重写的问题：

1. 在圣诞老人的帮助下，孩子们会因为表现好而获得礼物。求开发需求模型，用例模型和领域模型

需求模型：1) 系统必须有一个孩子和他们成绩的数据库。

2) 系统应为家长提供在系统中注册孩子的机会。

3）系统必须存储关于每个孩子的行为的信息。

4) 系统必须有一个礼物目录和他们的描述。

5) 系统应该能够向父母发送关于他们孩子的成就和礼物的通知。

6) 系统应能根据孩子的成绩自动分配礼物。

用例模型：

儿童登记：

家长填写儿童登记表。

系统将孩子添加到数据库中。

添加成就：

父母添加孩子的成就。

系统更新数据库中有关孩子行为的信息。

领取礼物：

取得一定成绩的孩子会收到礼物通知。

家长会收到给孩子的礼物通知。

查看成就：

家长可以在系统中查看孩子的成绩。

领域模型：

孩子：名字、姓氏、年龄、成就。

父母：名字，姓氏，联系方式。

成就：描述，点数。

礼物：名称、描述、积分数。

1. Disciplined Agile Delivery开发模式，和其他模式的

Disciplined Agile Delivery (DAD) 是一种敏捷交付框架，旨在帮助组织实现高质量、可预测和可扩展的交付流程。

DAD模型的优点：

•为不同类型的项目和组织提供灵活性和适应性

•适用于大型项目。

• 考虑项目的整个生命周期，包括规划、架构、测试和部署。

• 允许您根据项目要求结合敏捷和非敏捷方法。

• 提供许多工具和实践来提高产品质量。

DAD模型的缺点：

• 由于大量不同的流程、实践和工具，DAD 具有高度的复杂性。

•此模型需要高水平的团队能力才能正确实施所有流程和实践。

1. 用例表

|  |  |  |  |
| --- | --- | --- | --- |
| **Начальное состояние** | **Ввод пользователя** | **Вывод системы** | **Конечное состояние** |
| Готов  (Количество добрых дел ребенка = 12) | 12 | Назначен подарок | Ожидание утверждения от  Санта-Клауса |
| Ожидание утверждения от  Санта-Клауса | Санта-Клаус дает подтверждение | Подтверждение  Санта-Клауса | Готов |

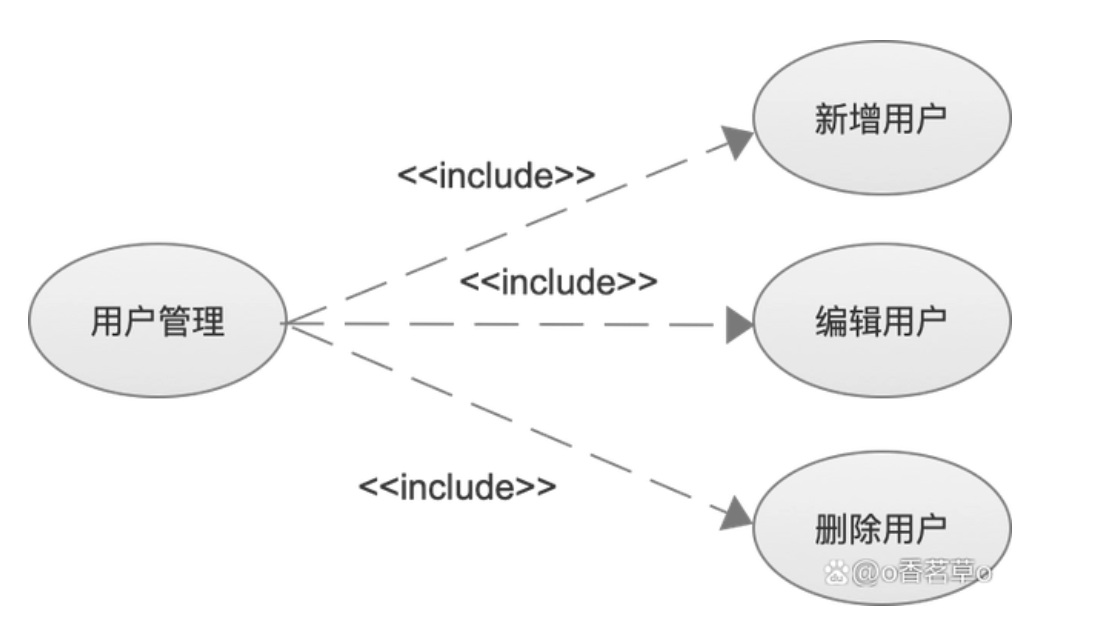
|  |  |  |  |
| --- | --- | --- | --- |
| **Начальное состояние** | **Ввод пользователя** | **Вывод системы** | **Конечное состояние** |
| Готов  (Количество добрых дел ребенка = 0) | 0 | Назначен уголек | Ожидание утверждения от  Санта-Клауса |
| Ожидание утверждения от  Санта-Клауса | Санта-Клаус дает подтверждение | Подтверждение  Санта-Клауса | Готов |

