其二，若counter>15 成立，则输出如下信息并退出循环：

                            "本线程完成订阅处理量，即刻退出。处理者@"+当前线程名称

                        其三，若小于15，继续输出如下信息，并调用共享池处理订阅的方法：

                            "处理者@"+当前线程名+":处理第 "+counter+"份订阅。"

                            this.pool.consume();

    2-5.  模拟现实生产/处理的结果：

        2-5.1. 在包cn.campsg.java.experiment中创建业务主类：MainClass；

        2-5.2. 在MainClass中定义程序入口主方法：main；

        2-5.3. 把以下测试代码粘贴到main方法中：

|  |
| --- |
| + 提示   public static void main(String[] args) {        SharePool pool = new SharePool();        //模拟批量订阅：开启5个线程订阅，共订阅15份        for(int i=1;i<=5;i++){           new Thread(new Producer(pool,i)).start();        }        //只创建一个消费者线程        new Thread(new Consumer(pool)).start();   } |

3、验证与测试

    3-1. 在工程中定位主类MainClass。

    3-2. 右键点击MainClass类，依次选择：Run As->Java Application。

    3-3. 运行程序，查看输出结果是否符合预期。