Exercise #2: **Subscription**

Details:

The contract handles one subscription. Anybody can pay into it but there is only one recipient. During construction time the contract creator supplies the address of the recipient, the duration of one payment period and the price in wei per payment period. Anyone can then pay ether into the contract. This money cannot be retrieved until either a payment is due or the subscription is cancelled.

When there is an outstanding payment, anybody can call the collect method. If there is a due payment, the money for one time period is paid out to the recipient and a Paid event is emitted. If there is not enough money, a FailedToPay event is emitted instead. (The recipient should suspend his provided service temporarily).

The contract creator can also cancel the subscription. This is only possible if there are no due payments. The remaining ether in the contract gets sent to the creator and a Cancelled event is emitted. If the sender of cancel is not the creator the contract throws. Both the collect and cancel function don't work afterwards.

Interface:

```
function Subscription(address recipient_, uint price_, uint time_)
event FailedToPay();
event Paid();
event Cancelled();
function collect() returns (bool);
function cancel() returns (bool);
```

Example:

00:00 Bob creates a contract for 10 ether per 4 hours with the Alice as recipient.
Bob deposits 25 ether into the contract.
01:00 Alice calls the collect function and gets 10 ether. A paid event is emitted.
15 ether remain in the contract.
02:00 Alice calls the collect function and gets nothing. Not enough time has passed.
15 ether remain in the contract.
03:00 Carol pays 2 ether into the contract. This is ok, anybody can pay.
17 ether remain in the contract.
04:00 Carol calls the collect function and Alice gets 10 ether. A Paid event is emitted.
7 ether remain in the contract.
05:00 Carol calls the cancel function. Nothing happens (throws), Carol is not the creator.
09:00 Bob calls the cancel function. Nothing happens (does not throw), there is one outstanding payment.
10:00 Alice calls the collect function. There are too few ether in the contract.
A FailedToPay event is emitted. 7 ether remain in the contract.
10:30 Bob deposits 5 ether. 12 ether remain in the contract.
10:40 Bob calls collect and Alice gets 10 ether. A Paid event is emitted.
2 ether remain in the contract.
11:00 Bob calls cancel. The subscription ends and the remaining 2 ether are sent to Bob.
13:00 Alice calls collect. Nothing happens, the subscription has already been cancelled.
14:00 Bob calls cancel. Nothing happens, the subscription has already been cancelled.

Relevant global variables: msg.sender, block.timestamp, this.balance
Hints: address.send(value), use modifiers for code reuse