|  |  |  |  |
| --- | --- | --- | --- |
| **Начальное состояние** | **Ввод пользователя** | **Вывод системы** | **Конечное состояние** |
| Готов (Количество добрых дел ребенка = 9) | 9 | Назначен уголек | Ожидание утверждения от  Санта-Клауса |
| Ожидание утверждения от  Санта-Клауса | Санта-Клаус отклоняет | Отклонение Санта-Клауса | Готов |

|  |  |  |  |
| --- | --- | --- | --- |
| **Начальное состояние** | **Ввод пользователя** | **Вывод системы** | **Конечное состояние** |
| Готов (Количество добрых дел ребенка =  -3) | -3 | Неверный ввод данных | Готов  (Начальное состояние) |
| Готов (Количество добрых дел ребенка = «десять») | «десять» | Неверный ввод данных | Готов  (Начальное состояние) |

**Вариант 1:**

1, 用例图



1. Royce的模型

Royce模型，也称为瀑布模型（Waterfall Model），是软件开发过程中最早提出的一种线性顺序模型

1. preliminary desig初步设计

需求生成阶段与分析阶段插入一个预设计阶段，我们可以提前确保程序不会因为时间，储存或者数据流方面出现问题而失败，而在分析时，确保设计师把限制提交给分析师面前，让分析师分析时必须考虑这些限制。

1. documentation文档

管理需要大量的文档，没有足够的文档，根本无法对项目做出有效的管理。

1. do it twice做两次

如果是第一次开发某个计算机程序，先整理相关问题，考虑设计/运营方面的问题，再做一次，确保交付给客户的实际上是第二个版本

1. testing测试

有很多错误是可以通过视觉上的审查来发现的，应该把分析和设计提交给除分析者和编码者之外的人来评审

1. user connection用户连接

使用户尽早参与开发过程，可以澄清一些不确定的问题，初步软件审评，关键软件审评，最终软件审评

1. 哪些风险是资源风险

a开发商设备损坏风险

b承包商经理与客户之间发生的冲突风险

c从生产转移到其它城市时缺乏必要人员的风险

d制成品不足的风险

e禁止销售内燃机汽车以减少对地球环境负面影响的风险

1-资源

2-商业风险

3-资源

4-商业风险

5-政治

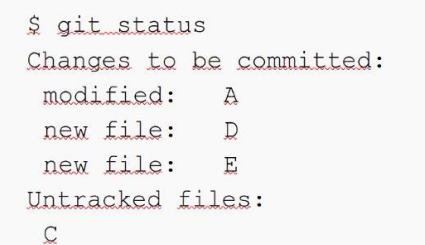
答案a c

1. 前三大风险，是由风险概率x风险损失金额
2. 0.1 \* 20000 = 2000
3. 0.25 \* 100000 = 25000
4. 0.2 \* 100000 = 20000
5. 0.15 \* 10000 =1500
6. 0.25 \* 20000 = 5000

Ответ: 2 3 5

1. Git pull指令有什么作用

下载最新版本的远程存储库并将其应用到本地存储库



提交ace

git reset HEAD D

git add C

1. 程序员心情的来上班，告诉他的同事他讨厌他们所有人，而且不小心将不正确的代码添加到程序中
2. mistake （error）人为的失误
3. Fault软件本身的缺陷（指在软件代码中存在的技术性错误或逻辑错误）
4. Failure软件在执行中无法满满足预期的效果或功能
5. Error用于描述软件开发中一般的问题，相对广泛的术语
6. Bug 是指在软件中存在的实际问题或缺陷

Gradle

task COMP {

doLast { println("COMP") }

}

task RESOURCE {

doLast { println("RESOURCE") }

}

task LIB {

doLast { println("LIB") }

}

task BUILD {

doLast { println("BUILD") }

}

task DEPLOY {

doLast { println("DEPLOY") }

}

build.dependsOn COMP, RESOURCE, LIB, BUILD, DEPLOY

Make

.PHONY: all COMP RESOURCE LIB BUILD DEPLOY

all: COMP RESOURCE LIB BUILD DEPLOY

COMP: @echo "COMP"

RESOURCE: @echo "RESOURCE"

LIB: @echo "LIB"

BUILD: @echo "BUILD"

DEPLOY: @echo "DEPLOY"

<project name="example build script">

<target name="COMP">

<echo message="COMP"/>

</target>

<target name="RESOURCE">

<echo message="RESOURCE"/>

</target>

<target name="LIB">

<echo message="LIB"/>

</target>

<target name="BUILD">

<echo message="BUILD"/>

</target>

<target name="DEPLOY">

<echo message="DEPLOY"/>

</target>

</project>

<project>

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-antrun-plugin</artifactId>

<version>1.8</version>

<executions>

<execution>

<id>print-targets</id>

<phase>compile</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo message="COMP" />

</tasks>

</configuration>

</execution>

<execution>

<id>print-targets</id>

<phase>compile</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo message="RESOURCE" />

</tasks>

</configuration>

</execution>

<execution>

<id>print-targets</id>

<phase>compile</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo message="LIB" />

</tasks>

</configuration>

</execution>

<execution>

<id>print-targets</id>

<phase>compile</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo message="BUILD" />

</tasks>

</configuration>

</execution>

<execution>

<id>print-targets</id>

<phase>compile</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo message="DEPLOY" />

</tasks>

</configuration>

</execution>

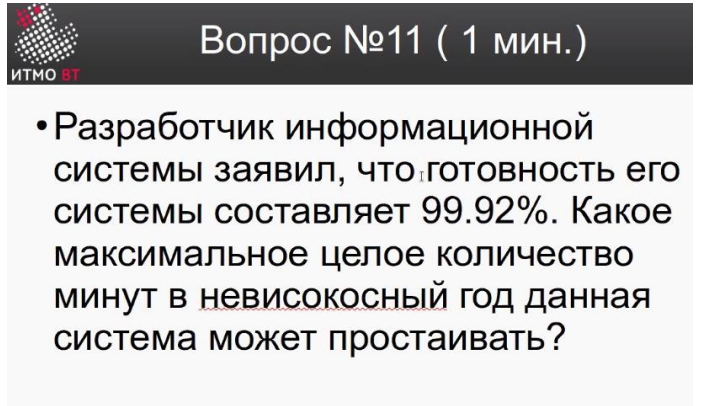
</executions>

</plugin>

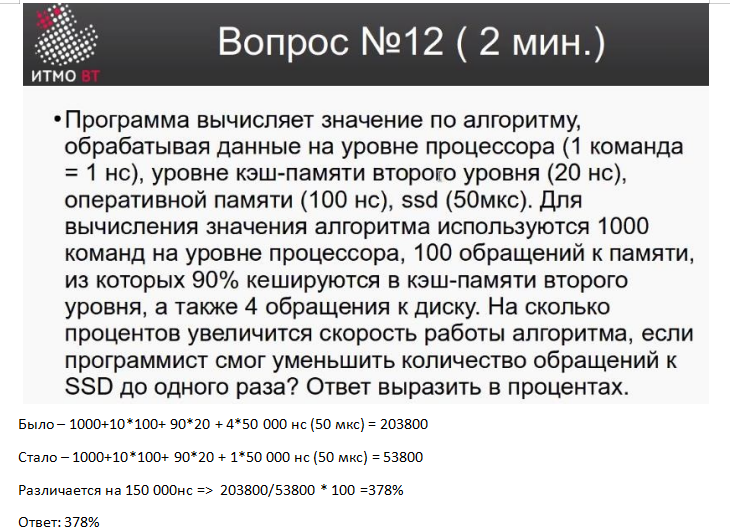
</plugins>

</build>

</project>



Ответ: 365\*24\*60 \* 0.08% (0.0008) = 420.48 = **420**



**Вариант 2:**

1. 领域模型旨在对特定领域的问题和需求进行建模